© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

# RANCANG BANGUN SISTEM MANAJEMEN PROSES BISNIS PADA PERUSAHAAN JASA KONSULTAN
# (STUDI KASUS: PT DSK GLOBAL KONSULTAMA)

## SUBJUDUL:
## PENGEMBANGAN APLIKASI *MONITORING* PROYEK DAN PENILAIAN KINERJA KARYAWAN BERBASIS *KEY PERFORMANCE INDICATOR* (KPI)

**SKRIPSI**

**Muhamad Faried 2207412041**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER**

**POLITEKNIK NEGERI JAKARTA**

**2026**

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

# RANCANG BANGUN SISTEM MANAJEMEN PROSES BISNIS PADA PERUSAHAAN JASA KONSULTAN (STUDI KASUS: PT DSK GLOBAL KONSULTAMA)

## SUBJUDUL:

## PENGEMBANGAN APLIKASI *MONITORING* PROYEK DAN PENILAIAN KINERJA KARYAWAN BERBASIS *KEY PERFORMANCE INDICATOR* (KPI)

**SKRIPSI**

**Dibuat untuk Melengkapi Syarat-Syarat yang Diperlukan untuk Memperoleh Diploma Empat Politeknik**

**Muhamad Faried**

**2207412041**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER**

**POLITEKNIK NEGERI JAKARTA**

**2026**

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

## SURAT PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan di bawah ini:

| | | |
|---|---|---|
| Nama | : | Muhamad Faried |
| NIM | : | 2207412041 |
| Jurusan/Program Studi | : | T.Informatika dan Komputer/Teknik Informatika |
| Judul Skripsi | : | Rancang Bangun Sistem Manajemen Proses Bisnis pada Perusahaan Jasa Konsultan (Studi Kasus: PT DSK Global Konsultama) |
| Subjudul Skripsi | : | Pengembangan Aplikasi *Monitoring* Proyek dan Penilaian Kinerja Karyawan Berbasis *Key Performance Indicator* (KPI) |

Menyatakan dengan sebenarnya bahwa skripsi ini benar-benar merupakan hasil karya saya sendiri, bebas dari peniruan terhadap karya dari orang lain. Kutipan pendapat dan tulisan orang lain ditunjuk sesuai dengan cara-cara penulisan karya ilmiah yang berlaku.

Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa dalam skripsi ini terkandung ciri-ciri plagiat dan bentuk-bentuk peniruan lain yang dianggap melanggar peraturan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Depok, 9 Juni 2026

Yang membuat pernyataan,




Muhamad Faried

NIM 2207412041

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

## LEMBAR PENGESAHAN

Saya yang bertanda tangan di bawah ini:

| | | |
|---|---|---|
| Nama | : | Muhamad Faried |
| NIM | : | 2207412041 |
| Jurusan/Program Studi | : | T.Informatika dan Komputer/Teknik Informatika |
| Judul Skripsi | : | Rancang Bangun Sistem Manajemen Proses Bisnis pada Perusahaan Jasa Konsultan (Studi Kasus: PT DSK Global Konsultama) |
| Subjudul Skripsi | : | Pengembangan Aplikasi *Monitoring* Proyek dan Penilaian Kinerja Karyawan Berbasis *Key Performance Indicator* (KPI) |

Telah diuji oleh tim penguji dalam Sidang Skripsi pada hari Senin, tanggal 29 bulan Juni tahun 2026 dan dinyatakan **LULUS**.

Disahkan Oleh

| | | | |
|---|---|---|---|
| Pembimbing I | : | Risna Sari, S.Kom., M.T.I. | ( ) |
| Penguji I | : | Dr. Anita Hidayati, S.Kom., M.Kom. | ( ) |
| Penguji II | : | Asep Taufik Muharram, S.Kom, M.Kom. | ( ) |
| Penguji II | : | Fachroni Arbi Murad, S.Kom, M.Kom. | ( ) |

Mengetahui:

Jurusan Teknik Informatika dan Komputer

Ketua

Dr. Anita Hidayati, S.Kom., M.Kom.

NIP. 197908032003122003

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

## KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi berjudul "Pengembangan Aplikasi *Monitoring* Proyek dan Penilaian Kinerja Karyawan Berbasis *Key Performance Indicator* (Studi Kasus: PT DSK Global Konsultama)" sebagai salah satu syarat untuk memperoleh gelar Sarjana Terapan pada Program Studi D4 Teknik Informatika, Politeknik Negeri Jakarta. Penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

a. Bapak Galih Gumilang, S.E. BKP, selaku CEO PT DSK Global Konsultama, atas izin penelitian, dukungan pengembangan sistem, dan wawasan operasional yang sangat berarti bagi penelitian ini.
b. Ibu Risna Sari, S.Kom., M.T.I., selaku dosen pembimbing, yang dengan sabar memberikan arahan, saran, dan motivasi yang sangat berarti selama proses penyusunan skripsi ini dari awal hingga selesai.
c. Keluarga, orang tua, serta teman-teman yang senantiasa memberikan dukungan moral, doa, dan semangat tanpa henti, sehingga penulis dapat menyelesaikan skripsi ini dengan penuh keyakinan.
d. Nada Asyifa, sebagai kekasih saya yang selalu memberi dukungan, motivasi dan menemani proses penyusunan skripsi ini.

Penulis menyadari skripsi ini masih jauh dari sempurna dan mengharapkan kritik serta saran yang membangun. Semoga skripsi ini dapat memberikan kontribusi yang bermanfaat bagi pengembangan ilmu pengetahuan, khususnya di bidang sistem informasi dan manajemen kinerja berbasis teknologi.

Depok, 9 Juni 2026

Penulis

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

## SURAT PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Politeknik Negeri Jakarta, saya bertanda tangan di bawah ini:

| | | |
|---|---|---|
| Nama | : | Muhamad Faried |
| NIM | : | 2207412041 |
| Jurusan/Program Studi | : | T.Informatika dan Komputer/Teknik Informatika |

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Politeknik Negeri Jakarta Hak Bebas Royalti Non-Eksklusif atas karya ilmiah saya yang berjudul:

**Rancang Bangun Sistem Manajemen Proses Bisnis pada Perusahaan Jasa Konsultan (Subjudul: Pengembangan Aplikasi *Monitoring* Proyek dan Penilaian Kinerja Karyawan Berbasis *Key Performance Indicator* (KPI) (Studi Kasus: PT DSK Global Konsultama))**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksklusif ini Politeknik Negeri Jakarta Berhak menyimpan, mengalihmediakan/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan skripsi saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini Saya buat dengan sebenarnya.

Depok, 9 Juni 2026

Yang membuat pernyataan,




Muhamad Faried

NIM 2207412041

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

# PENGEMBANGAN APLIKASI *MONITORING* PROYEK DAN PENILAIAN KINERJA KARYAWAN BERBASIS *KEY PERFORMANCE INDICATOR* (KPI)

***Abstrak***

*PT DSK Global Konsultama menghadapi permasalahan monitoring proyek Transfer Pricing Documentation yang dilakukan secara manual melalui tiga media tidak terintegrasi, sehingga 12 hingga 15 dari sekitar 50 proyek per tahun mengalami keterlambatan, serta penilaian kinerja karyawan yang subjektif karena bobot indikator belum konsisten. Penelitian ini bertujuan merancang dan membangun aplikasi monitoring proyek berbasis workflow yang terintegrasi dengan penilaian kinerja berbasis Key Performance Indicator (KPI) menggunakan metode Systems Development Life Cycle. Sistem dibangun dengan React, TypeScript, dan Tailwind CSS pada sisi frontend serta Node.js, Express, dan MySQL pada sisi backend, mengimplementasikan modul Workflow Management System berbasis state machine deterministik, modul KPI Calculation Engine empat dimensi (Task Completion 35%, Timeliness 25%, Output Quality 25%, Update Compliance 15%) dengan metode Weighted Scoring Method, dan dashboard tersegmentasi per peran berbasis Role-Based Access Control dua lapis. Pengujian Black Box mencapai 100% pada 32* test case*, User Acceptance Testing menerima seluruh 5 skenario, dan Usability Testing dengan System Usability Scale terhadap 10 responden menghasilkan skor 78,0 berkategori Acceptable dengan Task Success Rate 92,5%. Sistem terbukti memenuhi kebutuhan fungsional dan diterima pengguna.*

***Kata Kunci:*** *monitoring Proyek, penilaian kinerja karyawan, Key Performance Indicator, Weighted Scoring Method, Workflow Management System*

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

# DAFTAR ISI

**SURAT PERNYATAAN BEBAS PLAGIARISME** ..... **i**
**LEMBAR PENGESAHAN** ..... **ii**
**KATA PENGANTAR** ..... **iii**
**SURAT PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS** ..... **iv**
**ABSTRAK** ..... **v**
**DAFTAR ISI** ..... **vi**
**DAFTAR GAMBAR** ..... **x**
**DAFTAR TABEL** ..... **xii**
**BAB I PENDAHULUAN** ..... **1**
1.1 Latar Belakang Masalah ..... 1
1.2 Perumusan Masalah ..... 2
1.3 Batasan Masalah ..... 2
1.4 Tujuan dan Manfaat ..... 3
1.4.1 Tujuan Penelitian ..... 3
1.4.2 Manfaat Penelitian ..... 4
1.5 Sistematika Penulisan ..... 4
**BAB II TINJAUAN PUSTAKA** ..... **6**
2.1 Kajian Teoretis ..... 6
2.1.1 Sistem Informasi ..... 6
2.1.2 Manajemen dan *Monitoring* Proyek ..... 7
2.1.3 *Workflow Management System* ..... 8
2.1.4 Manajemen Penilaian Kinerja Karyawan ..... 9
2.1.5 *Key Performance Indicator* ..... 9
2.1.6 *Weighted Scoring Method* (WSM) ..... 10
2.1.7 Hubungan *Monitoring* Proyek dengan Penilaian Kinerja ..... 11
2.1.8 *Dashboard* dan Visual Data ..... 12
2.1.9 *Role-Based* Access Control (RBAC) ..... 12
2.1.10 *Business Process Model and Notation* (BPMN) ..... 14
2.1.11 *Use Case Diagram* ..... 15
2.1.12 *Activity Diagram* ..... 15

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

2.1.13 *Black Box Testing* ..... 16
2.1.14 *User Acceptance Testing* ..... 17
2.1.15 *Usability Testing* dan *System Usability Scale* ..... 17
2.2 Teknologi yang digunakan ..... 19
2.2.1 React.js ..... 19
2.2.2 Node.js ..... 20
2.2.3 MySQL ..... 22
2.2.4 RESTful API ..... 23
2.2.5 Fetch API ..... 24
2.3 Penelitian Terdahulu ..... 26
**BAB III METODE PENELITIAN ..... 32**
3.1 Rancangan Penelitian ..... 32
3.1.1 Teknik Pengumpulan Data ..... 32
3.1.2 Teknik Analisis Data ..... 33
3.2 Tahapan Penelitian ..... 33
3.2.1 Identifikasi Masalah ..... 35
3.2.2 Analisis Kebutuhan ..... 35
3.2.3 Perancangan Sistem ..... 36
3.2.4 Implementasi ..... 37
3.2.5 Pengujian ..... 38
3.2.6 Evaluasi ..... 38
3.2.7 Penyusunan Laporan ..... 39
3.3 Objek Penelitian ..... 39
3.3.1 Objek Fungsional ..... 40
3.3.2 Objek Pengguna ..... 41
3.3.3 Objek Data dan Dokumen ..... 41
**BAB IV HASIL DAN PEMBAHASAN ..... 42**
4.1 Analisis Kebutuhan ..... 42
4.1.1 Analisis Proses Bisnis *Existing* ..... 42
4.1.2 Analisis Proses Bisnis *Future* ..... 43
4.1.3 Perbandingan *Existing* dan *Future* ..... 45
4.1.4 Identifikasi Permasalahan dan Tema Kebutuhan ..... 48
4.1.5 Analisis Pengguna Sistem ..... 50
4.1.6 Analisis KPI ..... 52

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

4.1.7 Analisis Kebutuhan Fungsional Sistem ..... 54
4.1.8 Analisis Kriteria Penerimaan Pengguna ..... 57
4.1.9 Analisis Kebutuhan Nonfungsional Sistem ..... 65
4.1.10 *Use Case Diagram* ..... 67
4.1.11 *Activity Diagram* ..... 69
4.2 Perancangan Sistem ..... 89
4.2.1 Perancangan Arsitektur Sistem ..... 89
4.2.2 Perancangan Hak Akses Pengguna ..... 92
4.2.3 Perancangan Basis Data ..... 94
4.2.4 Perancangan *Workflow* ..... 101
4.2.5 Perancangan API/*Endpoint* ..... 109
4.2.6 Perancangan Antarmuka Pengguna ..... 116
4.3 Implementasi Sistem ..... 133
4.3.1 Lingkungan Pengembangan ..... 134
4.3.2 Struktur Kode Project ..... 138
4.3.3 Implementasi *Backend* dan *Database* ..... 144
4.3.4 Implementasi Autentikasi dan Hak Akses ..... 148
4.3.5 Implementasi Modul Utama Sistem ..... 153
4.3.6 Implementasi Antarmuka Pengguna ..... 175
4.3.7 Implementasi Verifikasi Fungsional ..... 183
4.4 Pengujian dan Evaluasi Sistem ..... 188
4.4.1 Pendekatan dan Metode Pengujian ..... 189
4.4.2 Prosedur Pengujian ..... 191
4.4.3 *Black Box Testing* ..... 193
4.4.4 Skenario Pengujian Perhitungan KPI ..... 201
4.4.5 *User Acceptence Testing* ..... 204
4.4.6 *Usability Testing* ..... 208
4.4.7 Pengujian *Non-Functional Requirements* ..... 213
4.4.8 Hasil Analisis Pengujian ..... 218
4.4.9 Evaluasi terhadap Tujuan Penelitian ..... 220
**BAB V PENUTUP ..... 224**
5.1 Kesimpulan ..... 224
5.2 Saran ..... 224
**DAFTAR PUSTAKA ..... 225**

DAFTAR RIWAYAT HIDUP PENULIS ..... 228
LAMPIRAN ..... 229

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
   a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
   b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

# DAFTAR GAMBAR

Gambar 2. 1 Komponen Sistem Informasi ..... 6
Gambar 2. 2 Peringkat adjektif, Skor Acceptable dan Peringkat dalam SUS ..... 18
Gambar 3. 1 Diagram Tahapan Penelitian ..... 34
Gambar 3. 2 Flowchart Aplikasi Monitoring Proyek Berbasis KPI ..... 40
Gambar 4. 1 BPMN Proses Bisnis Existing ..... 43
Gambar 4. 2 BPMN Proses Bisnis Future ..... 44
Gambar 4. 3 Use Case Diagram Monitoring Proyek dan Penilaian KPI ..... 69
Gambar 4. 4 Activity Diagram Login ..... 71
Gambar 4. 5 Activity Diagram Menugaskan PM ke Project ..... 72
Gambar 4. 6 Activity Diagram Menugaskan Consultant ..... 73
Gambar 4. 7 Activity Diagram Mengelola Roster Consultant ..... 74
Gambar 4. 8 Activity Diagram Mengelola Status Project ..... 75
Gambar 4. 9 Activity Diagram Menyelesaikan Project ..... 76
Gambar 4. 10 Activity Diagram Memperbarui Status Milestone ..... 77
Gambar 4. 11 Activity Diagram Menilai Kualitas Milestone ..... 78
Gambar 4. 12 Activity Diagram Mengatur KPI ..... 79
Gambar 4. 13 Activity Diagram Memfinalisasi KPI Periode ..... 80
Gambar 4. 14 Activity Diagram Menghitung Ulang KPI ..... 81
Gambar 4. 15 Activity Diagram Melihat Daftar Project ..... 82
Gambar 4. 16 Activity Diagram Melihat Detail Project ..... 83
Gambar 4. 17 Activity Diagram Melihat Riwayat Lifecycle ..... 84
Gambar 4. 18 Activity Diagram Melihat Dashboard (per Peran) ..... 85
Gambar 4. 19 Activity Diagram Melihat Dashboard KPI ..... 86
Gambar 4. 20 Activity Diagram Mengelola Pengguna ..... 87
Gambar 4. 21 Activity Diagram Mengonfigurasi Sistem ..... 88
Gambar 4. 22 Arsitektur Sistem Monitoring Proyek dan Penilaian KPI ..... 90
Gambar 4. 23 Tabel Relasi Database Sistem ..... 98
Gambar 4. 24 State Diagram Lifecycle Proyek ..... 103
Gambar 4. 25 State Diagram Lifecycle Milestone ..... 106
Gambar 4. 26 Wireframe Halaman Project Detail ..... 124
Gambar 4. 27 Wireframe Halaman Dashboard CEO ..... 126

Gambar 4. 28 Wireframe Halaman KPI Snapshot per Consultant ..... 128
Gambar 4. 29 Wireframe Halaman KPI Configuration ..... 130
Gambar 4. 30 Wireframe Halaman Task Templates Management ..... 131
Gambar 4. 31 Implementasi Connection Pool MySQL pada db.js ..... 146
Gambar 4. 32 Implementasi Penerbitan JWT pada Proses Login ..... 150
Gambar 4. 33 Implementasi Middleware Autentikasi ..... 151
Gambar 4. 34 Penerapan Middleware Autentikasi dan Otorisasi pada Route Proyek ..... 152
Gambar 4. 35 Implementasi State Definitions dan Transition Matrix ..... 155
Gambar 4. 36 Kode Implementasi Fungsi transitionProject ..... 157
Gambar 4. 37 Kode Implementasi Precondition Gate ..... 159
Gambar 4. 38 Kode Implementasi Otorisasi Lapisan Kedua WfMS ..... 160
Gambar 4. 39 Kode Implementasi Audit Logger ..... 161
Gambar 4. 40 Kode Implementasi WFMSError dan Helper sendWFMSError ..... 163
Gambar 4. 41 Kode Implementasi Perhitungan Task Completion ..... 166
Gambar 4. 42 Kode Implementasi Perhitungan Timeliness ..... 167
Gambar 4. 43 Kode Implementasi Perhitungan Update Compliance ..... 168
Gambar 4. 44 Kode Implementasi Perhitungan Output Quality ..... 169
Gambar 4. 45 Kode Implementasi Weighted Scoring Method dan Validasi Bobot ..... 170
Gambar 4. 46 Kode Implementasi Persistensi Snapshot KPI ..... 171
Gambar 4. 47 Kode Implementasi Side Effects completeProject ..... 173
Gambar 4. 48 Halaman Project Detail dengan Tab Overview ..... 176
Gambar 4. 49 Halaman Dashboard CEO dengan Section Consultant KPI ..... 178
Gambar 4. 50 Halaman My KPI dengan Snapshot Empat Dimensi ..... 179
Gambar 4. 51 Halaman KPI Configuration dengan Dua Tingkat Change Management ..... 180
Gambar 4. 52 Halaman Task Templates Settings untuk Service Line Transfer Pricing ..... 182
Gambar 4. 53 Intepretasi Skor SUS Sistem ..... 212

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

# DAFTAR TABEL

Tabel 2. 1 Perbandingan RBAC vs ABAC vs ACL ..... 13
Tabel 2. 2 Perbandingan React.js vs Angular vs Vue.js ..... 19
Tabel 2. 3 Perbandingan Node.js vs Java Spring Boot vs Python vs Go ..... 21
Tabel 2. 4 Perbandingan MySQL vs PostgreSQL vs MongoDB ..... 22
Tabel 2. 5 Perbandingan RESTful API vs GraphQL vs gRPC ..... 24
Tabel 2. 6 Perbandingan Fetch API vs Axios vs XMLHttpRequest ..... 25
Tabel 2. 7 Penelitian Terdahulu ..... 26
Tabel 4. 1 Perbandingan Proses Bisnis Existing dan Future ..... 45
Tabel 4. 2 Pemetaan Tema Permasalahan ke Tema Kebutuhan ..... 48
Tabel 4. 3 Analisis Pengguna Sistem ..... 51
Tabel 4. 4 Daftar Dimensi KPI dan Bobotnya ..... 52
Tabel 4. 5 Daftar Kebutuhan Fungsional Sistem ..... 54
Tabel 4. 6 Kriteria Penerimaan Pengguna Sistem ..... 58
Tabel 4. 7 Daftar Kebutuhan Nonfungsional Sistem ..... 66
Tabel 4. 8 Matriks Hak Akses Pengguna per Peran ..... 93
Tabel 4. 9 Daftar Entitas Basis Data Sistem ..... 95
Tabel 4. 10 Daftar State Proyek pada Sistem ..... 102
Tabel 4. 11 Matrix Transisi Valid pada State Machine Proyek ..... 104
Tabel 4. 12 Daftar State Milestone pada Sistem ..... 105
Tabel 4. 13 Matrix Transisi Valid pada State Machine Milestone ..... 106
Tabel 4. 14 Pemetaan Kode Error State Machine ..... 109
Tabel 4. 15 Konvensi Penamaan Endpoint pada Sistem ..... 110
Tabel 4. 16 Struktur Standar Request dan Response ..... 111
Tabel 4. 17 Ringkasan Endpoint per Modul Fungsional ..... 112
Tabel 4. 18 Design Tokens Sistem ERP ..... 117
Tabel 4. 19 Pemetaan State terhadap Token Warna Status Badge ..... 120
Tabel 4. 20 Inventaris Halaman dalam Ruang Lingkup Penelitian ..... 121
Tabel 4. 21 Daftar Pustaka Utama Sistem ..... 134
Tabel 4. 22 Tools Pendukung Pengembangan ..... 136
Tabel 4. 23 Struktur Folder Package backend-development ..... 139
Tabel 4. 24 Struktur Folder Package frontend-development ..... 140

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tabel 4. 25 Pemetaan Layer Arsitektur dengan Lokasi Kode ..... 143
Tabel 4. 26 Middleware Stack pada app.js ..... 144
Tabel 4. 27 Migration File yang Relevan dengan Ruang Lingkup Penelitian ..... 147
Tabel 4. 28 Payload JWT Sistem ..... 149
Tabel 4. 29 Mode Evaluasi requirePermission ..... 151
Tabel 4. 30 Struktur Modul services/WfMS ..... 154
Tabel 4. 31 Struktur Modul services/kpi ..... 165
Tabel 4. 32 Penanganan Edge Case pada Perhitungan KPI ..... 174
Tabel 4. 33 Pemetaan Karakteristik Fungsional terhadap Implementasi ..... 183
Tabel 4. 34 Pemetaan Metode Pengujian terhadap Aspek Verifikasi ..... 189
Tabel 4. 35 Spesifikasi Lingkungan Pengujian ..... 192
Tabel 4. 36 Pemetaan Peran Tester pada Setiap Metode Pengujian ..... 193
Tabel 4. 37 Pelaksanaan dan Hasil Pengujian Black Box ..... 194
Tabel 4. 38 Parameter Konfigurasi Pengujian KPI ..... 201
Tabel 4. 39 Hasil Validasi Perhitungan KPI ..... 202
Tabel 4. 40 Profil Tester User Acceptance Testing ..... 205
Tabel 4. 41 Skenario dan Hasil User Acceptance Testing ..... 206
Tabel 4. 42 Pernyataan System Usability Scale yang Digunakan ..... 209
Tabel 4. 43 Interpretasi Skor SUS ..... 210
Tabel 4. 44 Task Usability dan Tingkat Keberhasilan ..... 211
Tabel 4. 45 Hasil System Usability Scale per Responden ..... 211
Tabel 4. 46 Hasil Pengujian Response Time ..... 213
Tabel 4. 47 Hasil Pengujian RBAC ..... 215
Tabel 4. 48 Hasil Pengujian Keandalan ..... 216
Tabel 4. 49 Hasil Pengujian Kompatibilitas Browser ..... 216
Tabel 4. 50 Hasil Pengujian Responsivitas ..... 217
Tabel 4. 51 Rekapitulasi Pengujian NFR ..... 217
Tabel 4. 52 Rekapitulasi Hasil Tiga Metode Pengujian ..... 219
Tabel 4. 53 Pemetaan Tujuan Penelitian terhadap Hasil Pengujian ..... 221

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

# BAB I
# PENDAHULUAN

## 1.1 Latar Belakang Masalah

PT DSK Global Konsultama merupakan perusahaan Konsultan yang menyediakan layanan profesional di bidang perpajakan, dengan fokus utama pada layanan *Transfer Pricing Documentation* (TP-Doc) yang mencakup penyusunan *Local File*, *Master File*, dan *Country-by-Country Report*. Berdasarkan hasil observasi dan wawancara dengan Direksi dan Project Manager (Lampiran 1 dan Lampiran 2), *monitoring* proyek TP-Doc saat ini dilakukan secara manual melalui kombinasi tiga media yang tidak terintegrasi, yaitu pencatatan progres di Excel pribadi Project Manager, koordinasi tim melalui WhatsApp, dan pelaporan status ke Direksi melalui *email* atau pertemuan tatap muka. Akibatnya, informasi progres, status proyek, dan tenggat waktu tersebar di berbagai media sehingga sulit diakses secara cepat ketika dibutuhkan untuk pengambilan keputusan. Penilaian kinerja karyawan juga belum optimal karena bobot *Key Performance Indicator* (KPI) belum ditetapkan melalui mekanisme yang konsisten dan terdokumentasi, sehingga penilaian cenderung subjektif dan menyulitkan manajemen mengidentifikasi akar permasalahan kinerja secara objektif (Rizana et al., 2023).

Permasalahan tersebut berdampak langsung pada ketepatan penyelesaian proyek TP-Doc. Project Manager menjelaskan bahwa dalam satu tahun terakhir perusahaan menerima sekitar 110 klien untuk layanan TP-Doc, namun yang benar-benar masuk tahap eksekusi proyek sampai dengan selesai hanya sekitar 50 proyek per tahun 2025. Dari jumlah tersebut, sekitar 12 sampai 15 proyek mengalami keterlambatan penyelesaian proyek dengan akar penyebab yang beragam. Sebanyak 5 proyek di antaranya merupakan keterlambatan murni dari sisi Consultant internal dengan keterlambatan terparah mencapai dua minggu dari yang sudah dijadwalkan, sementara sisanya disebabkan oleh keterlambatan respons klien dalam memberikan revisi atau melengkapi dokumen. Karena keterlambatan yang tersebar di berbagai penyebab, maka dibutuhkan sistem *monitoring* berbasis *workflow* yang mampu mengidentifikasi akar penyebab keterlambatan secara objektif, sehingga manajemen dapat melakukan secara tepat sasaran (Agustha et al., 2024).

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Di sisi penilaian kinerja, indikator yang digunakan masih bersifat umum, belum terdokumentasi dengan teratur per posisi, dan tidak digunakan secara konsisten setiap periode. Bobot KPI belum baku, data realisasi pekerjaan belum terkumpul secara otomatis, dan proses evaluasi masih bergantung pada rekapitulasi manual Project Manager dari ketiga media pencatatan tersebut. Dalam konteks transformasi digital, praktik *performance management* dituntut menjadi lebih fleksibel dengan memanfaatkan sistem informasi untuk memantau progres pekerjaan dan capaian target secara cepat (Cosa and Torelli, 2024). Penerapan *Electronic Performance Monitoring* (EPM) menjadi relevan karena mampu meningkatkan visibilitas dan kontrol kinerja, meskipun tetap memerlukan perancangan sistem yang adil dan sebanding (Kalischko and Riedl, 2021). Pengukuran kinerja karyawan berbasis KPI menjadi kebutuhan strategis agar kontribusi individu maupun tim dapat dinilai secara objektif dan konsisten.

Berdasarkan uraian tersebut, permasalahan *monitoring* proyek dan penilaian kinerja karyawan di PT DSK Global Konsultama saling berkaitan. Oleh karena itu, perusahaan membutuhkan sistem *monitoring* proyek berbasis *workflow* yang terintegrasi dengan modul penilaian kinerja berbasis KPI, sehingga perhitungan kinerja dapat memanfaatkan data realisasi aktivitas proyek seperti ketepatan waktu penyelesaian tugas, kedisiplinan pembaruan progres, dan kualitas hasil pekerjaan. Penelitian ini berfokus pada pengembangan aplikasi tersebut sebagai studi kasus, yang didukung *dashboard* informatif untuk mendukung pengambilan keputusan manajerial.

## 1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah dalam penelitian ini adalah: “Bagaimana merancang dan membangun aplikasi *monitoring* proyek berbasis *workflow* yang terintegrasi dengan penilaian kinerja karyawan berbasis *Key Performance Indicator* (KPI) pada PT DSK Global Konsultama?”

## 1.3 Batasan Masalah

Agar penelitian lebih terarah dan tidak melebar, maka batasan masalah dalam penelitian ini adalah sebagai berikut:

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

1. Penelitian ini difokuskan pada pengembangan aplikasi *monitoring* proyek berbasis *workflow* dan penilaian kinerja karyawan berbasis Key Performance Indicator (KPI) pada PT DSK Global Konsultama dengan layanan Transfer Pricing Documentation (TP-Doc) sebagai studi kasus utama.
2. Sistem yang dikembangkan hanya mencakup proses *monitoring* proyek internal perusahaan untuk layanan TP-Doc, mulai dari penugasan, pelaksanaan, pembaruan progres, hingga pelaporan status proyek berdasarkan alur *workflow* yang telah ditetapkan.
3. Aktor sistem dibatasi pada pihak yang terlibat langsung dalam proses bisnis penelitian, yaitu CEO, COO, Project Manager, Consultant, dan Superadmin sesuai hak akses masing-masing.
4. Penilaian kinerja karyawan dibatasi pada indikator KPI yang berkaitan langsung dengan aktivitas proyek, seperti ketepatan penyelesaian tugas atau *milestone*, kepatuhan pembaruan progres, dan kualitas *output* kerja, serta tidak membahas aspek penilaian non-kinerja seperti kepribadian atau faktor psikologis.
5. Penetapan indikator, target, bobot, dan sumber data KPI dibatasi pada kebutuhan perusahaan yang diperoleh melalui observasi, wawancara, dan dokumen pendukung yang tersedia.
6. Penelitian ini berfokus pada perancangan, pembangunan, dan pengujian fungsional sistem, serta tidak membahas pengukuran dampak jangka panjang terhadap produktivitas organisasi secara keseluruhan.
7. Sistem yang dikembangkan berfokus pada aspek fungsional aplikasi berbasis web dan *dashboard* informasi, tanpa membahas integrasi dengan sistem pihak ketiga di luar ruang lingkup penelitian.

## 1.4 Tujuan dan Manfaat

Tujuan dan manfaat dari penelitian ini adalah sebagai berikut.

### 1.4.1 Tujuan Penelitian

Penelitian ini bertujuan merancang dan membangun aplikasi *monitoring* proyek berbasis *workflow* yang terintegrasi dengan penilaian kinerja karyawan berbasis Key Performance Indicator (KPI). Pada sisi *monitoring*, aplikasi dirancang agar

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

pemantauan progres proyek dapat dilakukan secara terpusat melalui alur kerja yang baku dan terdokumentasi. Pada sisi penilaian kinerja, aplikasi memanfaatkan data realisasi aktivitas proyek sebagai dasar perhitungan KPI agar penilaian menjadi lebih objektif dan dapat ditelusuri sumber datanya. Hasil *monitoring* dan penilaian kemudian disajikan dalam bentuk *dashboard* manajerial sebagai dukungan pengambilan keputusan oleh direksi.

### 1.4.2 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Bagi perusahaan, sistem menyediakan pencatatan status proyek dan *milestone* secara otomatis dalam *state* yang baku, sehingga keterlambatan dapat terdeteksi lebih dini dan identifikasi *bottleneck* dapat dilakukan berdasarkan data, bukan komunikasi informal.
2. Pada sisi penilaian kinerja, integrasi dengan *workflow* akan menghasilkan perhitungan KPI yang dilakukan secara otomatis dari data realisasi aktivitas proyek tanpa *input* ulang manual, sehingga hasil penilaian lebih objektif dan dapat ditelusuri sumbernya.
3. Bagi karyawan, ketersediaan indikator kinerja dan dasar penilaian yang terdokumentasi memberikan transparansi dalam proses evaluasi sehingga dapat dijadikan acuan perbaikan kinerja secara individu.
4. Bagi penulis, penelitian ini menjadi sarana penerapan ilmu rekayasa perangkat lunak dan pengembangan sistem informasi dalam menyelesaikan permasalahan nyata di industri konsultasi perpajakan.
5. Bagi pengembangan ilmu pengetahuan, penelitian ini dapat menjadi referensi bagi penelitian selanjutnya yang berkaitan dengan integrasi *monitoring* proyek dan penilaian kinerja berbasis KPI pada organisasi jasa konsultasi maupun sektor lainnya.

## 1.5 Sistematika Penulisan

Sistematika penulisan pada penelitian ini adalah sebagai berikut:

a. BAB 1 PENDAHULUAN, berisi latar belakang masalah, perumusan masalah, batasan masalah, tujuan dan manfaat penelitian, serta sistematika penulisan.

b. BAB 2 TINJAUAN PUSTAKA, berisi landasan teori dan penelitian terdahulu yang relevan dengan topik *monitoring* proyek, *workflow management system*, penilaian kinerja, Key Performance Indicator (KPI), *dashboard*, kontrol akses, serta teknologi yang digunakan dalam pengembangan sistem.

c. BAB 3 METODE PENELITIAN, berisi rancangan penelitian, tahapan penelitian, dan objek penelitian.

d. BAB 4 HASIL DAN PEMBAHASAN, berisi analisis kebutuhan, perancangan sistem, implementasi sistem, dan pengujian dan evaluasi terhadap sistem yang dikembangkan.

e. BAB 5 PENUTUP, berisi kesimpulan dari hasil penelitian serta saran untuk pengembangan lebih lanjut.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

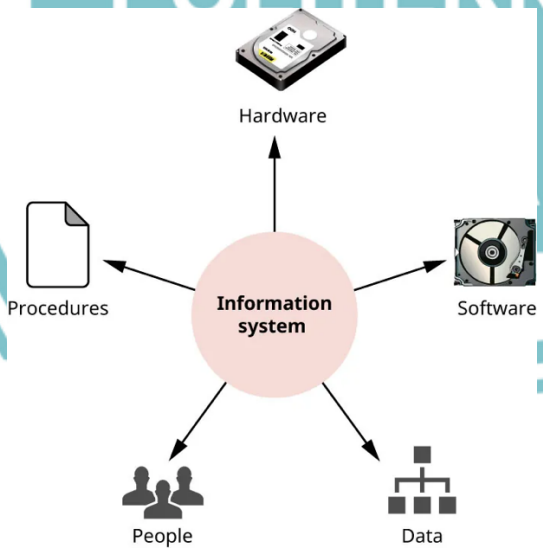
# BAB II
# TINJAUAN PUSTAKA

## 2.1 Kajian Teoretis

Kajian teoritis pada penelitian ini disusun sebagai landasan konseptual dalam perancangan sistem *monitoring* proyek berbasis *workflow* dan penilaian kinerja karyawan menggunakan *Key Performance Indicator* (KPI).

### 2.1.1 Sistem Informasi

Sistem informasi adalah kumpulan komponen yang saling terkait untuk mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi guna mendukung pengambilan keputusan, koordinasi, dan pengendalian organisasi. Pada Gambar 2.1 menunjukkan bahwa komponen sistem informasi mencakup perangkat keras, perangkat lunak, data, pengguna serta prosedur yang dapat bekerja terpadu agar menghasilkan informasi yang relevan dan berkualitas (Soufitri, 2023). Dalam konteks organisasi modern, sistem informasi tidak hanya berfungsi sebagai sarana pencatatan data operasional, tetapi juga sebagai infrastruktur penting yang dapat menopang proses bisnis *end-to-end* serta menjadi sumber data untuk pengambilan keputusan strategis manajemen (Inayah et al., 2024).




Gambar 2. 1 Komponen Sistem Informasi

Sumber: openstax.org, 2024

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Sistem informasi juga menjadi sarana pengelolaan operasi, penyusunan strategi, dan analisis kinerja secara konsisten serta akurat, di mana kombinasi teknologi, manusia, dan proses kerja berkontribusi pada tersedianya data yang relevan untuk mendukung tujuan organisasi (Hananiya et al., 2024). Seiring kebutuhan akses dan integrasi yang lebih luas, sistem informasi modern banyak diimplementasikan dalam bentuk berbasis *website* agar dapat diakses melalui *browser* secara *real-time* melalui jaringan internet, memungkinkan pengguna mengelola informasi dari berbagai lokasi. Platform *website* mengintegrasikan data dan fungsi aplikasi secara terpusat sehingga proses bisnis seperti pengelolaan data, pemantauan status, dan pelaporan menjadi lebih efektif dibanding cara manual, sekaligus meningkatkan akurasi data dan kemudahan akses informasi bagi pengguna (Utama, 2022).

Selain itu, arsitektur klien-*server* memberi fleksibilitas dan skalabilitas untuk melayani banyak pengguna sekaligus. Implementasi sistem berbasis *website* terbukti membantu integrasi modul layanan, mengurangi duplikasi data, dan mempercepat respons layanan, sehingga pendekatan ini menjadi relevan untuk mendukung *monitoring* proyek dan evaluasi kinerja di perusahaan melalui konsep manajemen proyek (Akbartio et al., 2025).

### 2.1.2 Manajemen dan *Monitoring* Proyek

Manajemen proyek mencakup perencanaan, pengorganisasian, pengendalian, dan pengawasan sumber daya agar tujuan proyek tercapai sesuai waktu, biaya, dan kualitas yang ditetapkan. Dalam konteks sistem informasi, manajemen proyek membutuhkan koordinasi informasi yang efektif untuk mendukung keputusan pada tiap tahap. Sistem informasi manajemen proyek berbasis *website* dapat memfasilitasi perencanaan, pelaksanaan, pemantauan, hingga penyelesaian proyek secara terintegrasi dan *real-time* (Barus and Yahfizham, 2024). Keberhasilan proyek bergantung pada pengelolaan waktu, biaya, risiko, serta komunikasi pemangku kepentingan secara sistematis, di mana dukungan sistem informasi membantu menekan keterlambatan, pembengkakan biaya, dan konflik sumber daya yang umum terjadi pada proyek kompleks (Hermawan et al., 2023).

Sebagai bagian penting dari manajemen proyek, *monitoring* proyek merupakan pengawasan sistematis atas perkembangan pekerjaan agar kegiatan berjalan sesuai

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

rencana, jadwal, dan target, sehingga risiko keterlambatan serta ketidaksesuaian dapat diminimalkan. Dalam sistem informasi modern, *monitoring* proyek dilakukan melalui aplikasi berbasis *website* yang mengumpulkan, menyimpan, dan menampilkan progres pekerjaan secara *real-time*, sehingga keputusan dapat diambil secara cepat dan akurat berdasarkan data terkini. Pengembangan aplikasi *monitoring* proyek berbasis *website* mampu meningkatkan transparansi pelaporan progres dan mempercepat identifikasi keterlambatan pekerjaan (Adieb et al., 2024). Supervisor proyek yang masih mengandalkan WhatsApp dan *email* menghasilkan komunikasi terfragmentasi, keterbatasan mengidentifikasi progres, serta dokumentasi yang tidak terstruktur, sehingga digitalisasi *monitoring* proyek menjadi kebutuhan mendesak bagi organisasi yang ingin meningkatkan akuntabilitas operasional (Musva and Ramadhani, 2025).

Sistem *monitoring* terkomputerisasi juga mencatat histori proyek, memudahkan pencarian data, dan mengurangi ketergantungan pada dokumen fisik. Fitur seperti *dashboard* visual dan integrasi database meningkatkan transparansi serta kolaborasi antar tim dan pemangku kepentingan, sehingga pemantauan menjadi lebih efisien dan kualitas keputusan manajemen meningkat (Nurlisa, 2021). Untuk memastikan alur kerja berjalan terstruktur dan konsisten, *monitoring* proyek perlu didukung oleh konsep *Workflow Management System* (WfMS).

### 2.1.3 *Workflow Management System*

*Workflow Management System* (WfMS) merupakan perangkat lunak yang menyediakan infrastruktur untuk mendefinisikan, mengoordinasikan, dan memantau eksekusi serangkaian tugas yang membentuk suatu proses bisnis. Berbeda dari sekadar alat otomatis tugas individual, WfMS berperan sebagai lapisan kendali yang memastikan setiap tahapan pekerjaan berjalan sesuai aturan proses bisnis yang telah didefinisikan. Fungsi inti WfMS mencakup tiga aspek utama, yaitu pemodelan proses, eksekusi proses, dan pemantauan eksekusi (Ochoa et al., 2023).

Penerapan WfMS dalam organisasi terbukti memberikan dampak positif terhadap efisiensi operasional, transparansi proses, serta kepatuhan terhadap prosedur kerja yang telah disepakati. Standardisasi proses bisnis melalui WfMS membantu

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

organisasi mengurangi variasi pelaksanaan pekerjaan antar individu, sekaligus menyediakan basis data yang konsisten untuk keperluan evaluasi kinerja (Goel et al., 2023). Menempatkan WfMS sebagai komponen fundamental dalam transformasi digital proses bisnis, di mana otomasi alur kerja tidak hanya menggantikan tugas manual, tetapi juga membuka peluang bagi pengambilan keputusan berbasis data yang lebih akurat, sehingga setiap tahapan pekerjaan dapat dipantau, ditelusuri, dan dikendalikan secara lebih terstruktur (Rosemann et al., 2024).

### 2.1.4 Manajemen Penilaian Kinerja Karyawan

Manajemen penilaian kinerja karyawan adalah proses organisasi untuk mengukur, dan mengevaluasi kinerja individu maupun tim guna mencapai tujuan strategis dan meningkatkan produktivitas berkelanjutan. Proses ini meliputi penetapan tujuan, pemantauan berkala, penilaian hasil, serta tindak lanjut berupa umpan balik dan pengembangan kapabilitas. Dengan demikian, kontribusi karyawan dapat dinilai lebih objektif dan mendukung budaya kerja yang berorientasi perbaikan (Zulfikar, 2024). Dengan dukungan teknologi informasi, sistem *monitoring* kinerja berbasis *website* dapat mempercepat evaluasi dan mengurangi subjektivitas sehingga transparansi serta akurasi data meningkat. Karena itu, manajemen kinerja menjadi komponen penting untuk menopang evaluasi KPI (Rizal and Hidayatullah, 2025).

### 2.1.5 *Key Performance Indicator*

*Key Performance Indicator* (KPI) merupakan ukuran yang digunakan untuk menilai tingkat pencapaian tujuan strategis organisasi, sehingga kinerja individu, tim, maupun proses dapat diukur secara objektif dan konsisten. KPI berfungsi sebagai alat evaluasi yang membantu organisasi mengetahui sejauh mana target yang telah ditetapkan dapat dicapai dalam periode tertentu. Pengukuran kinerja yang baik perlu mempertimbangkan indikator yang mampu merepresentasikan kontribusi sumber daya manusia secara terukur, baik dari aspek hasil kerja maupun proses kerja (Karlina and Samanhudi, 2023).

KPI memiliki beberapa karakteristik utama, yaitu spesifik, terukur, relevan, memiliki target yang jelas, serta ditetapkan dalam periode pengukuran tertentu (kpi.org, 2026). Selain itu, KPI harus dapat dipahami oleh pihak yang dinilai dan

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

didukung oleh sumber data yang valid agar hasil evaluasi dapat dipertanggungjawabkan. Dalam operasionalnya, perancangan KPI mencakup beberapa komponen pokok, yaitu definisi indikator, target, bobot, sumber data, periode penilaian, dan mekanisme pelaporan. Definisi indikator menjelaskan aspek kinerja yang diukur, target menunjukkan capaian yang diharapkan, bobot menunjukkan tingkat kepentingan masing-masing indikator, sedangkan sumber data menjelaskan asal data yang digunakan untuk menghitung nilai indikator. Dengan komponen tersebut, KPI tidak hanya berfungsi sebagai alat penilaian, tetapi juga sebagai pedoman kerja bagi karyawan agar memahami ekspektasi organisasi secara lebih jelas.

### 2.1.6 *Weighted Scoring Method* (WSM)

*Weighted Scoring Method* merupakan salah satu metode pengambilan keputusan multi-kriteria (*Multi-Criteria Decision Making*) yang paling sederhana dan banyak digunakan dalam sistem penilaian berbasis indikator. Metode ini menghitung skor total dari sebuah penilaian, dalam konteks penelitian ini berupa kinerja karyawan, dengan menjumlahkan hasil perkalian antara skor pencapaian masing-masing kriteria dengan bobotnya. Hasil perhitungan WSM menghasilkan skor tunggal pada skala tertentu yang merepresentasikan kinerja keseluruhan, sehingga memudahkan interpretasi dan perbandingan antar individu maupun antar periode penilaian.

WSM dipilih sebagai metode perhitungan KPI dalam penelitian ini karena beberapa pertimbangan yaitu, kesederhanaan komputasi yang mendukung perhitungan otomatis secara *real-time* pada sistem yang dikembangkan tanpa membutuhkan algoritma kompleks, interpretasi hasil yang dapat memudahkan pengguna dari berbagai peran memahami skor kinerjanya, serta fleksibilitas dalam penyesuaian bobot yang memungkinkan CEO sebagai pemegang kewenangan pengelolaan KPI mengubah prioritas penilaian ketika strategi perusahaan berubah, tanpa harus mengubah arsitektur sistem.

Perhitungan capaian KPI per indikator pada metode WSM umumnya menggunakan persentase capaian. Untuk indikator yang bersifat *benefit* atau semakin besar nilainya semakin baik, capaian KPI dihitung dengan rumus:

$$Capaian\ KPI\ (Benefit) = \frac{Realisasi}{Target} \times 100\% \quad (1)$$

Sedangkan untuk indikator yang bersifat *cost* atau semakin kecil nilainya semakin baik, capaian KPI dihitung dengan rumus:

$$Capaian\ KPI\ (Cost) = \frac{Target}{Realisasi} \times 100\% \quad (2)$$

Apabila penilaian kinerja terdiri atas beberapa indikator, maka skor total KPI dihitung menggunakan rumus inti dengan pembobotan sebagai berikut:

$$Skor\ KPI_{total} = \sum_{i=1}^{n} (w_i \times c_i) \quad (3)$$

dengan ketentuan total bobot seluruh indikator sama dengan 1 atau 100%, yaitu:

$$\sum_{i=1}^{n} w_i = 100\% \quad (4)$$

Apabila bobot belum dinormalisasi, maka skor total KPI dapat dihitung dengan rumus:

$$KPI_{total} = \frac{\sum_{i=1}^{n} (b_i \times c_i)}{\sum_{i=1}^{n} b_i} \quad (5)$$

Keterangan:

- $Capaian\ KPI$ = persentase capaian indikator kinerja
- $KPI_{total}$ = skor total Key Performance Indicator
- $w_i$= bobot indikator ke-i yang telah dinormalisasi
- $b_i$ = bobot indikator ke-i yang belum dinormalisasi
- $c_i$ = persentase capaian indikator ke-i
- $n$ = jumlah indikator yang digunakan dalam penilaian

### 2.1.7 Hubungan *Monitoring* Proyek dengan Penilaian Kinerja

*Monitoring* proyek yang terintegrasi dengan penilaian kinerja karyawan dapat menghasilkan data yang lebih objektif dalam menghitung KPI. *Monitoring* yang



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

terstruktur menghasilkan informasi progres, tenggat waktu, dan realisasi *output* sehingga penilaian kinerja menjadi lebih akurat, transparan, dan dapat dipertanggungjawabkan. Penerapan KPI tidak berdiri sendiri, tetapi dirancang untuk terhubung langsung dengan data operasional proyek yang dihasilkan oleh sistem *monitoring*. Penerapan KPI yang ditopang oleh sistem pencatatan data yang terstruktur dapat meningkatkan optimalisasi penilaian kinerja karyawan secara signifikan, sekaligus mengurangi subjektivitas evaluasi yang sering muncul ketika penilaian dilakukan tanpa basis data yang jelas (Hersusetiyati et al., 2024). Data progres dan realisasi yang dikumpulkan secara berkala melalui sistem *monitoring* tidak hanya berfungsi sebagai alat pengawasan operasional, tetapi juga menjadi pengukuran kinerja yang lebih objektif bagi organisasi. Dengan integrasi keduanya, akuntabilitas dan efisiensi organisasi diperkuat karena proses operasional dan evaluasi performa saling melengkapi.

### 2.1.8 *Dashboard* dan Visual Data

*Dashboard* adalah antarmuka visual yang menyajikan data dalam tampilan ringkas agar pengguna cepat memahami informasi, kondisi, tren, serta deviasi dari target. Melalui visualisasi seperti grafik, *gauge*, dan tabel, *dashboard* memudahkan identifikasi anomali dan prioritas tindakan secara *real-time*. *Dashboard* yang dirancang baik dengan integrasi data dan visualisasi yang tepat akan mempercepat pengambilan keputusan dan meningkatkan respons organisasi (Gonçalves et al., 2023). Prinsip *user-centered design* dan metrik usabilitas juga menjadi acuan evaluasi agar *dashboard* efektif mendukung keputusan.

### 2.1.9 *Role-Based* Access Control (RBAC)

*Role-Based Access Control* (RBAC) adalah pendekatan kontrol akses yang membuat keputusan otorisasi berdasarkan peran (*role*) pengguna dalam organisasi, bukan berdasarkan identitas individual semata, sehingga mekanisme ini lebih selaras dengan struktur pekerjaan dan lebih mudah dikelola pada skala organisasi yang kompleks. RBAC mendukung penerapan prinsip *least privilege* dengan memberikan pengguna hanya hak akses minimum yang diperlukan untuk menjalankan tugasnya, sehingga risiko keamanan dapat ditekan secara signifikan (Prasetia and Manongga, 2024).

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Secara model, literatur RBAC arus utama menjelaskan pola relasi inti antara *user*, *role*, dan *permission*, yang dilengkapi dengan konsep sesi untuk mengaktifkan peran tertentu saat *runtime*. Mengimplementasikan RBAC pada sistem berbasis *cloud* dengan membentuk hirarki peran yang merepresentasikan struktur organisasi, sehingga manajemen hak akses dapat dilakukan secara terpusat, ditinjau ulang, dan diaudit secara sistematis (Yuricha and Phan, 2023). Integrasi RBAC terbukti efektif memperkuat keamanan aplikasi, terutama dalam membatasi akses fitur sensitif berdasarkan peran pengguna, sehingga prinsip *separation of concern* dapat diterapkan secara konsisten di seluruh lapisan aplikasi (Mary and Febriyani, 2025).

Tabel 2. 1 Perbandingan RBAC vs ABAC vs ACL

| Kriteria | RBAC | ABAC | ACL |
|---|---|---|---|
| Basis Keputusan Akses | Peran yang melekat pada pengguna | Atribut pengguna, sumber daya, dan konteks | Identitas pengguna pada tiap objek secara individual |
| Tingkat Detail Kontrol | Sedang, dikelompokkan per peran | Sangat detail, dapat dikombinasikan per atribut | Sangat detail, namun dikelola per objek individual |
| Skalabilitas pada Organisasi Bertingkat | Sangat baik, penambahan pengguna cukup dipetakan ke peran yang sudah ada | Baik, namun kompleksitas kebijakan meningkat seiring jumlah atribut | Buruk pada organisasi besar karena data akses membengkak |
| Kompleksitas Pengelolaan | Rendah, peran dapat dipetakan langsung ke jabatan organisasi | Tinggi, perlu keahlian khusus untuk merancang kebijakan atribut | Rendah pada sistem kecil, tinggi pada sistem |

| Kriteria | RBAC | ABAC | ACL |
|---|---|---|---|
| | | | dengan banyak objek |
| Kesesuaian dengan Struktur Jabatan | Sangat sesuai karena peran selaras dengan jabatan formal | Kurang natural pada organisasi dengan struktur jabatan jelas | Kurang natural karena akses dikelola per objek, bukan per peran |

Berdasarkan perbandingan pada Tabel 2.1, RBAC dipilih sebagai mekanisme kontrol akses pada sistem karena paling sesuai dengan karakteristik organisasi perusahaan. Struktur jabatan perusahaan yang jelas dan bertingkat seperti CEO, COO, Project Manager, Consultant, dan Superadmin yang dapat dipetakan langsung ke peran dalam sistem tanpa konfigurasi tambahan. Dibandingkan ABAC yang memerlukan keahlian khusus dalam merancang kebijakan berbasis atribut, dan ACL yang tidak praktis pada organisasi dengan banyak pengguna dan data, RBAC menawarkan keseimbangan terbaik antara kemudahan pengelolaan, skalabilitas, dan kesesuaian dengan struktur organisasi yang ada.

### 2.1.10 *Business Process Model and Notation* (BPMN)

Business Process Model and Notation (BPMN) merupakan notasi grafis standar untuk memodelkan proses bisnis yang dikelola oleh Object Management Group. BPMN dirancang agar dapat dipahami oleh seluruh pemangku kepentingan, baik pengguna bisnis maupun pengguna teknis, sehingga notasi ini berfungsi sebagai bahasa bersama yang menjembatani perancangan proses bisnis dengan implementasinya pada sistem informasi. Elemen pemodelan pada BPMN dikelompokkan ke dalam beberapa kategori utama, yaitu *flow object* yang terdiri atas *event*, *activity*, dan *gateway*, *swimlane* yang terdiri atas *pool* dan *lane*, serta *connecting object* yang terdiri atas *sequence flow*, *message flow*, dan *association*. Melalui elemen-elemen tersebut, BPMN dapat digunakan untuk memetakan proses bisnis yang sedang berjalan sekaligus merancang usulan perbaikannya, sehingga



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

perbedaan antara kondisi berjalan dan kondisi usulan dapat diidentifikasi secara jelas (Haerudin and Devianto, 2023).

Dalam penelitian ini, BPMN digunakan untuk memodelkan proses bisnis *monitoring* proyek pada kondisi berjalan (*existing*) dan kondisi usulan (*future*). Pemilihan BPMN didasarkan pada kemampuannya merepresentasikan pembagian tanggung jawab antar peran melalui *pool* dan *lane*, yang sesuai dengan karakteristik proses bisnis perusahaan yang melibatkan lima peran berbeda.

### 2.1.11 *Use Case Diagram*

*Use Case Diagram* merupakan salah satu diagram dalam *Unified Modeling Language* (UML) yang menggambarkan kebutuhan fungsional sistem dari sudut pandang pengguna. Diagram ini memetakan interaksi antara aktor, yaitu pihak di luar batas sistem yang berinteraksi dengannya, dengan *use case*, yaitu unit fungsionalitas yang memberikan nilai bagi aktor tersebut. Komponen utama *Use Case Diagram* meliputi aktor, *use case* yang digambarkan sebagai elips, batas sistem (*system boundary*) yang memisahkan lingkup sistem dari lingkungannya, serta relasi antar elemen berupa asosiasi, *include*, dan *extend*. Pemodelan menggunakan UML yang mencakup *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram* terbukti mampu memvisualisasikan kebutuhan sistem *monitoring* dan evaluasi berdasarkan hak akses masing-masing pengguna, sehingga rancangan yang dihasilkan dapat dijadikan dasar pengembangan sistem (Anardani et al., 2023).

Pada penelitian ini, *Use Case Diagram* digunakan untuk memetakan kebutuhan fungsional sistem terhadap lima aktor, yaitu CEO, COO, Project Manager, Consultant, dan Superadmin, sesuai kewenangan masing-masing peran. Pemetaan tersebut menjadi dasar penyusunan daftar kebutuhan fungsional sekaligus rancangan kontrol akses berbasis peran pada sistem.

### 2.1.12 *Activity Diagram*

*Activity Diagram* merupakan diagram dalam UML yang menggambarkan alur kerja atau urutan aktivitas dalam suatu proses. Berbeda dengan *Use Case Diagram* yang menjawab pertanyaan mengenai apa yang dapat dilakukan sistem, *Activity Diagram*

menguraikan bagaimana setiap fungsi tersebut dijalankan langkah demi langkah, sehingga diagram ini umum digunakan untuk memerinci skenario dari sebuah *use case*. Elemen utama *Activity Diagram* mencakup *initial node* sebagai titik awal, *action* yang merepresentasikan langkah pekerjaan, *decision node* yang menyatakan percabangan berdasarkan kondisi tertentu, *merge node* yang menyatukan kembali cabang alur, *final node* sebagai titik akhir, serta *swimlane* yang memisahkan aktivitas berdasarkan pelaku yang bertanggung jawab. Implementasi *Activity Diagram* bersama *Use Case Diagram* dalam perancangan aplikasi memungkinkan setiap rangkaian langkah proses tergambar secara bertahap, termasuk logika percabangan yang terjadi di dalamnya (Pebriad et al., 2023).

Dalam penelitian ini, *Activity Diagram* digunakan untuk memerinci alur setiap *use case* yang telah dipetakan sebelumnya, dengan pemisahan *swimlane* antara pengguna dan sistem. Perincian tersebut menjadi dasar penetapan validasi, prasyarat, dan transisi status yang selanjutnya diimplementasikan dalam modul *Workflow Management System*.

### 2.1.13 *Black Box Testing*

*Black Box Testing* adalah teknik pengujian perangkat lunak yang memverifikasi fungsionalitas sistem berdasarkan spesifikasi kebutuhan tanpa memperhatikan struktur internal kode program. Dalam pendekatan ini, penguji memperlakukan sistem sebagai kotak hitam yang hanya diamati dari sisi masukan dan keluaran, sehingga pengujian difokuskan pada apa yang dilakukan sistem, bukan bagaimana sistem melakukannya. *Black Box Testing* dapat diterapkan pada seluruh tingkatan pengujian mulai dari *unit*, integrasi, sistem, hingga penerimaan (ISTQB, 2024).

Dua teknik *black box testing* yang paling umum digunakan untuk perancangan *test case* adalah *Equivalence Partitioning* (EP) dan *Boundary Value Analysis* (BVA). EP membagi domain masukan ke dalam partisi-partisi ekuivalensi di mana seluruh nilai dalam satu partisi diharapkan menghasilkan perilaku sistem yang sama, sehingga cukup satu nilai representatif dari setiap partisi yang diuji untuk mewakili keseluruhan partisi tersebut. Teknik ini mengurangi jumlah *test case* dari tak terbatas menjadi jumlah yang terbatas dan dapat dikelola, namun tetap mencakup cakupan kebutuhan yang maksimal. BVA merupakan perluasan dari EP yang secara



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

khusus menguji nilai-nilai di batas antar partisi, karena pengalaman empiris menunjukkan bahwa kesalahan program lebih sering terjadi pada batas partisi dibandingkan pada nilai-nilai di tengah partisi. Kombinasi kedua teknik ini menghasilkan seperangkat *test case* yang efisien dan memiliki kemampuan deteksi *defect* yang tinggi (Santi et al., 2022).

### 2.1.14 *User Acceptance Testing*

*User Acceptance Testing* (UAT) adalah tahap pengujian terakhir dalam siklus pengembangan perangkat lunak yang dilakukan oleh pengguna atau perwakilan pengguna untuk memvalidasi apakah sistem yang dikembangkan memenuhi kebutuhan bisnis dan dapat diterima untuk digunakan secara operasional. Berbeda dari pengujian teknis seperti *unit testing* atau *integration testing* yang berfokus pada kebenaran kode dan keterpaduan komponen, UAT berfokus pada apakah perilaku sistem sesuai dengan proses bisnis dan harapan pengguna dalam konteks nyata. Pengujian dilakukan berdasarkan kriteria penerimaan yang telah disepakati sebelumnya, dan hasil pengujian menentukan apakah sistem dapat dirilis untuk digunakan (ISTQB, 2024).

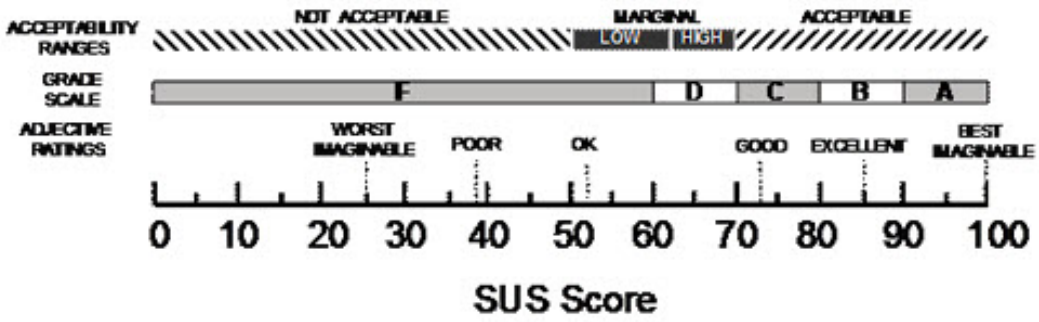
Pelaksanaan UAT umumnya mengikuti skenario yang merepresentasikan alur kerja nyata pengguna dari awal hingga akhir. Setiap skenario terdiri dari serangkaian langkah yang mencerminkan aktivitas operasional sehari-hari, dan masing-masing dikaitkan dengan satu atau lebih kriteria penerimaan yang terukur. Penguji yang terlibat dalam UAT idealnya adalah pengguna yang akan mengoperasikan sistem dalam lingkungan produksi, sehingga penilaian yang diberikan mencerminkan kesesuaian sistem dengan kebutuhan aktual pengguna. Keluaran UAT berupa keputusan penerimaan, apakah skenario diterima, ditolak, atau diterima dengan catatan (ISO/IEC/IEEE, 2022).

### 2.1.15 *Usability Testing* dan *System Usability Scale*

*Usability testing* adalah metode evaluasi yang mengukur sejauh mana pengguna dapat menggunakan sistem untuk mencapai tujuan tertentu secara efektif, efisien, dan memuaskan dalam konteks penggunaan yang ditentukan. Definisi ini mengacu pada standar ISO 9241-11 yang menetapkan tiga dimensi *usability*, yaitu efektivitas yang diukur dari keberhasilan pengguna menyelesaikan tugas, efisiensi yang diukur

dari sumber daya yang dikeluarkan untuk menyelesaikan tugas, dan kepuasan yang diukur dari persepsi subjektif pengguna terhadap pengalaman penggunaan (ISO, 2018). Dalam konteks pengujian perangkat lunak, *usability testing* menjadi pelengkap pengujian fungsional karena sistem yang secara teknis berfungsi dengan benar belum tentu mudah dipahami dan digunakan oleh pengguna akhir.

*System Usability Scale* (SUS) adalah instrumen kuesioner standar yang dikembangkan oleh John Brooke untuk mengukur persepsi *usability* secara cepat dan andal. SUS terdiri dari sepuluh pernyataan yang diukur menggunakan skala *Likert* lima poin, di mana pernyataan bernomor ganjil bersifat positif dan pernyataan bernomor genap bersifat negatif. Perhitungan skor SUS dilakukan dengan menghitung kontribusi setiap pernyataan, untuk pernyataan positif, kontribusinya adalah posisi skala dikurangi 1, sedangkan untuk pernyataan negatif, kontribusinya adalah 5 dikurangi posisi skala, kemudian menjumlahkan seluruh kontribusi dan mengalikannya dengan 2,5 sehingga menghasilkan skor akhir dalam rentang 0 hingga 100 (Brooke, 1996).




Gambar 2. 2 Peringkat adjektif, Skor *Acceptable* dan Peringkat dalam SUS

Sumber: Bangor et al., 2009

Interpretasi skor SUS mengacu pada skala penilaian adjektif yang dikembangkan berdasarkan analisis empiris terhadap lebih dari seribu studi *usability*. Pada Gambar 2.2 menunjukkan skor di bawah 50 dikategorikan sebagai tidak dapat diterima, skor antara 50 hingga 70 dikategorikan sebagai marginal, dan skor di atas 68 dikategorikan sebagai dapat diterima. Rata-rata skor SUS dari berbagai studi berada di sekitar 68, sehingga skor di atas angka tersebut mengindikasikan *usability* di atas rata-rata industri. SUS dipilih sebagai instrumen evaluasi *usability* karena beberapa



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

keunggulan: instrumen ini bersifat *technology-independent* sehingga dapat digunakan untuk berbagai jenis sistem, terbukti andal bahkan dengan jumlah responden yang kecil, dan menghasilkan satu skor tunggal yang mudah dipahami dan dikomunikasikan kepada pemangku kepentingan non-teknis (Bangor et al., 2009).

## 2.2 Teknologi yang digunakan

Teknologi yang digunakan pada penelitian ini adalah sebagai berikut.

### 2.2.1 React.js

React adalah *library* JavaScript *open-source* untuk membangun antarmuka pengguna berbasis komponen. Berbeda dari *framework* yang mendikte struktur aplikasi secara menyeluruh, React fokus pada penyusunan komponen UI yang dapat digunakan ulang, sementara aspek lain seperti *routing*, manajemen *state*, dan komunikasi HTTP dapat ditambahkan sesuai kebutuhan aplikasi. Karakteristik ini menjadi relevan pada aplikasi *dashboard* yang banyak menampilkan elemen visualisasi data, karena setiap elemen seperti kartu metrik, grafik tren, tabel proyek, dan filter periode dapat dirancang sebagai komponen terpisah yang dikombinasikan untuk membentuk tampilan utuh (react.dev, 2026).

Tabel 2. 2 Perbandingan React.js vs Angular vs Vue.js

| Kriteria | React | Angular | Vue.js |
|---|---|---|---|
| Sifat | *Library UI* | *Framework* lengkap | *Framework* fleksibel |
| Bahasa Utama | JavaScript atau TypeScript | TypeScript | JavaScript atau TypeScript |
| Manajemen *State* | Eksternal | *Built-in* | *Built-in* |
| Ekosistem Visualisasi Data | Sangat luas | Cukup luas | Cukup |
| Tingkat Pembelajaran | Sedang | Curam | Rendah |

| Kriteria | React | Angular | Vue.js |
|---|---|---|---|
| Adopsi pada *Dashboard* | Sangat luas dengan banyak referensi implementasi | Lazim di sistem *enterprise* | Berkembang, umum pada aplikasi menengah |

Berdasarkan perbandingan pada Tabel 2.2, React dipilih sebagai kerangka kerja *frontend* pada sistem. Dibandingkan Angular yang mewajibkan TypeScript dan memiliki tingkat pembelajaran yang curam, serta Vue.js yang ekosistem visualisasi datanya lebih terbatas, React menawarkan ekosistem visualisasi data paling luas yang relevan untuk kebutuhan *dashboard monitoring* dan KPI pada penelitian ini. Selain itu, fleksibilitas React dalam memilih manajemen *state* dan bahasa pemrograman, serta banyaknya referensi implementasi *dashboard* yang tersedia, menjadikannya pilihan yang paling sesuai dengan kebutuhan pengembangan sistem ini.

### 2.2.2 Node.js

Node.js adalah *runtime* JavaScript yang dirancang untuk dijalankan di sisi *server*, dibangun di atas mesin V8 milik Chromium. Yang membedakan Node.js dari *server* konvensional adalah cara ia menangani permintaan, seperti Karakter utama Node.js terletak pada model konkurensi *event loop* yang bersifat *non-blocking*, sehingga operasi seperti pembacaan basis data, permintaan ke layanan eksternal, dan pengiriman respons HTTP dapat ditangani secara konkuren tanpa membuka satu *thread* baru untuk setiap permintaan. Pendekatan ini berbeda dari model *thread-per-request* yang umum digunakan pada *server* berbasis Java atau PHP tradisional (nodejs.org, 2026). Selain itu, penggunaan JavaScript pada sisi *server* memungkinkan penyatuan stack pengembangan dengan React di sisi antarmuka, sehingga tim pengembang dapat berbagi format data, struktur tipe, dan utilitas validasi di kedua sisi tanpa perlu menerjemahkan antar bahasa. Konsolidasi stack ini menjadi pertimbangan praktis untuk pengembangan skripsi yang ditangani satu pengembang dalam ruang waktu yang terbatas.



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tabel 2. 3 Perbandingan Node.js vs Java Spring Boot vs Python vs Go

| Kriteria | Node.js | Java Spring Boot | Python | Go (Gin) |
|---|---|---|---|---|
| Bahasa | JavaScript, asinkron *event-driven* | Java, OOP dengan *thread* pool | Python, dinamis dengan opsi *async* | Go, statis dengan *goroutines* |
| Model Konkurensi | Event loop single-*thread* non-blocking | *Thread*-per-*request* via *thread* pool | Sinkron (Flask) atau *async* (FastAPI) | *Goroutines* ringan multi-core |
| Keselarasan dengan React di *Frontend* | Sama bahasa, tipe data dapat dibagi | Berbeda bahasa, perlu konversi tipe | Berbeda bahasa, perlu konversi tipe | Berbeda bahasa, perlu konversi tipe |
| Kemudahan Setup | Sangat sederhana (Node dan npm) | Lebih kompleks (JVM, Maven atau Gradle) | Sederhana (Python dan pip) | Sederhana (binary tunggal) |
| Ekosistem *Library* | npm sangat luas, *library* terus berkembang | Spring ecosystem matang untuk *enterprise* | Sangat kuat untuk data science | Berkembang, fokus performa tinggi |
| Adopsi pada Aplikasi Web | Sangat luas, dominan di *startup* | Dominan di *enterprise* besar | Lazim untuk API yang ringan | Lazim untuk layanan performa tinggi |

Berdasarkan perbandingan pada Tabel 2.3, Node.js dipilih sebagai *runtime* sisi *server* untuk sistem. Keunggulan utama Node.js dibandingkan ketiga alternatif

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

lainnya terletak pada keselarasan bahasa dengan React di sisi *frontend*. Penggunaan JavaScript di kedua sisi memungkinkan tipe data dan utilitas validasi digunakan bersama tanpa konversi tambahan. Dibandingkan Java Spring Boot yang setup-nya lebih kompleks dan lazim untuk sistem *enterprise* berskala besar, serta Python dan Go yang memerlukan konversi tipe data antara *backend* dan *frontend*, Node.js menawarkan kemudahan pengembangan yang paling sesuai untuk sistem yang dibangun oleh satu pengembang dalam waktu terbatas.

### 2.2.3 MySQL

MySQL adalah sistem manajemen basis data relasional *open-source* yang menggunakan SQL sebagai bahasa kueri standar. Pada versi modern, MySQL menggunakan InnoDB sebagai *storage engine* bawaan yang mendukung penuh prinsip transaksi *Atomicity*, *Consistency*, *Isolation*, *Durability* (ACID), sehingga operasi penulisan data yang melibatkan beberapa tabel secara bersamaan dapat dijamin selesai secara utuh atau dibatalkan seluruhnya bila terjadi kegagalan. InnoDB juga mendukung *constraint foreign key* untuk menjaga integritas referensial antar tabel yang saling berelasi (dev.mysql, 2026).

Tabel 2. 4 Perbandingan MySQL vs PostgreSQL vs MongoDB

| Kriteria | MySQL | PostgreSQL | MongoDB |
|---|---|---|---|
| Model Data | Relasional dalam bentuk tabel | Relasional | Dokumen |
| Kepatuhan ACID | Penuh pada InnoDB | Penuh dengan fitur tambahan | *Subset*, dengan transaksi multi-dokumen |
| *Constraint Foreign Key* | Didukung penuh | Didukung penuh | Tidak ada |
| Tipe Kueri Kompleks | Mendukung JOIN multi-tabel | Mendukung CTE, window function, partial index | *Aggregation pipeline* |

| Kriteria | MySQL | PostgreSQL | MongoDB |
|---|---|---|---|
| Tipe Data Khusus | Standar SQL dan JSON | SQL, JSON, *array* dan tipe kustom | Dokumen fleksibel tanpa skema kaku |
| Kelaziman pada Sistem Web Bisnis | Sangat lazim, banyak referensi pada aplikasi serupa | Lazim, sering pada sistem *enterprise* dengan logika kompleks | Lazim untuk data semi-terstruktur seperti katalog produk |

Berdasarkan perbandingan pada Tabel 2.4, MySQL dipilih sebagai sistem basis data untuk sistem. Data pada sistem ini bersifat relasional dan terstruktur. Proyek memiliki banyak *milestone* yang dimiliki oleh Consultant, dan setiap transaksi status harus tercatat dengan konsisten, sehingga model relasional dengan dukungan *foreign key* dan kepatuhan ACID penuh menjadi kebutuhan utama. Dibandingkan PostgreSQL yang fiturnya lebih lengkap namun cenderung digunakan pada sistem *enterprise* dengan logika data yang lebih kompleks, MySQL menawarkan kemudahan *setup* dan banyaknya referensi implementasi pada aplikasi web bisnis yang serupa. MongoDB tidak dipilih karena ketiadaan *foreign key* dan model dokumennya kurang sesuai untuk data yang sifatnya relasional dan membutuhkan konsistensi antar tabel.

### 2.2.4 RESTful API

*Representational State Transfer* (REST) adalah gaya arsitektur untuk perancangan sistem terdistribusi. REST tidak menetapkan protokol baru, melainkan menetapkan sekumpulan *constraint* yang ketika diterapkan bersama menghasilkan sistem yang skalabel dan mudah dipelihara. *Constraint* utama REST meliputi pemisahan *client* dan *server*, sifat *stateless* pada setiap permintaan, dukungan mekanisme *caching*, antarmuka yang seragam (*uniform interface*) berbasis *resource*, serta struktur sistem yang berlapis. Pada implementasi modern, REST umumnya dibangun di atas protokol HTTP dengan format data JSON sebagai standar pertukaran (Datatracker, 2026).



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tabel 2. 5 Perbandingan RESTful API vs GraphQL vs gRPC

| Kriteria | RESTful API | GraphQL | gRPC |
|---|---|---|---|
| Pola Komunikasi | *Resource*-centric berbasis URL | *Query* language fleksibel | Remote Procedure Call |
| Format Data | JSON | JSON | Protocol Buffers |
| Dukungan *Native Browser* | Penuh | Penuh | Terbatas |
| Mekanisme *Caching* | HTTP *caching native* | *Caching* kustom di klien | Tidak ada *caching* HTTP otomatis |
| *Tooling Debugging* | Sangat luas | *Tooling* khusus | *Tooling* khusus |
| Cocok untuk Aplikasi Web CRUD | Sangat cocok | Cocok untuk relasi data dalam | Lebih cocok untuk komunikasi antar layanan |

Berdasarkan perbandingan pada Tabel 2.5, RESTful API dipilih sebagai pola komunikasi antara *frontend* dan *backend* sistem. Dibandingkan GraphQL yang memerlukan *tooling* khusus dan lebih tepat digunakan ketika relasi antar data sangat kompleks, serta gRPC yang dukungan *browser*-nya terbatas dan lebih cocok untuk komunikasi antar layanan *backend*, RESTful API menawarkan dukungan *browser* penuh, *caching* HTTP bawaan, dan kemudahan *debugging* melalui *tooling* yang sudah sangat luas. Karakteristik ini paling sesuai untuk sistem dengan operasi CRUD terstruktur seperti pengelolaan proyek, *milestone*, dan KPI pada penelitian ini.

### 2.2.5 Fetch API

Fetch API adalah antarmuka pemrograman HTTP *client native* yang tersedia pada seluruh *browser* modern dan *runtime* JavaScript termasuk Node.js versi 18 ke atas.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Fetch API menggantikan XMLHttpRequest yang lebih tua dengan menyediakan antarmuka berbasis Promise yang lebih bersih dan mudah dipadukan dengan *syntax async/await*. Fetch API menjadi standar bawaan platform web dan tidak memerlukan instalasi pustaka tambahan, sehingga menghasilkan *bundle size* aplikasi yang lebih kecil dan mengurangi ketergantungan pada pustaka pihak ketiga (MDN Web Docs, 2024).

Fetch API menyediakan fungsi global *fetch()* yang menerima dua argumen, yaitu URL target permintaan dan objek opsi yang memuat konfigurasi permintaan seperti *method*, *headers*, dan *body*. Fungsi ini mengembalikan Promise yang *resolve* dengan objek *Response* berisi metadata respons seperti status *code*, *headers*, dan *body* yang dapat di-*parse* sebagai JSON, *text*, atau *blob* sesuai kebutuhan. Karakter *low-level* dari Fetch API memberikan fleksibilitas penuh kepada pengembang untuk merancang abstraksi yang sesuai dengan kebutuhan spesifik aplikasi, termasuk penanganan *token* autentikasi, deteksi error berdasarkan status HTTP, dan transformasi data respons.

Tabel 2. 6 Perbandingan Fetch API vs Axios vs XMLHttpRequest

| **Aspek** | **Fetch API** | **Axios** | **XMLHttpRequest** |
|---|---|---|---|
| Tipe | *Native browser* | *Library* pihak ketiga | *Native browser* |
| *Bundle Size* | 0 KB | 13 KB | 0 KB |
| API *Style* | Promise-*based* | Promise-*based* | *Callback-based* |
| Auto JSON *Parsing* | Manual | Otomatis | Manual |
| *Interceptor* | Tidak ada | Bawaan | Tidak ada |
| *Request Cancellation* | AbortController | CancelToken | xhr.abort() |
| *Browser Support* | Modern *browser* | Semua *browser* | Semua *browser* |
| *Dependency* | Tidak ada | Perlu instalasi | Tidak ada |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| Aspek | Fetch API | Axios | XMLHttpRequest |
|---|---|---|---|
| Penanganan Error | Hanya *network error* | Berdasarkan status HTTP | Manual |
| *Streaming* | *Native* ReadableStream | Tidak didukung penuh | Terbatas |

Berdasarkan perbandingan pada Tabel 2.6, Fetch API dipilih sebagai mekanisme komunikasi HTTP pada sistem. Sebagai API *native browser*, Fetch API tidak menambah ukuran *bundle* aplikasi dan tidak memerlukan instalasi dependensi tambahan. Dibandingkan XMLHttpRequest yang penulisan kodenya berbasis *callback* dan sudah dianggap cara lama, serta Axios yang menambah sekitar 13 KB dan memerlukan instalasi, Fetch API menawarkan pendekatan Promise-*based* yang modern dengan dukungan *streaming* bawaan melalui ReadableStream. Ketiadaan *interceptor* bawaan dan penanganan *error* yang harus dilakukan manual bukan menjadi kendala berarti karena kebutuhan tersebut dapat dipenuhi melalui *wrapper* sederhana, sehingga Fetch API tetap menjadi pilihan yang paling ringan dan sesuai untuk kebutuhan sistem ini.

## 2.3 Penelitian Terdahulu

Tabel 2. 7 Penelitian Terdahulu

| Penelitian Terdahulu | Ringkasan | Masalah | Metode | Gap Research |
|---|---|---|---|---|
| *The Use of Business Intelligence Software to Monitor Key Performance Indicators (KPIs) for the* | Identifikasi 18 KPI pemeliharaan dan membangun *dashboard* Power BI | *Monitoring* pemeliharaan alat medis belum efektif karena minim | *Business Intelligence* dan Power BI untuk visualisasi KPI *real-time*. | Berfokus pada visualisasi KPI pemeliharaan alat medis melalui Power BI sebagai alat analisis terpisah dari |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| Penelitian Terdahulu | Ringkasan | Masalah | Metode | Gap Research |
|---|---|---|---|---|
| *Evaluation of a Computerized Maintenance Management System* (CMMS) (Picozzi et al., 2024) | untuk analisis dinamis. | kontrol otomatis. | | sistem operasional. Belum mengintegrasikan perhitungan KPI secara otomatis dari data aktivitas pekerjaan yang berjalan. |
| *Implementing Key Performance Indicators and Designing Dashboard Solutions in an Automotive Components Company: A Case Study* (Nunes et al., 2024) | Implementasi 22 KPI lintas departemen dan *dashboard* Power BI berdasarkan masukan karyawan. | KPI belum terstandarisasi sehingga keputusan manajemen kurang optimal. | Survei karyawan untuk desain KPI dan *dashboard* interaktif. | KPI dikumpulkan melalui survei karyawan dan divisualisasikan melalui Power BI yang berdiri sendiri. Belum ada mekanisme yang menghitung skor kinerja secara otomatis dari data progres pekerjaan aktual dalam satu platform. |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| Penelitian Terdahulu | Ringkasan | Masalah | Metode | Gap Research |
|---|---|---|---|---|
| Analisis Kinerja Karyawan Melalui Penerapan Key Performance Indicator (KPI) di PT Jasa dan Kepariwisataan Jawa Barat (Perseroda) (Hersusetiyati et al., 2024) | Menganalisis penerapan KPI sebagai alat penilaian kinerja karyawan. KPI yang terstruktur terbukti optimal mencapai target perusahaan, namun terkendala ketidakdisiplinan karyawan dalam pengisian data KPI berkala. | Penilaian kinerja tanpa sistem terstruktur menghasilkan evaluasi subjektif. Keterlambatan *input* data KPI menurunkan efektivitas evaluasi. | Kualitatif analisis deskriptif (observasi, wawancara, dokumentasi). | Penelitian ini sendiri mengakui kelemahannya, karyawan yang terlambat mengisi KPI mendapat nilai 0, artinya sistem masih bergantung penuh pada *input* manual. Belum ada solusi teknis untuk mengotomasi pengumpulan data kinerja dari aktivitas pekerjaan. |
| Pengembangan Aplikasi *Monitoring* Informasi Proyek Berbasis Web Menggunakan Metode | Aplikasi web untuk sentralisasi data proyek, pelaporan terintegrasi, progres & | Pelaporan manual via Excel tersebar yang tidak terintegrasi & rawan ereor. | *ICONIX Process*, UML, Laravel, MySQL, *black box testing*. | Berhasil mensentralisasi *monitoring* proyek berbasis web, namun terbatas pada pelaporan progres dan |

| Penelitian Terdahulu | Ringkasan | Masalah | Metode | Gap Research |
|---|---|---|---|---|
| *Iconix Process* (Studi Kasus: PT Telkom Akses Witel Semarang) (Agustha et al., 2024) | tagihan *real-time*. | | | tagihan saja. Data aktivitas pekerjaan yang terekam belum dimanfaatkan sebagai dasar perhitungan KPI karyawan. |
| Pengukuran Kinerja Sumber Daya Manusia menggunakan Metode *Human Resources Scorecard* (Karlina and Samanhudi, 2023) | Pengukuran kinerja SDM seimbang finansial & non-finansial (HRS). | Penilaian belum komprehensif (khususnya non-finansial) yang menurunkan kualitas layanan. | HRS 4 perspektif, bobot AHP, evaluasi *Traffic Light System* (TLS) | Menghasilkan kerangka pengukuran kinerja SDM yang komprehensif, namun data kinerja tetap dikumpulkan secara manual dan terpisah dari sistem operasional. Tidak membahas integrasi dengan *monitoring* pekerjaan. |
| Perancangan Sistem | Perancangan sistem web | Layanan & data belum | Lean Business | Berfokus pada perancangan |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| Penelitian Terdahulu | Ringkasan | Masalah | Metode | Gap Research |
|---|---|---|---|---|
| Informasi Pembelajaran Berbasis Web (Sabar et al., 2017) | terintegrasi, menghasilkan dokumen desain & SRS. | terintegrasi sehingga *monitoring* dan akses terbatas. | Model Design, SWOT, UML (OOAD). | sistem informasi pendidikan berbasis web. Tidak mencakup *monitoring* proyek berbasis *workflow* maupun penilaian kinerja berbasis KPI, sehingga hanya relevan sebagai referensi metodologi perancangan sistem. |

Berdasarkan Tabel 2.7, penelitian terdahulu menunjukkan dua kelompok kajian yang relevan dengan penelitian ini, yaitu penelitian di bidang *monitoring* proyek berbasis web dan penelitian di bidang penilaian kinerja berbasis KPI. Pada kelompok *monitoring* proyek, (Agustha et al., 2024), membangun aplikasi web untuk mensentralisasi data proyek dan pelaporan progres yang sebelumnya dilakukan secara manual melalui Excel tersebar. Penelitian tersebut berhasil mengintegrasikan data proyek dan tagihan secara terpusat, namun cakupannya terbatas pada aspek pelaporan progres saja tanpa menghubungkan data aktivitas

pekerjaan yang terekam dengan penilaian kinerja karyawan yang mengerjakan proyek tersebut.

Pada kelompok penilaian kinerja, (Picozzi et al., 2024) dan (Nunes et al., 2024), mengembangkan KPI dan *dashboard* Power BI untuk mendukung pengambilan keputusan berbasis data, sedangkan (Hersusetiyati et al., 2024) serta (Karlina and Samanhudi, 2023), berfokus pada perancangan kerangka penilaian kinerja karyawan yang lebih terstruktur dan komprehensif. Meskipun demikian, keempat penelitian tersebut memiliki kesamaan kelemahan, yaitu data kinerja dikumpulkan secara manual dan *dashboard* yang digunakan berdiri sendiri terpisah dari sistem operasional, sehingga masih terdapat ketergantungan pada kedisiplinan pengguna dalam melakukan *input* data. (Sabar et al., 2017) turut memberikan kontribusi sebagai referensi metodologi perancangan sistem berbasis web, meskipun konteksnya berbeda pada bidang pendidikan.

Berdasarkan penelusuran terhadap keenam penelitian terdahulu tersebut, terdapat celah yang belum dijawab, yaitu belum adanya sistem yang mengintegrasikan *monitoring* proyek berbasis *workflow* dengan penilaian kinerja karyawan berbasis KPI dalam satu platform, di mana data aktivitas pekerjaan yang terekam pada sistem digunakan langsung sebagai dasar perhitungan skor kinerja secara otomatis tanpa memerlukan *input* manual. Penelitian ini hadir untuk mengisi celah tersebut melalui pengembangan sistem yang menggabungkan kedua aspek dalam satu aplikasi berbasis web untuk menjawab kebutuhan operasional perusahaan.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

# BAB III
# METODE PENELITIAN

## 3.1 Rancangan Penelitian

Penelitian ini menggunakan pendekatan kualitatif untuk memahami kondisi operasional perusahaan. Pendekatan ini diperlukan karena permasalahan melibatkan kebiasaan kerja, kebutuhan informasi lintas peran, dan integrasi sistem yang kompleks. Penelitian ini menggunakan strategi studi kasus karena berfokus pada satu organisasi spesifik dan menghasilkan aplikasi yang dapat langsung digunakan, bukan sekadar analisis konseptual. Pengembangan sistem dilakukan menggunakan metode rekayasa perangkat lunak yang mencakup tahapan identifikasi masalah, analisis kebutuhan, perancangan, implementasi, pengujian, dan evaluasi. Tahapan yang berurutan ini memastikan solusi yang dihasilkan berfungsi, teruji, dan dievaluasi secara sistematis.

Data dikumpulkan melalui observasi, wawancara semi-terstruktur, dan dokumentasi yang terdokumentasi pada (Lampiran 1 dan Lampiran 2). Data kualitatif dominan pada tahap analisis kebutuhan untuk menggali pemahaman mendalam, sedangkan data kuantitatif digunakan pada tahap pengujian melalui metrik *test case*, validasi KPI, dan *System Usability Scale*. Kombinasi ini memastikan sistem relevan dengan kebutuhan organisasi dan mudah digunakan.

### 3.1.1 Teknik Pengumpulan Data

Data dikumpulkan melalui tiga teknik yang dirancang saling melengkapi. Observasi langsung dilakukan untuk melihat bagaimana proses *monitoring* dan penilaian kinerja berjalan dalam praktik sehari-hari. Observasi mencakup media yang digunakan (excel, WhatsApp, *email*), pola koordinasi antar peran (CEO, COO, PM, dan Consultant), dan hambatan operasional yang terjadi.

Wawancara semi-terstruktur dilakukan dengan CEO, COO, Project Manager, dan Consultant untuk menggali perspektif mereka. Teknik ini dipilih karena memungkinkan peneliti memandu percakapan namun tetap fleksibel terhadap

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

jawaban informan. Melalui wawancara, peneliti memahami kebutuhan, kendala, dan harapan setiap peran terhadap sistem yang akan dikembangkan.

Dokumentasi dilakukan dengan mengumpulkan dan menelaah dokumen kerja yang ada, seperti *template* penugasan, format pelaporan, excel tracking, dan dokumen KPI. Dokumentasi memberikan bukti konkret tentang proses dan media yang sedang berjalan dalam praktik. Kombinasi ketiga teknik memastikan kebutuhan sistem didasarkan pada multiple sources sehingga benar-benar representatif dari kebutuhan organisasi.

### 3.1.2 Teknik Analisis Data

Analisis data menggunakan pendekatan kualitatif untuk mengolah data observasi, wawancara, dan dokumentasi menjadi informasi terstruktur. Proses dimulai dengan penelaahan menyeluruh terhadap semua data yang terkumpul untuk memahami konteks masalah. Dilanjutkan dengan identifikasi tema-tema utama terkait *monitoring* proyek (deteksi keterlambatan, pelaporan status) dan penilaian kinerja (parameter KPI, transparansi).

Data yang telah dianalisis kemudian diterjemahkan menjadi kebutuhan sistem yang spesifik dan implementatif. Kebutuhan difokuskan pada aspek fungsional seperti *dashboard monitoring*, *alert* keterlambatan, dan perhitungan KPI otomatis. Analisis juga mencakup kebutuhan non-fungsional seperti keamanan berbasis RBAC, performa *real-time*, dan *usability* yang baik lintas peran.

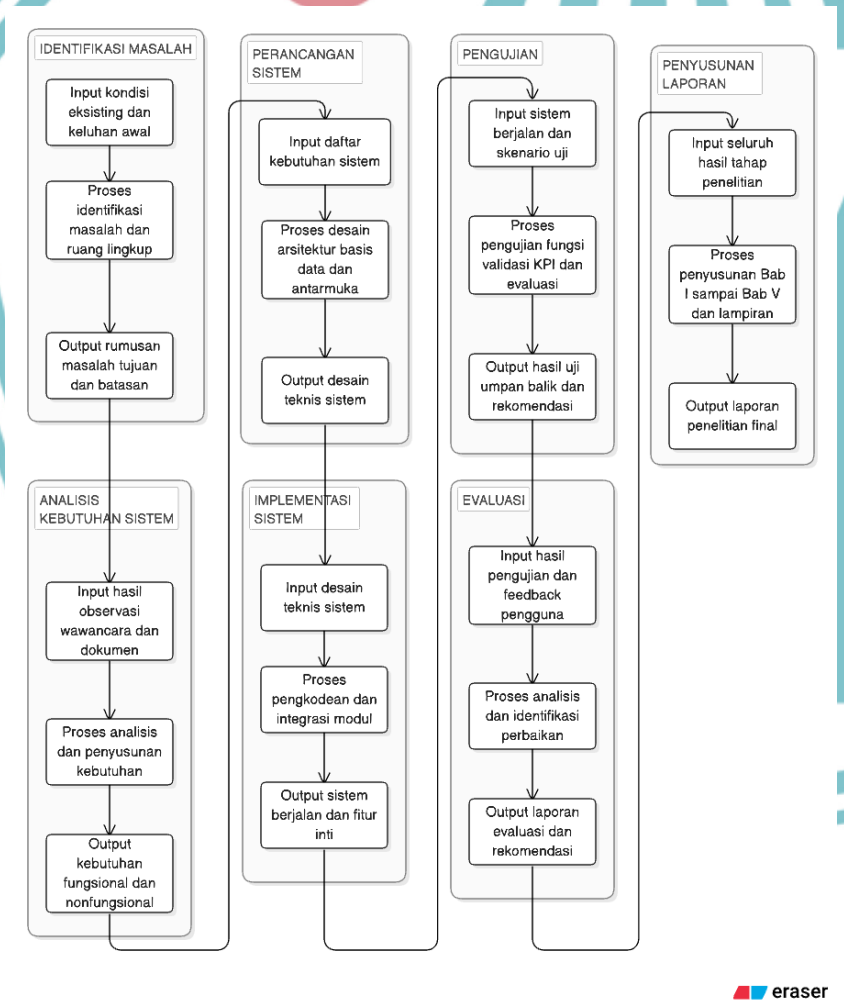
Pada tahap pengujian, analisis dilengkapi dengan aspek kuantitatif melalui tiga metrik utama. Metrik pertama adalah *test case* dari *black box testing* untuk verifikasi fungsi. Metrik kedua adalah *User Acceptance Testing* dengan memastikan sistem memenuhi kriteria sesuai dengan pengguna inginkan. Metrik ketiga adalah *System Usability Scale* (SUS) yang diisi pengguna untuk mengukur kemudahan penggunaan sistem secara objektif.

## 3.2 Tahapan Penelitian

Tahapan penelitian dilakukan secara sistematis agar proses pengembangan sistem berjalan terarah dan memiliki hubungan jelas antar tahap. Penelitian ini mencakup

tujuh tahap utama, yaitu: identifikasi masalah, analisis kebutuhan, perancangan sistem, implementasi, pengujian, evaluasi, dan penyusunan laporan. Dengan susunan tersebut, setiap tahap saling mendukung dan membentuk alur kerja yang runtut dari awal hingga akhir penelitian.

Setiap tahapan penelitian memiliki *input* yang jelas, metode atau teknik yang digunakan, dan *output* yang dihasilkan. *Input* dari satu tahap menjadi dasar bagi tahap berikutnya, sehingga masalah yang diidentifikasi pada tahap pertama terhubung dengan analisis kebutuhan, kemudian perancangan, implementasi, dan evaluasi. Pendekatan sistematis ini memastikan bahwa solusi yang dihasilkan berdasarkan pemahaman mendalam tentang masalah dan kebutuhan pengguna.




Gambar 3. 1 Diagram Tahapan Penelitian

Sumber: Olahan Peneliti



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Gambar 3.1 menunjukkan diagram tahapan penelitian yang menggambarkan hubungan antar tahap secara visual. Diagram tersebut memperlihatkan bahwa proses penelitian tidak dilakukan secara terpisah, melainkan saling berkaitan dan saling mendukung. Dengan visualisasi ini, pembaca dapat memahami bahwa penelitian mengikuti alur pengembangan sistem yang sistematis dan terstruktur dari identifikasi masalah hingga penyusunan laporan akhir.

### 3.2.1 Identifikasi Masalah

Tahap identifikasi masalah adalah tahap awal yang bertujuan memperoleh pemahaman menyeluruh tentang kondisi *monitoring* proyek dan penilaian kinerja karyawan yang berjalan. Pada tahap ini, peneliti melakukan observasi langsung terhadap proses yang berjalan untuk melihat media atau *tools* yang digunakan, pola koordinasi antar peran, dan hambatan operasional, serta melakukan wawancara dengan *stakeholder* kunci (CEO, COO, PM, dan Consultant) untuk memahami perspektif mereka tentang masalah dan dampaknya terhadap operasional perusahaan. Temuan dari observasi dan wawancara kemudian dianalisis untuk mengidentifikasi akar masalah dan kesenjangan antara kondisi aktual dengan kondisi ideal, diverifikasi kembali dengan *stakeholder* untuk memastikan akurasi, dan menghasilkan daftar masalah terverifikasi yang disepakati bersama. *Output* dari tahap ini mencakup: daftar masalah terverifikasi, ruang lingkup penelitian, rumusan masalah spesifik, tujuan penelitian terukur, dan batasan penelitian yang menjelaskan fokus dan cakupan penelitian sebagai acuan untuk tahap analisis kebutuhan berikutnya.

### 3.2.2 Analisis Kebutuhan

Tahap analisis kebutuhan bertujuan menerjemahkan masalah terverifikasi menjadi kebutuhan sistem yang spesifik dan implementatif dengan menganalisis data observasi dan wawancara menggunakan pendekatan analisis kualitatif untuk mengidentifikasi tema-tema berkaitan dengan kebutuhan sistem. Kebutuhan dikelompokkan dan dipetakan berdasarkan peran pengguna (CEO, COO, PM, dan Consultant) untuk memastikan setiap peran memiliki kebutuhan yang jelas dan spesifik, dibagi menjadi dua kategori: kebutuhan fungsional seperti *dashboard monitoring*, *alert* keterlambatan, perhitungan KPI otomatis, dan laporan penilaian

kinerja, serta kebutuhan non-fungsional seperti pembatasan akses berbasis peran (RBAC), kecepatan akses *real-time*, dan kemudahan penggunaan lintas peran. *Output* dari tahap ini adalah dokumen spesifikasi kebutuhan sistem yang mendetail dan prioritas kebutuhan, menjelaskan setiap kebutuhan dengan jelas (siapa yang membutuhkan, konteks apa, hasil apa yang diharapkan) agar prioritas kebutuhan membantu tim implementasi fokus pada fitur-fitur kritis yang mengatasi masalah utama.

### 3.2.3 Perancangan Sistem

Tahap perancangan sistem bertujuan mengubah kebutuhan sistem menjadi desain teknis yang siap diimplementasikan dengan merancang beberapa aspek penting yaitu, arsitektur sistem seperti bagaimana komponen *frontend*, *backend*, *database* bisa bekerja bersama, struktur *database* dengan *Entity Relationship Diagram* (ERD), *workflow monitoring* proyek dengan status dan aturan transisi jelas, mekanisme bagaimana data progres proyek menjadi sumber perhitungan KPI (indikator, formula, bobot), dan *mockup* atau *wireframe* antarmuka (*dashboard monitoring*, laporan per peran). *Output* dari tahap ini mencakup: diagram arsitektur sistem, diagram ERD, *flowchart* proses bisnis atau *Swim Lane Diagram*, *mockup* atau *wireframe* antarmuka pengguna, matriks RBAC menunjukkan hak akses setiap peran, dan spesifikasi teknis fitur-fitur utama yang menjadi *blueprint* bagi developer untuk memulai implementasi.

Perancangan *workflow* pada penelitian ini dilakukan secara sistematis melalui empat tahapan utama. Tahap pertama adalah identifikasi *state* pekerjaan berdasarkan hasil observasi proses bisnis *existing*. Pada penelitian ini, perancangan *workflow* dilakukan pada dua tingkat *granularity*, yaitu tingkat proyek dengan *state Awaiting Consultant*, *In Progress*, *On Hold*, *Completed*, dan *Cancelled*, serta tingkat *milestone* dengan *state Pending*, *In Progress*, *Done*, dan *Blocked*. Pemisahan tingkat *granularity* ini dimaksudkan agar *lifecycle* bisnis proyek secara keseluruhan dapat dibedakan dari eksekusi *task-level* yang lebih operasional. Tahap kedua adalah perumusan aturan transisi antar *state*, yang mencakup pemicu transisi (*event*), pihak yang memiliki kewenangan melakukan transisi, serta validasi yang harus dipenuhi sebelum transisi dieksekusi oleh sistem. Tahap ketiga adalah



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

penyusunan *Standard Operating Procedure* (SOP) digital yang mendokumentasikan aktivitas pengguna dan respon otomatis sistem pada setiap transisi *state*, termasuk *side effect* lintas modul seperti pemblokiran transisi saat DP belum dibayar dan pemicu final *invoice* saat proyek selesai. Tahap keempat adalah pemetaan *workflow* tersebut ke dalam komponen sistem agar setiap *state* dan transisinya terimplementasi secara terstruktur, dapat ditelusuri, dan konsisten dengan rule yang telah ditetapkan. Pendekatan ini sejalan dengan konsep WfMS, di mana *workflow* tidak hanya direpresentasikan sebagai alur status visual, tetapi sebagai *rule engine* yang mengendalikan pelaksanaan proses bisnis *end-to-end*.

### 3.2.4 Implementasi

Tahap implementasi bertujuan mewujudkan desain sistem menjadi aplikasi yang berfungsi dan dapat dijalankan dengan membangun aplikasi secara modular (modul- modul kecil dibangun independen kemudian diintegrasikan) mencakup modul autentikasi dan otorisasi (*login* berbasis peran), modul pengelolaan data proyek dan penugasan, modul pencatatan progres pekerjaan, modul pendeteksian keterlambatan dan *alert*, modul pengelolaan KPI, dan modul *dashboard monitoring*. Setiap modul dibangun mengikuti spesifikasi desain dengan testing awal (unit testing) untuk memastikan berfungsi sesuai spesifikasi, kemudian tahap integrasi menghubungkan modul-modul agar data mengalir dengan benar dan alur proses berjalan sesuai *workflow* yang dirancang. *Output* dari tahap ini adalah aplikasi *monitoring* proyek dan penilaian kinerja karyawan berbasis KPI yang sudah berjalan dan siap diuji, beserta dokumentasi teknis berupa API *endpoint*, struktur *database* yang diimplementasikan, dan penjelasan alur proses setiap modul.

Implementasi *workflow* dilakukan dengan menerjemahkan rancangan *state machine* ke dalam komponen sistem yang berfungsi sebagai *rule engine* operasional. *State* setiap proyek dan *milestone* disimpan dalam basis data sebagai kolom bertipe ENUM yang membatasi nilai status hanya pada *state* yang telah didefinisikan, sehingga konsistensi data terjamin pada level skema *database*. Transisi antar *state* dieksekusi melalui *transition service* pada lapisan *backend* yang melakukan empat tahap pemeriksaan secara berurutan, yaitu validasi terhadap matriks transisi yang



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

diperbolehkan, pemeriksaan prasyarat transisi seperti status pembayaran DP dan kelengkapan *milestone*, pemeriksaan otorisasi berdasarkan peran dan kepemilikan, serta eksekusi transisi dalam DB *transaction* agar bersifat *atomic*. Setiap transisi yang berhasil akan menghasilkan satu *record audit trail* pada tabel terpisah, dengan *project status transitions* mencatat perubahan status proyek dan *project milestone updates* mencatat perubahan status *milestone*. Selain *audit trail*, transisi tertentu juga memicu *side effect* otomatis seperti pemicu *final invoice* ketika proyek berstatus *Completed* dan pencatatan *activity* log pada *modul handover*. Sementara itu, deteksi keterlambatan diimplementasikan sebagai *derived monitoring metric* pada *dashboard*, yaitu nilai yang dihitung di *runtime* dari kombinasi target_date dan status *milestone* tanpa menyimpan *state* tambahan di database.

### 3.2.5 Pengujian

Tahap pengujian bertujuan memverifikasi aplikasi berfungsi sesuai kebutuhan, hasil perhitungan KPI akurat, dan dapat diterima pengguna dengan *usability* baik melalui tiga jenis pengujian: pengujian fungsional menggunakan *black box testing* memverifikasi setiap fungsi (*login*, *input* data, perhitungan, laporan) sesuai spesifikasi, validasi perhitungan KPI membandingkan hasil sistem versus manual menggunakan data sampel nyata dari proyek selesai, *User Acceptance Testing* (UAT) melibatkan pengguna setiap peran menjalankan skenario tugas nyata dan memberikan *feedback* kesesuaian sistem. Pengujian juga mencakup pengukuran *usability* menggunakan *System Usability Scale* (SUS), kuesioner standar diisi pengguna berbagai peran setelah menggunakan sistem, hasil dikonversi skor (0-100) menunjukkan tingkat kemudahan penggunaan. *Output* adalah laporan pengujian lengkap mencakup: test report *black box testing* (fungsi pass/fail), validasi KPI (perbandingan sistem vs manual), hasil UAT per peran beserta *feedback*, dan skor SUS dengan interpretasinya.

### 3.2.6 Evaluasi

Tahap evaluasi bertujuan menganalisis hasil pengujian dan menentukan apakah sistem memenuhi kriteria kelulusan atau perlu perbaikan dengan menganalisis semua hasil pengujian (black box, validasi KPI, UAT, SUS) secara menyeluruh untuk mengidentifikasi aspek yang berhasil dan aspek yang memerlukan perbaikan,

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

investigatif mengidentifikasi akar penyebab kegagalan, kemudian diterjemahkan menjadi daftar perbaikan diprioritaskan berdasarkan dampak terhadap fungsi kritis dan pengalaman pengguna. Perbaikan dikerjakan iteratif, setiap perbaikan diikuti pengujian ulang memastikan berhasil tanpa menimbulkan masalah baru, berlanjut sampai sistem memenuhi kriteria kelulusan. *Output* adalah laporan evaluasi komprehensif mencakup: analisis hasil pengujian, daftar masalah dan statusnya (solved/in progress/pending), rekomendasi perbaikan yang diimplementasikan, dan kesimpulan apakah sistem siap diimplementasikan atau masih memerlukan perbaikan lebih lanjut.

### 3.2.7 Penyusunan Laporan

Tahap penyusunan laporan adalah tahap akhir mendokumentasikan seluruh proses penelitian dan hasil-hasilnya dalam bentuk laporan skripsi terstruktur dengan menyusun narasi penelitian mulai dari Bab I (Pendahuluan: latar belakang dan rumusan masalah), Bab II (Tinjauan Pustaka: landasan teori), Bab III (Metode Penelitian: metodologi), Bab IV (Hasil dan Pembahasan: implementasi dan hasil pengujian sistem), dan Bab V (Penutup: kesimpulan dan saran), serta menyiapkan lampiran mencakup dokumen-dokumen pendukung seperti hasil observasi, transkrip wawancara, instrumen penelitian, desain sistem, test report, hasil UAT, dan skor SUS yang berfungsi sebagai bukti empiris pertanggungjawaban ilmiah bahwa penelitian dilakukan dengan metodologi ketat dan data valid. Setiap bab dan lampiran dirancang agar pembaca mengikuti alur penelitian dengan jelas, memahami justifikasi setiap keputusan, dan melihat hubungan antara masalah teridentifikasi, teori digunakan, metodologi diterapkan, dan hasil dicapai. *Output* adalah laporan skripsi final siap dipertahankan di hadapan dosen penguji, mengintegrasikan semua pencapaian penelitian dalam narasi runtut terstruktur sebagai bentuk pertanggungjawaban ilmiah bahwa penelitian dilakukan dengan baik dan hasil dapat diterima akademik.

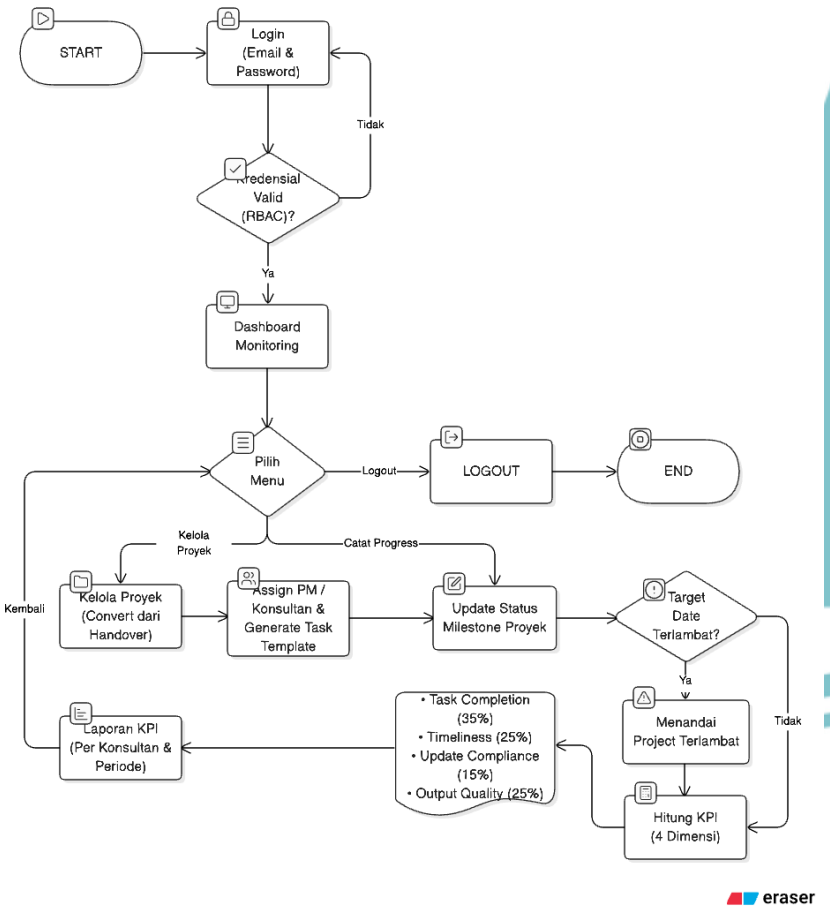
## 3.3 Objek Penelitian

Objek penelitian ini adalah sistem *monitoring* proyek dan penilaian kinerja karyawan yang akan dikembangkan, dengan fokus pada proses pengelolaan proyek, penugasan karyawan, pembaruan progres pekerjaan, pengendalian status pekerjaan,

serta evaluasi kinerja karyawan yang berkaitan langsung dengan aktivitas proyek. Fokus tersebut dipilih karena kedua proses tersebut merupakan inti permasalahan yang ditemukan pada kondisi existing perusahaan, di mana *monitoring* proyek masih manual dan penilaian kinerja belum terstandar.

### 3.3.1 Objek Fungsional

Objek fungsional dalam penelitian ini adalah kemampuan sistem dalam mendukung dua proses inti, yaitu *monitoring* proyek dan penilaian kinerja karyawan berbasis KPI. Lingkup objek fungsional mencakup fungsi autentikasi, pemusatan data proyek, pengelolaan *workflow* proyek dan *milestone*, deteksi keterlambatan, pengelolaan dan perhitungan KPI, serta visualisasi *dashboard* untuk dukungan keputusan manajerial.




Gambar 3. 2 *Flowchart* Aplikasi *Monitoring* Proyek Berbasis KPI

Sumber: Olahan Peneliti



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Pada Gambar 3.2 ditampilkan *flowchart* secara garis besar bagaimana cara kerja sistem *monitoring* proyek dan penilaian KPI. Alur dimulai dari proses *login*, dilanjutkan dengan navigasi *dashboard* sesuai peran pengguna, kemudian operasi inti berupa pengelolaan proyek atau pencatatan progres pekerjaan. Setelah progres diperbarui, sistem secara otomatis menjalankan pengecekan keterlambatan dan perhitungan KPI, lalu menyajikan hasilnya dalam laporan per Consultant dan per periode sebagai dasar pengambilan keputusan manajerial.

### 3.3.2 Objek Pengguna

Objek pengguna mencakup pihak-pihak yang akan menggunakan dan mengevaluasi sistem yang dikembangkan. Pengguna sistem terdiri dari lima peran utama sesuai struktur organisasi perusahaan, yaitu CEO, COO, Project Manager, Consultant, dan Superadmin. CEO memegang kewenangan strategis yang mencakup pengelolaan parameter KPI, sementara empat peran lainnya berperan pada level operasional dan administrasi sesuai tanggung jawab masing-masing.

### 3.3.3 Objek Data dan Dokumen

Objek data dan dokumen meliputi informasi yang dikelola oleh sistem untuk mendukung dua proses inti penelitian. Empat kategori utama yang menjadi objek penelitian, yaitu: pertama, data proyek dan penugasan yang mencakup identitas proyek, *scope* pekerjaan, jadwal perencanaan, dan tim yang ditugaskan, kedua, data pencatatan progres pekerjaan yang meliputi status *task* per *milestone*, persentase progres, dan indikator keterlambatan, ketiga, data penilaian kinerja yang meliputi indikator KPI, target, bobot, dan hasil perhitungan skor, serta keempat, dokumen pendukung operasional seperti *scope* proyek, *project* plan, dan dokumen *audit trail*.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

# BAB IV
# HASIL DAN PEMBAHASAN

## 4.1 Analisis Kebutuhan

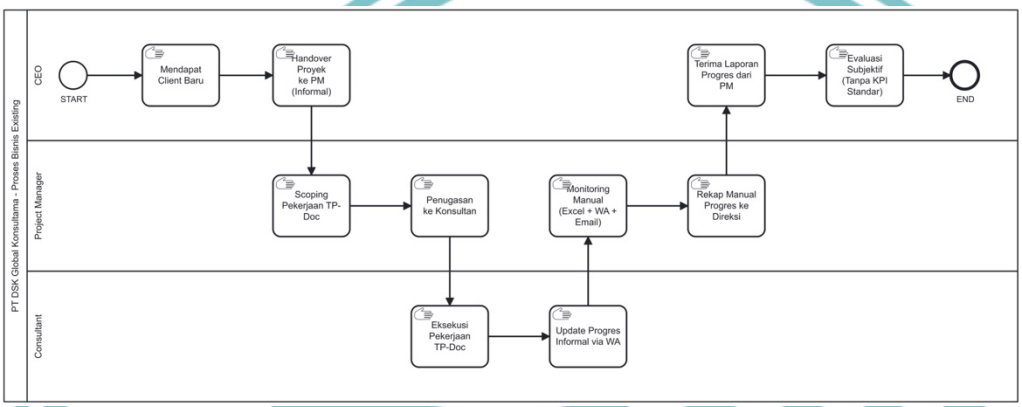
Bab ini menyajikan hasil analisis kebutuhan sistem yang diperoleh dari proses pengumpulan data melalui observasi, wawancara, dan dokumentasi terhadap kondisi operasional PT DSK Global Konsultama. Analisis kebutuhan dilakukan untuk memahami proses bisnis existing, mengidentifikasi permasalahan yang terjadi, serta merumuskan kebutuhan fungsional dan non-fungsional sistem yang akan dikembangkan. Hasil analisis ini menjadi landasan dalam tahap perancangan agar solusi yang dibangun benar-benar menjawab permasalahan nyata di perusahaan.

Pendekatan analisis dimulai dari pemetaan proses bisnis *existing*, dilanjutkan dengan perancangan proses bisnis *future*, dan diakhiri dengan identifikasi gap antara kondisi *existing* dan kondisi yang diinginkan. Setiap temuan dirumuskan menjadi tema kebutuhan yang spesifik dan diterjemahkan ke dalam kebutuhan fungsional dan non-fungsional. Kebutuhan tersebut kemudian dimodelkan secara visual menggunakan *Use Case Diagram* dan *Activity Diagram* untuk memberikan gambaran teknis yang siap diimplementasikan.

### 4.1.1 Analisis Proses Bisnis *Existing*

Proses bisnis utama perusahaan dimulai dari mendapatkan klien baru, dilanjutkan dengan penyusunan *scope* pekerjaan, penugasan tim Consultant, eksekusi proyek, hingga *delivery* hasil pekerjaan kepada klien. Selama proyek berjalan, Project Manager melakukan *monitoring* progres pekerjaan dan melaporkan status kepada CEO secara berkala. Proses *monitoring* proyek *existing* dilakukan secara manual menggunakan kombinasi beberapa media yang tidak terintegrasi. Project Manager mencatat progres pekerjaan di Excel pribadi, melakukan koordinasi tim melalui grup WhatsApp, dan melaporkan status kepada Direksi melalui *email* atau pertemuan tatap muka. Pendekatan ini menyebabkan informasi proyek tersebar di banyak tempat dan sulit diakses secara cepat ketika dibutuhkan untuk pengambilan keputusan. Proses penilaian kinerja karyawan *existing* juga dilakukan secara

manual oleh Project Manager dan CEO tanpa parameter *Key Performance Indicator* (KPI) yang terstandar. Penilaian dilakukan berdasarkan observasi subjektif PM terhadap kinerja Consultant selama proyek, dengan kriteria yang sering kali tidak konsisten antar proyek dan antar Project Manager. Akibatnya, hasil penilaian rentan terhadap bias personal dan sulit dipertanggungjawabkan secara objektif untuk keperluan *reward*, promosi, atau pengembangan karyawan.




Gambar 4. 1 BPMN Proses Bisnis *Existing*

Sumber: Olahan Peneliti

Pada Gambar 4.1 ditampilkan diagram alur proses bisnis *existing* perusahaan dalam bentuk *swim lane* diagram. Diagram tersebut memperlihatkan keterlibatan CEO, Project Manager, dan Consultant dalam siklus proyek konsultasi mulai dari akuisisi klien hingga evaluasi kinerja. Visualisasi ini menunjukkan bahwa proses bisnis melibatkan koordinasi lintas peran yang sangat bergantung pada komunikasi manual antar peran, yang menjadi sumber utama permasalahan operasional.

### 4.1.2 Analisis Proses Bisnis *Future*

Proses bisnis *future* dirancang untuk mengatasi permasalahan yang teridentifikasi pada proses bisnis *existing* dengan memanfaatkan sistem aplikasi yang akan dikembangkan. Pada proses bisnis *future*, seluruh aktivitas *monitoring* proyek dan penilaian kinerja karyawan dilakukan melalui satu platform terpusat yang dapat diakses oleh seluruh peran pengguna sesuai dengan hak akses masing-masing. Pendekatan ini menggantikan praktik kerja yang tersebar di berbagai media dengan basis data terintegrasi yang konsisten dan dapat ditelusuri.



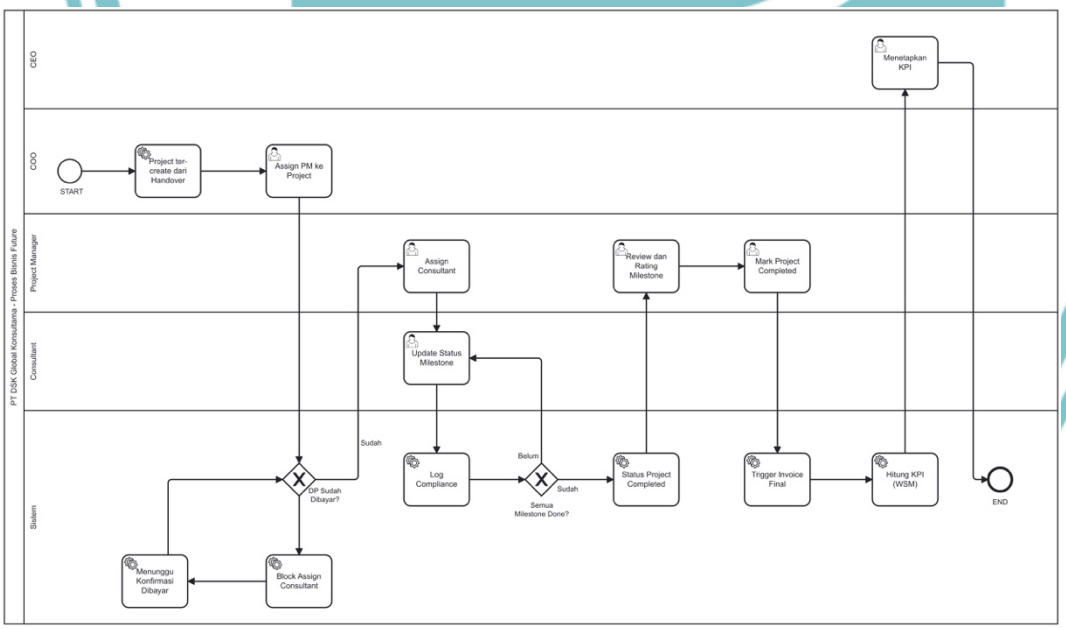
© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Pada proses bisnis *future*, Project Manager mencatat progres pekerjaan langsung di sistem dengan status terstruktur yang membedakan level proyek dan level *milestone*, sehingga informasi tersedia secara *real-time* bagi seluruh *stakeholder* yang berkepentingan. Sistem secara otomatis akan membandingkan progres aktual dengan jadwal rencana dan mengirimkan *alert* kepada PM dan Direksi ketika terdeteksi indikasi keterlambatan. Mekanisme ini menggantikan pelaporan manual yang reaktif dengan deteksi yang memungkinkan tindakan dilakukan sebelum keterlambatan berdampak signifikan.

Untuk proses penilaian kinerja, sistem mengelola parameter KPI yang ditetapkan oleh CEO sebagai pemegang kewenangan strategis dalam penilaian kinerja, dan menghitung pencapaian KPI secara otomatis berdasarkan data progres proyek yang tercatat di sistem. CEO dan COO dapat memantau status seluruh proyek dan kinerja tim melalui *dashboard* eksekutif yang menyajikan informasi secara visual dan *real-time*. Consultant juga memiliki akses untuk melihat KPI pribadi mereka, sehingga proses penilaian menjadi transparan dan mendorong perbaikan kinerja berkelanjutan.




Gambar 4. 2 BPMN Proses Bisnis Future

Sumber: Olahan Peneliti

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Pada Gambar 4.2 ditampilkan diagram alur proses bisnis future yang telah diintegrasikan dengan sistem aplikasi yang akan dikembangkan. Diagram tersebut menunjukkan adanya penambahan kolom "Sistem" pada *swim lane* yang merepresentasikan peran sistem dalam mengotomasi proses-proses yang sebelumnya dilakukan secara manual. Visualisasi ini memperlihatkan bagaimana sistem berperan sebagai penghubung antar peran dengan menyediakan basis data terpusat, deteksi keterlambatan otomatis, perhitungan KPI berbasis data, dan *dashboard real-time* bagi Direksi.

### 4.1.3 Perbandingan *Existing* dan *Future*

Berdasarkan pemodelan proses bisnis yang telah dilakukan, perbandingan antara proses *existing* dan proses *future* disusun berdasarkan tahapan proses bisnis yang terjadi dalam siklus pengelolaan proyek. Setiap tahapan dibandingkan untuk menunjukkan secara konkret perubahan yang dihasilkan oleh penerapan sistem terhadap proses operasional yang sebelumnya berjalan secara manual. Ringkasan perbandingan disajikan pada Tabel 4.1.

Tabel 4. 1 Perbandingan Proses Bisnis Existing dan Future

| No | Tahapan Proses Bisnis | Kondisi Existing | Kondisi *Future* |
|---|---|---|---|
| 1 | Inisiasi Proyek | CEO melakukan *handover* proyek ke PM secara informal tanpa pencatatan formal di sistem | COO menerima *handover* yang sudah disetujui, sistem otomatis membuat *record* proyek, COO melakukan *assign* PM secara formal |
| 2 | Validasi Sebelum Penugasan | Tidak ada | Sistem melakukan pengecekan pembayaran DP melalui gateway. apabila DP belum |

| No | Tahapan Proses Bisnis | Kondisi Existing | Kondisi *Future* |
|---|---|---|---|
| | | | dibayar, sistem memblokir *assign consultant* hingga konfirmasi pembayaran diterima |
| 3 | Penugasan Consultant | PM menugaskan Consultant langsung tanpa validasi kondisi proyek | PM melakukan *assign consultant* di sistem hanya setelah gateway validasi DP terpenuhi |
| 4 | Pelaksanaan Pekerjaan | Consultant mengeksekusi pekerjaan TP-Doc dan melakukan *update* progres secara informal melalui WhatsApp | Consultant melakukan *update* status *milestone* langsung di sistem, setiap *update* otomatis dicatat sebagai *log compliance* oleh Sistem |
| 5 | *Monitoring* Progres | PM melakukan *monitoring* manual menggunakan Excel, WhatsApp, dan *email* secara terpisah | Sistem mencatat setiap perubahan status *milestone* secara otomatis, PM melakukan review dan rating *milestone* melalui sistem |
| 6 | Validasi Penyelesaian | Tidak ada validasi formal. PM membuat rekap manual progres ke direksi | Sistem memvalidasi melalui gateway apakah seluruh *milestone* berstatus Done. Apabila belum, |



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| No | Tahapan Proses Bisnis | Kondisi Existing | Kondisi *Future* |
|---|---|---|---|
| | | | Consultant melanjutkan *update* hingga seluruh *milestone* selesai |
| 7 | Penyelesaian Proyek | CEO menerima laporan dari PM, dilanjutkan *delivery* ke klien | PM menjalankan Mark Project Completed di sistem, sistem otomatis mengubah status proyek menjadi Completed |
| 8 | Penerbitan Invoice | Tidak ada mekanisme otomatis | Sistem otomatis memicu *Trigger* Invoice Final segera setelah proyek berstatus Completed |
| 9 | Penilaian Kinerja | CEO melakukan evaluasi subjektif tanpa standar KPI yang terukur | Sistem otomatis menghitung KPI seluruh Consultant yang terlibat menggunakan metode *Weighted Scoring Method* (WSM). CEO menetapkan KPI berdasarkan hasil perhitungan sistem |

Sumber: Olahan Peneliti

Berdasarkan perbandingan pada Tabel 4.1, terdapat tiga perubahan struktural utama yang dihasilkan oleh penerapan sistem. Pertama, tahapan inisiasi proyek yang sebelumnya dilakukan secara informal oleh CEO kini diformalkan melalui

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

mekanisme *handover* terstruktur yang melibatkan COO dan pencatatan otomatis di sistem. Kedua, dua tahapan validasi baru diperkenalkan pada proses *future* yang tidak ada pada proses *existing*, yaitu validasi pembayaran DP sebelum penugasan Consultant dan validasi penyelesaian seluruh *milestone* sebelum proyek dapat ditutup, sehingga kondisi-kondisi kritis dikendalikan oleh sistem secara pasti. Ketiga, tiga tahapan yang sebelumnya bersifat manual dan tidak terintegrasi seperti *monitoring* progres, penerbitan invoice, dan penilaian kinerja digantikan oleh mekanisme otomatis yang berjalan sebagai bagian dari *workflow* sistem, sehingga seluruh siklus proyek dari inisiasi hingga penilaian kinerja terekam dalam satu platform yang terpadu.

### 4.1.4 Identifikasi Permasalahan dan Tema Kebutuhan

Berdasarkan hasil analisis perbandingan proses bisnis eksisting dan *future* serta hasil wawancara dan observasi (Lampiran 1 dan Lampiran 2), teridentifikasi lima permasalahan utama yang menjadi akar penyebab ketidakefisienan operasional pada pengelolaan proyek di perusahaan. Setiap permasalahan dipetakan terhadap solusi yang diharapkan dan tema kebutuhan yang menjadi landasan perancangan sistem.

Tabel 4. 2 Pemetaan Tema Permasalahan ke Tema Kebutuhan

| No | Tema Permasalahan | Solusi yang Diharapkan | Tema Kebutuhan |
|---|---|---|---|
| 1 | *Handover* proyek dilakukan secara informal tanpa pencatatan formal di sistem | Sistem menyediakan mekanisme konversi *handover* yang disetujui menjadi *record* proyek secara otomatis disertai penugasan PM yang terdokumentasi | Pengelolaan *lifecycle* proyek berbasis *workflow* |
| 2 | Tidak ada validasi pembayaran DP | Sistem memvalidasi status pembayaran DP klien dan memblokir | Pengelolaan *lifecycle* proyek berbasis *workflow* |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| No | Tema Permasalahan | Solusi yang Diharapkan | Tema Kebutuhan |
|---|---|---|---|
| | sebelum Consultant ditugaskan | penugasan Consultant apabila DP belum dikonfirmasi | |
| 3 | Pencatatan progres pekerjaan tersebar di tiga media tidak terintegrasi (Excel, WhatsApp, dan *email*) | Consultant melakukan *update* status *milestone* langsung di sistem, setiap perubahan tercatat otomatis sebagai *log compliance* yang dapat ditelusuri | *Monitoring* progres terpusat dengan *audit trail* otomatis |
| 4 | Tidak ada validasi formal sebelum proyek dinyatakan selesai | Sistem memvalidasi bahwa seluruh *milestone* berstatus Done sebelum PM dapat menjalankan aksi penyelesaian proyek | Pengelolaan *lifecycle* proyek berbasis *workflow* |
| 5 | Penilaian kinerja Consultant dilakukan secara subjektif tanpa standar KPI yang terukur | Sistem menghitung KPI seluruh Consultant secara otomatis menggunakan metode *Weighted Scoring Method* berbasis data aktivitas proyek yang sudah terekam di sistem | Penilaian kinerja berbasis KPI dengan perhitungan otomatis |

Sumber: Olahan Peneliti

Berdasarkan pemetaan pada Tabel 4.2, lima permasalahan yang teridentifikasi bermuara pada tiga tema kebutuhan utama. Tema pertama adalah pengelolaan *lifecycle* proyek berbasis *workflow* yang mencakup tiga permasalahan sekaligus, yaitu ketiadaan formalisasi inisiasi proyek, ketiadaan validasi pembayaran sebelum penugasan, dan ketiadaan validasi penyelesaian sebelum proyek ditutup. Tema kedua adalah *monitoring* progres terpusat dengan *audit trail* otomatis sebagai solusi

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

atas pencatatan progres yang selama ini tersebar di tiga media berbeda tanpa integrasi. Tema ketiga adalah penilaian kinerja berbasis KPI dengan perhitungan otomatis sebagai solusi atas evaluasi kinerja yang selama ini dilakukan secara subjektif oleh CEO tanpa dasar data yang terukur. Ketiga tema kebutuhan ini menjadi acuan utama dalam perumusan Kebutuhan Fungsional dan Kebutuhan Nonfungsional sistem.

### 4.1.5 Analisis Pengguna Sistem

Sistem yang akan dikembangkan harus mengakomodasi lima peran pengguna dengan kebutuhan informasi dan kewenangan yang berbeda-beda. Pemetaan peran ini didasarkan pada struktur organisasi perusahaan dan pola interaksi antar peran dalam proses bisnis *existing*. Setiap peran memiliki tanggung jawab spesifik yang harus didukung oleh fitur sistem yang relevan, sekaligus harus dibatasi aksesnya terhadap fitur dan data yang tidak menjadi kewenangannya untuk menjaga keamanan dan kerahasiaan informasi.

CEO sebagai pengambil keputusan strategis membutuhkan visibilitas penuh terhadap semua proyek dan kinerja seluruh tim, sekaligus memegang kewenangan dalam pengelolaan parameter KPI (indikator, target, bobot, dan rumus perhitungan). COO sebagai operator harian membutuhkan akses operasional yang mendalam terhadap semua proyek dan tindakan korektif. Project Manager membutuhkan akses operasional yang mendalam terhadap proyek-proyek yang dipimpinnya, termasuk kemampuan untuk mengelola *task*, mencatat progres, dan memberikan penilaian terhadap Consultant dalam timnya. Consultant sebagai pelaksana pekerjaan hanya memiliki akses terbatas pada *task* yang ditugaskan kepadanya dan KPI pribadinya. Selain empat peran fungsional tersebut, sistem juga membutuhkan peran Superadmin sebagai administrator teknis yang bertanggung jawab atas pengelolaan akun pengguna dan konfigurasi sistem.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tabel 4. 3 Analisis Pengguna Sistem

| No | Peran | Tanggungjawab Utama | Kebutuhan Informasi | Kewenangan |
|---|---|---|---|---|
| 1 | CEO | Pengambilan keputusan strategis dan pengelolaan parameter KPI | Ringkasan status semua proyek, KPI tim, dan parameter KPI | Lihat semua dan Kelola KPI dan laporan kinerja |
| 2 | COO | Pengelolaan operasional harian | Detail proyek, KPI, dan tindakan korektif | *Full access* operasional |
| 3 | Project Manager | Pengelolaan proyek yang dipimpin | Detail proyek, progres *task*, KPI tim | CRUD untuk proyek sendiri |
| 4 | Consultant | Pelaksanaan *task* | *Task* yang ditugaskan dan KPI pribadi | *Update* progres *task* pribadi |
| 5 | Superadmin | Administrasi sistem | Data master dan konfigurasi | *Full access* teknis |

Sumber: Olahan Peneliti

Pada Tabel 4.3 disajikan analisis lima peran pengguna sistem beserta tanggung jawab, kebutuhan informasi, dan kewenangan masing-masing. Tabel tersebut menunjukkan bahwa setiap peran memiliki kebutuhan dan kewenangan yang spesifik sehingga sistem harus mengimplementasikan mekanisme RBAC untuk membatasi akses sesuai peran. Analisis ini menjadi dasar dalam perancangan matriks hak akses dan implementasi otorisasi pada sistem.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

### 4.1.6 Analisis KPI

Analisis KPI dilakukan untuk merumuskan indikator-indikator kinerja yang akan digunakan dalam sistem penilaian karyawan di perusahaan. Berdasarkan hasil wawancara dengan CEO selaku pemegang kewenangan strategis dalam penilaian kinerja (Lampiran 3), dirumuskan empat dimensi KPI utama yang akan menjadi dasar perhitungan skor kinerja Consultant menggunakan metode *Weighted Scoring Method* (WSM). Setiap dimensi KPI dirancang dengan prinsip *Specific, Measurable, Achievable, Relevant, dan Time-bound* (SMART) agar dapat dihitung secara objektif berdasarkan data progres proyek yang tercatat di sistem. Penentuan bobot setiap dimensi KPI dilakukan oleh CEO berdasarkan tingkat kepentingan masing-masing aspek dalam mendukung keberhasilan proyek. Dimensi *Task Completion* diberi bobot tertinggi sebesar 35% karena kemampuan menyelesaikan tugas merupakan tanggung jawab utama Consultant. Dimensi *Timeliness* diberi bobot 25% sebagai cerminan disiplin waktu yang menjadi faktor kritis dalam siklus proyek dengan tenggat pelaporan ke otoritas pajak. Dimensi *Output Quality* juga diberi bobot 25% untuk menjamin kualitas dokumen proyek sesuai standar profesional. Dimensi *Update Compliance* diberi bobot 15% sebagai indikator kedisiplinan Consultant dalam memperbarui progres pekerjaan di sistem. Total bobot keempat dimensi adalah 100%, sesuai dengan ketentuan normalisasi bobot.

Tabel 4. 4 Daftar Dimensi KPI dan Bobotnya

| Kode | Dimensi KPI | Bobot | Deskripsi | Sumber Data | Mekanisme Pengisian |
|---|---|---|---|---|---|
| KPI-01 | *Task Completion* | 35% | Persentase *task* yang diselesaikan dari total task yang ditugaskan | Data progres *task* | Otomatis oleh sistem dari status *milestone* |

| Kode | Dimensi KPI | Bobot | Deskripsi | Sumber Data | Mekanisme Pengisian |
|---|---|---|---|---|---|
| KPI-02 | *Timeliness* | 25% | Persentase *task* yang diselesaikan tepat waktu sesuai target date | Data progres *task* vs target *date* | Otomatis oleh sistem dari selisih target *date* dan tanggal selesai |
| KPI-03 | *Output Quality* | 25% | Rata-rata skor *review* hasil pekerjaan oleh PM (skala 1-5) | Data *review task* | Manual oleh PM saat menilai *milestone* selesai |
| KPI-04 | *Update Compliance* | 15% | Persentase *task* yang diperbarui progresnya secara berkala | Data *update history* | Otomatis oleh sistem dari frekuensi pembaruan progres |

Sumber: Olahan Peneliti

Berdasarkan Tabel 4.4, mekanisme pengisian data tiap dimensi KPI dirancang berbeda sesuai karakteristik aspek yang dinilai. Tiga dari empat dimensi (*Task Completion, Timeliness, dan Update Compliance*) dihitung secara otomatis oleh sistem berdasarkan data aktivitas yang tercatat selama eksekusi proyek, sehingga objektivitas perhitungan dapat dijamin karena tidak bergantung pada *input* manual. Dimensi keempat, yaitu *Output Quality*, diisi secara manual oleh Project Manager dalam bentuk *rating* skala 1–5 saat memvalidasi *milestone* yang telah diselesaikan oleh Consultant. Pemisahan mekanisme ini memastikan bahwa aspek kuantitatif



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

dinilai oleh sistem berdasarkan data aktual, sementara aspek kualitatif yang memerlukan penilaian profesional tetap dinilai oleh PM sebagai atasan langsung.

### 4.1.7 Analisis Kebutuhan Fungsional Sistem

Kebutuhan fungsional sistem dirumuskan berdasarkan tema kebutuhan yang teridentifikasi dan analisis pengguna sistem. Kebutuhan fungsional menjelaskan apa yang harus dilakukan oleh sistem untuk menjawab kebutuhan setiap peran pengguna. Setiap kebutuhan fungsional dirumuskan dengan kode unik agar dapat ditelusuri kembali pada tahap implementasi dan pengujian.

Kebutuhan fungsional dikelompokkan menjadi delapan modul utama berdasarkan domain fungsionalitas, yaitu modul autentikasi, modul pengelolaan proyek, modul pencatatan progres, modul deteksi keterlambatan, modul pengelolaan KPI, dan modul *dashboard* serta laporan. Pengelompokan ini bertujuan untuk memudahkan tahap implementasi secara modular sehingga setiap modul dapat dikembangkan, diuji, dan dievaluasi secara independen sebelum diintegrasikan menjadi sistem yang utuh.

Setiap kebutuhan fungsional ditulis dengan struktur yang konsisten mencakup kode, deskripsi kebutuhan, dan peran pengguna yang berkepentingan. Pendekatan ini memudahkan komunikasi antara analis dan pengembang serta memastikan bahwa implementasi sesuai dengan kebutuhan yang telah disepakati. Daftar lengkap kebutuhan fungsional disajikan pada tabel berikut.

Tabel 4. 5 Daftar Kebutuhan Fungsional Sistem

| Kode | Modul | Deskripsi Kebutuhan | Pengguna |
|---|---|---|---|
| KF-01 | Autentikasi | Sistem dapat melakukan autentikasi pengguna berdasarkan *username* dan *password* | Semua peran |



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| Kode | Modul | Deskripsi Kebutuhan | Pengguna |
|---|---|---|---|
| KF-02 | Autentikasi | Sistem dapat membatasi akses fitur berdasarkan peran pengguna (RBAC) | Semua peran |
| KF-03 | Pengelolaan Proyek | Sistem dapat mengelola data proyek (tambah, ubah, hapus, lihat) | COO, PM, Superadmin |
| KF-04 | Pengelolaan Proyek | Sistem dapat mengelola penugasan Consultant ke proyek | COO, PM, Superadmin |
| KF-05 | Pencatatan Progres | Sistem dapat mencatat progres pekerjaan per *task* dan *milestone* | PM, Consultant |
| KF-06 | Pencatatan Progres | Sistem dapat mengelola status *workflow* proyek (*Awaiting Consultant, In Progress, On Hold, Completed, Cancelled*) dan status *workflow milestone* (*Pending, In Progress, Done, Blocked*) | PM, COO |
| KF-07 | Deteksi Keterlambatan | Sistem dapat membandingkan progres aktual dengan jadwal rencana | Sistem (otomatis) |
| KF-08 | Deteksi Keterlambatan | Sistem dapat mengirimkan *alert* otomatis ketika terdeteksi keterlambatan | Sistem (otomatis) |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| Kode | Modul | Deskripsi Kebutuhan | Pengguna |
|---|---|---|---|
| KF-09 | Pengelolaan KPI | Sistem dapat mengelola parameter KPI (indikator, target, bobot, rumus) | CEO, Superadmin |
| KF-10 | Pengelolaan KPI | Sistem dapat menghitung pencapaian KPI secara otomatis dari data progres | Sistem (otomatis) |
| KF-11 | *Dashboard Monitoring* | Sistem dapat menampilkan *dashboard monitoring* proyek secara *real-time* | CEO, COO, PM |
| KF-12 | *Dashboard* KPI | Sistem dapat menampilkan ringkasan KPI tim dan individu | CEO, COO, PM |
| KF-13 | Laporan Proyek | Sistem dapat menghasilkan laporan progres proyek per periode | CEO, COO, PM |
| KF-14 | Laporan KPI | Sistem dapat menghasilkan laporan penilaian KPI per individu dan periode | CEO, COO |
| KF-15 | Pengelolaan Akun | Sistem dapat mengelola akun pengguna (tambah, ubah, hapus, reset password) | Superadmin |

Sumber: Olahan Peneliti

Pada Tabel 4.5 disajikan lima belas kebutuhan fungsional sistem yang dikelompokkan ke dalam delapan modul fungsional. Tabel tersebut menunjukkan bahwa setiap kebutuhan fungsional memiliki kode unik, deskripsi yang spesifik, dan pengguna yang berkepentingan, sehingga memudahkan tahap implementasi

dan pengujian. Daftar kebutuhan fungsional ini menjadi acuan utama dalam perancangan *use case diagram* dan implementasi sistem.

### 4.1.8 Analisis Kriteria Penerimaan Pengguna

Kriteria Penerimaan Pengguna (*Acceptance Criteria*) merupakan rumusan kondisi terukur yang harus dipenuhi oleh sistem agar dapat dinyatakan memenuhi kebutuhan fungsional yang telah didefinisikan. Berbeda dengan Karakteristik Fungsional yang menjelaskan *apa* yang sistem harus lakukan, Kriteria Penerimaan menjelaskan *bagaimana* sistem dinyatakan berhasil melakukannya secara objektif. Rumusan kriteria ini menjadi dasar utama dalam tahap pengujian sistem, sehingga setiap *test case* yang dieksekusi dapat ditelusuri kembali ke kriteria yang menjadi acuan keberhasilannya. Pendekatan ini memastikan bahwa pengujian sistem bukan sekadar eksekusi skenario teknis, melainkan verifikasi yang terstruktur terhadap ekspektasi pengguna yang telah disepakati sebelum implementasi dimulai.

Rumusan Kriteria Penerimaan pada penelitian ini mengadopsi format *Given-When-Then* yang lazim digunakan pada praktik *behavior-driven development*. Format ini memisahkan tiga elemen kriteria secara eksplisit, yaitu *Given* sebagai kondisi awal sistem yang harus terpenuhi, *When* sebagai aksi yang dilakukan oleh pengguna atau sistem, dan *Then* sebagai hasil terukur yang diharapkan terjadi. Pemilihan format ini memberikan tiga manfaat, yaitu kejelasan kondisi prasyarat yang harus disiapkan sebelum uji coba, eksplisitnya aksi yang menjadi pemicu pengujian, serta hasil yang dapat diverifikasi secara objektif tanpa interpretasi subjektif.

Kriteria Penerimaan dikelompokkan menjadi dua kategori. Kategori pertama adalah Kriteria Positif yang menguji *happy path*, yaitu jalur eksekusi normal ketika seluruh prasyarat terpenuhi dan sistem diharapkan menghasilkan *output* sesuai ekspektasi. Kategori kedua adalah Kriteria Negatif yang menguji *defensive behavior*, yaitu respons sistem ketika prasyarat tidak terpenuhi atau pengguna mencoba aksi yang seharusnya tidak diperbolehkan. Pengelompokan ini memastikan bahwa sistem tidak hanya bekerja benar pada kondisi ideal, tetapi juga memberikan respons yang aman dan informatif pada kondisi yang menyimpang dari ekspektasi.



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Mengingat sistem perusahaan memiliki lima belas Karakteristik Fungsional dengan kompleksitas yang berbeda-beda, perumusan Kriteria Penerimaan difokuskan pada delapan modul fungsional inti yang merepresentasikan kontribusi utama penelitian. Delapan modul tersebut mencakup autentikasi dan otorisasi, pengelolaan proyek, pencatatan progres, pengelolaan status *workflow*, perhitungan KPI, *dashboard monitoring*, pengelolaan akun pengguna, dan integrasi *cross-module*. Karakteristik Fungsional lain yang bersifat *auxiliary* seperti deteksi keterlambatan dan laporan terformat dievaluasi melalui pengujian fungsional umum tanpa Kriteria Penerimaan terpisah, mengingat modul-modul tersebut bersifat pelengkap terhadap modul inti yang menjadi kontribusi utama penelitian.

Tabel 4. 6 Kriteria Penerimaan Pengguna Sistem

| Kode | Modul | Tipe | Kriteria Penerimaan | Referensi |
|---|---|---|---|---|
| KP-01 | Autentikasi | Positif | Given pengguna terdaftar dengan kredensial valid When pengguna *submit form login* dengan *username* dan *password* yang benar Then sistem mengembalikan JWT dengan *payload role* pengguna dan mengarahkan ke *dashboard* sesuai peran | KF-01 |
| KP-02 | Autentikasi | Negatif | Given pengguna tidak terdaftar atau *password* salah When pengguna *submit form login* Then sistem mengembalikan respons HTTP 401 dengan pesan kesalahan tanpa membocorkan apakah *username* valid atau tidak | KF-01 |

| Kode | Modul | Tipe | Kriteria Penerimaan | Referensi |
|---|---|---|---|---|
| KP-03 | RBAC | Positif | Given pengguna terautentikasi dengan peran tertentu When pengguna mengakses *endpoint* yang sesuai dengan peran Then sistem memproses permintaan dan mengembalikan data yang relevan | KF-02 |
| KP-04 | RBAC | Negatif | Given pengguna terautentikasi dengan peran yang tidak memiliki *permission* When pengguna mengakses *endpoint* yang memerlukan *permission* khusus Then sistem mengembalikan respons HTTP 403 dengan kode UNAUTHORIZED_ROLE | KF-02 |
| KP-05 | Pengelolaan Proyek | Positif | Given *handover* memo berstatus APPROVED dan PM valid When COO melakukan konversi *handover* ke proyek Then sistem membuat proyek baru dengan status *Awaiting Consultant*, menyalin *milestone* dari *template*, dan mencatat audit log inisialisasi | KF-03 |
| KP-06 | Pengelolaan Proyek | Negatif | Given proyek sudah berstatus terminal (*Completed* atau *Cancelled*) | KF-03, KF-06 |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| Kode | Modul | Tipe | Kriteria Penerimaan | Referensi |
|---|---|---|---|---|
| | | | When pengguna mencoba menambah Consultant atau mengubah *milestone* Then sistem mengembalikan respons HTTP 409 dengan kode PROJECT_TERMINAL | |
| KP-07 | Penugasan Consultant | Positif | Given proyek berstatus *Awaiting Consultant* dan DP klien sudah PAID When PM menugaskan minimal satu Consultant dari departemen yang sesuai Then sistem memperbarui status proyek menjadi *In Progress* dan mencatat transisi audit | KF-04, KF-06 |
| KP-08 | Penugasan Consultant | Negatif | Given proyek berstatus *Awaiting Consultant* dan DP klien *UNPAID* When PM mencoba menugaskan Consultant Then sistem menolak permintaan dengan kode DP_UNPAID dan tidak melakukan transisi status | KF-04 |
| KP-09 | Pencatatan Progres | Positif | Given Consultant adalah pemilik *milestone* aktif When Consultant mengubah status *milestone* dari *In Progress* ke *Done* | KF-05 |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| Kode | Modul | Tipe | Kriteria Penerimaan | Referensi |
|---|---|---|---|---|
| | | | Then sistem mengisi kolom *completed_at* dengan *timestamp* saat ini, mencatat audit log, dan dimensi KPI *Timeliness* dapat dihitung pada periode aktif | |
| KP-10 | Pencatatan Progres | Negatif | Given transisi *milestone* tidak ada pada *matrix* transisi (misal dari *Done* ke Pending) When pengguna mencoba transisi tersebut Then sistem mengembalikan kode INVALID_TRANSITION tanpa mengubah *state* | KF-06 |
| KP-11 | *Workflow* Proyek | Positif | Given seluruh *milestone* proyek berstatus *Done* When PM melakukan aksi Mark Project *Completed* Then sistem mengubah status proyek menjadi *Completed*, mengisi end_date, memicu invoice final, dan menjalankan *auto-compute snapshot* KPI *preliminary* | KF-06, KF-10 |
| KP-12 | *Workflow* Proyek | Negatif | Given satu atau lebih *milestone* belum berstatus *Done* When PM mencoba menyelesaikan proyek Then sistem menolak permintaan dengan kode | KF-06 |



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| Kode | Modul | Tipe | Kriteria Penerimaan | Referensi |
|---|---|---|---|---|
| | | | *MILESTONES*_INCOMPLETE dan tombol Mark Project *Completed* tetap dalam keadaan *disabled* di antarmuka | |
| KP-13 | Pengelolaan KPI | Positif | Given CEO terautentikasi dan total bobot empat dimensi sama dengan satu dengan toleransi nol koma nol satu When CEO menyimpan konfigurasi bobot baru Then sistem menyimpan baris konfigurasi baru dan menjadikannya konfigurasi aktif untuk perhitungan KPI berikutnya | KF-09 |
| KP-14 | Pengelolaan KPI | Negatif | Given total bobot empat dimensi tidak sama dengan satu When CEO menyimpan konfigurasi Then sistem menolak penyimpanan dengan respons HTTP 400 dan menampilkan nilai aktual total bobot saat ini | KF-09 |
| KP-15 | Perhitungan KPI | Positif | Given Consultant memiliki minimal satu *milestone* pada periode yang sedang dihitung When sistem menjalankan *auto-compute* KPI setelah proyek *Completed* | KF-10 |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| Kode | Modul | Tipe | Kriteria Penerimaan | Referensi |
|---|---|---|---|---|
| | | | Then sistem menghasilkan *snapshot* dengan empat capaian dimensi, total skor WSM, dan *finalized_at* bernilai NULL (*preliminary*) | |
| KP-16 | Perhitungan KPI | Negatif | Given *snapshot* KPI Consultant sudah berstatus *finalized* When sistem menjalankan auto-*compute* KPI ulang Then sistem mengembalikan respons skipped: ALREADY_FINALIZED dan tidak menyentuh kolom *finalized_at* | KF-10 |
| KP-17 | *Dashboard Monitoring* | Positif | Given pengguna terautentikasi dengan peran CEO atau COO When pengguna mengakses *dashboard* utama Then sistem menampilkan ringkasan KPI organisasi, distribusi Consultant per kategori kinerja, dan tren periode terakhir | KF-11, KF-12 |
| KP-18 | Pengelolaan Akun | Positif | Given pengguna terautentikasi dengan peran Superadmin When Superadmin membuat akun baru dengan kredensial valid Then sistem menyimpan akun | KF-15 |



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| Kode | Modul | Tipe | Kriteria Penerimaan | Referensi |
|---|---|---|---|---|
| | | | dengan *password* yang sudah di-hash menggunakan bcrypt dan akun dapat digunakan untuk *login* | |
| KP-19 | Pengelolaan Akun | Negatif | Given pengguna terautentikasi dengan peran selain Superadmin When pengguna mencoba mengakses *endpoint* pengelolaan akun Then sistem mengembalikan respons HTTP 403 karena *permission USER*_MANAGE tidak dimiliki | KF-02, KF-15 |
| KP-20 | Integrasi Cross-Module | Positif | Given proyek mencapai status *Completed* When transisi proyek ke *Completed* dieksekusi Then sistem melakukan empat tindakan berurutan: pengisian end_date, pemicu *invoice* final, *auto-compute snapshot* KPI untuk seluruh Consultant, dan pencatatan *activity* log lintas modul | KF-06, KF-10 |

Sumber: Olahan Peneliti

Pada Tabel 4.6 memuat dua puluh Kriteria Penerimaan yang terdistribusi dalam sebelas kriteria positif dan sembilan kriteria negatif. Distribusi yang relatif seimbang ini mencerminkan filosofi pengujian yang dianut pada penelitian ini, yaitu sistem dinyatakan berhasil bukan hanya ketika menjalankan jalur normal

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

dengan benar, tetapi juga ketika menolak permintaan yang menyimpang dengan respons yang aman dan informatif. Kriteria penerimaan negatif yang berfokus pada validasi *precondition* seperti KP-08 (DP_UNPAID), KP-10 (INVALID_TRANSITION), KP-12 (*MILESTONES*_INCOMPLETE), dan KP-16 (ALREADY_FINALIZED) menunjukkan bahwa sistem dirancang dengan *defensive thinking* yang menyeluruh, sehingga integritas data tetap terjaga meskipun pengguna melakukan aksi yang tidak sesuai dengan kondisi yang dipersyaratkan.

Setiap Kriteria Penerimaan pada Tabel 4.6 akan menjadi acuan langsung pada perancangan *test case*. Setiap *test case* akan dipetakan ke kode KP yang sesuai, sehingga jejak verifikasi dapat ditelusuri dari Kebutuhan Fungsional melalui Kriteria Penerimaan ini hingga ke skenario pengujian aktual beserta bukti eksekusinya. Pemetaan tiga lapis ini menjadi bukti bahwa sistem yang dibangun tidak hanya menjalankan fitur yang dijanjikan, tetapi juga memenuhi kondisi keberhasilan yang telah didefinisikan sebelum implementasi dimulai.

### 4.1.9 Analisis Kebutuhan Nonfungsional Sistem

Kebutuhan non-fungsional sistem menjelaskan kualitas atau atribut yang harus dimiliki oleh sistem terkait dengan bagaimana sistem berperilaku, bukan apa yang dilakukan sistem. Kebutuhan non-fungsional sama pentingnya dengan kebutuhan fungsional karena menentukan tingkat kepuasan pengguna terhadap sistem secara keseluruhan. Sistem yang memenuhi semua kebutuhan fungsional namun gagal memenuhi kebutuhan non-fungsional seperti kinerja atau keamanan tidak akan memberikan nilai bisnis yang optimal bagi perusahaan.

Kebutuhan non-fungsional sistem dikelompokkan ke dalam enam aspek utama, yaitu kinerja, keamanan, ketersediaan, kompatibilitas, kemudahan penggunaan, dan skalabilitas. Aspek kinerja menentukan kecepatan respons sistem dalam menjalankan operasi-operasi penting seperti *login*, akses *dashboard*, dan perhitungan KPI. Aspek keamanan menentukan mekanisme perlindungan data sensitif seperti informasi penilaian kinerja yang bersifat rahasia. Aspek ketersediaan menentukan tingkat *uptime* sistem yang dibutuhkan agar tidak mengganggu operasional perusahaan.

Aspek kompatibilitas menentukan platform dan *browser* yang harus didukung oleh sistem agar dapat diakses oleh seluruh pengguna dengan berbagai jenis perangkat. Aspek kemudahan penggunaan menentukan tingkat intuitif antarmuka sistem agar dapat digunakan oleh pengguna dengan berbagai latar belakang teknis tanpa pelatihan yang intensif. Aspek skalabilitas menentukan kemampuan sistem untuk menangani peningkatan jumlah pengguna, proyek, dan data seiring dengan pertumbuhan perusahaan di masa depan.

Tabel 4. 7 Daftar Kebutuhan Nonfungsional Sistem

| Kode | Aspek | Deskripsi Kebutuhan | Target Pengukuran |
|---|---|---|---|
| KNF-01 | Kinerja | Sistem menampilkan *dashboard* dalam waktu singkat | *Response time* < 3 detik |
| KNF-02 | Kinerja | Sistem menjalankan perhitungan KPI dengan cepat | *Computation time* < 5 detik |
| KNF-03 | Keamanan | Sistem mengimplementasikan autentikasi berbasis *username* dan *password* | 100% *endpoint* terlindungi |
| KNF-04 | Keamanan | Sistem mengimplementasikan RBAC | Sesuai matriks hak akses |
| KNF-05 | Keamanan | Sistem menyimpan *password* dalam bentuk terenkripsi (*hashed*) | Penggunaan algoritma bcrypt |
| KNF-06 | Keandalan | Sistem menangani galat dan *input* tidak valid tanpa berhenti | Galat direspons terkontrol, sistem tetap berjalan |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| Kode | Aspek | Deskripsi Kebutuhan | Target Pengukuran |
|---|---|---|---|
| KNF-07 | Kompatibilitas | Sistem dapat diakses melalui *browser* modern | Chrome, Safari (versi 2 tahun terakhir) |
| KNF-08 | Kompatibilitas | Sistem responsif pada perangkat desktop dan *mobile* | *Layout* adaptif minimal 360px *width* |
| KNF-09 | Kemudahan Penggunaan | Sistem memiliki antarmuka yang intuitif dan mudah dipelajari | Skor SUS > 68 |
| KNF-10 | Skalabilitas | Sistem tetap stabil pada beban pengguna bersamaan | *Stress test* minimal 50 *concurrent users* |

Sumber: Olahan Peneliti

Pada Tabel 4.7 disajikan sepuluh kebutuhan non-fungsional sistem yang dikelompokkan ke dalam enam aspek kualitas. Tabel tersebut menunjukkan bahwa setiap kebutuhan non-fungsional memiliki target pengukuran yang spesifik dan dapat diuji secara objektif pada tahap pengujian sistem. Daftar kebutuhan non-fungsional ini akan menjadi acuan dalam tahap pengujian non-fungsional untuk memverifikasi bahwa sistem yang dibangun tidak hanya memenuhi kebutuhan fungsional, tetapi juga memberikan kualitas layanan yang sesuai dengan ekspektasi pengguna.

### 4.1.10 *Use Case Diagram*

*Use Case Diagram* disusun untuk memetakan interaksi antara aktor dengan sistem ke dalam bentuk fitur-fitur yang dapat diakses. Penyusunan diagram ini didasarkan pada hasil analisis pengguna dan proses bisnis *future* yang telah dirumuskan pada tahap analisis sebelumnya, sehingga aktor dan fitur sistem dapat dipetakan secara konsisten. Diagram ini menggunakan notasi *Unified Modeling Language* (UML) yang merupakan standar industri dalam pemodelan sistem berorientasi objek,

sehingga cakupan fungsionalitas sistem dapat dipahami secara cepat sebagai dasar perancangan, implementasi, dan komunikasi antar pemangku kepentingan proyek.

*Use Case Diagram* pada sistem ini melibatkan lima aktor utama, yaitu CEO, COO, Project Manager, Consultant, dan Superadmin, serta satu aktor sekunder berupa *Modul Invoice* yang merepresentasikan integrasi lintas modul untuk pemicu invoice final. Penamaan setiap *use case* menggunakan kata kerja aktif untuk menegaskan bahwa *use case* merepresentasikan fitur yang dapat dilakukan aktor terhadap sistem, bukan urutan alur proses. Sistem terdiri dari delapan belas use case utama yang dikelompokkan ke dalam autentikasi, pengelolaan proyek, eksekusi *milestone*, pengelolaan KPI, *monitoring*, dan administrasi sistem, ditambah delapan *sub-use-case* yang dieksekusi sistem secara otomatis sebagai bagian dari *use case* utama. Setiap aktor memiliki cakupan akses yang berbeda sesuai tanggung jawab dan kewenangannya, dengan CEO dan Superadmin memiliki cakupan paling luas sebagai pemegang kewenangan strategis dan administratif.

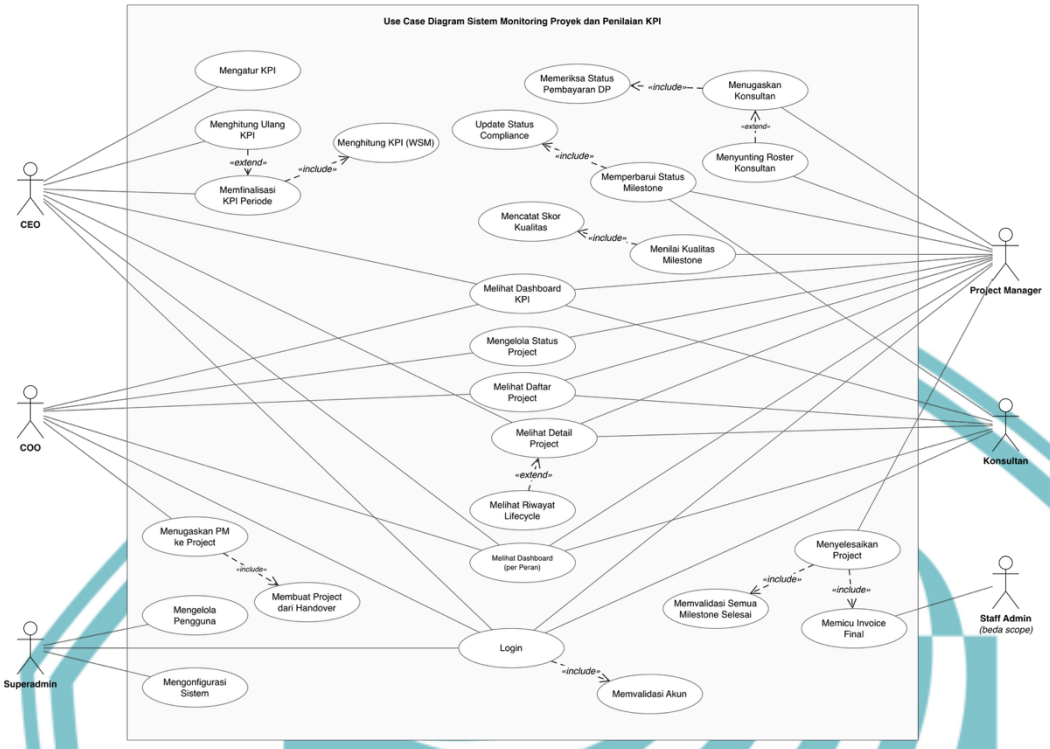
Hubungan antar *use case* dimodelkan menggunakan relasi *include* dan *extend*. Relasi *include* merepresentasikan aksi yang dieksekusi sistem secara otomatis dan wajib ketika *use case* utama dijalankan oleh aktor. Pada diagram ini terdapat delapan relasi *include*, yaitu *Login* dengan Memvalidasi Akun, Menugaskan PM ke Project dengan Membuat Project dari *Handover*, Menugaskan Consultant dengan Memeriksa Status Pembayaran DP, Memperbarui Status *Milestone* dengan Mencatat Kepatuhan Pembaruan, Menilai Kualitas *Milestone* dengan Mencatat Skor Kualitas, Menyelesaikan Project dengan Memvalidasi Semua *Milestone* Selesai dan Memicu *Invoice Final*, serta Memfinalisasi KPI Periode dengan Menghitung KPI yang menggunakan metode WSM, relasi *extend* merepresentasikan *use case* opsional yang diakses aktor melalui use case utamanya. Terdapat tiga relasi *extend* pada sistem ini, yaitu Menyunting Roster Consultant sebagai perluasan dari Menugaskan Consultant, Menghitung Ulang KPI sebagai perluasan dari Memfinalisasi KPI Periode, serta Melihat Riwayat *Lifecycle* sebagai perluasan dari Melihat Detail Project.



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta




Gambar 4. 3 *Use Case Diagram Monitoring* Proyek dan Penilaian KPI

Sumber: Olahan Peneliti

Pada Gambar 4.3 ditampilkan *Use Case Diagram* sistem yang akan dikembangkan. Diagram tersebut memvisualisasikan interaksi antara seluruh aktor dengan *use case* yang berada di dalam batas sistem, beserta relasi *include* dan *extend* yang menghubungkannya. *Use case* yang teridentifikasi pada diagram ini selanjutnya menjadi dasar perumusan kebutuhan fungsional sistem, sekaligus menjadi referensi utama bagi tim pengembang dalam tahap perancangan dan implementasi.

### 4.1.11 *Activity Diagram*

*Activity Diagram* disusun untuk menggambarkan alur langkah demi langkah dari setiap use case yang telah teridentifikasi pada *Use Case Diagram*. *Activity Diagram* menggunakan notasi *Unified Modeling Language* (UML) yang menggambarkan urutan aktivitas, percabangan keputusan, paralelisme proses, serta pembagian tanggung jawab antara pengguna dan sistem melalui partisi *swimlane*. Setiap *Activity Diagram* merupakan turunan langsung dari *use case* pada Gambar 4.3,

sehingga relasi antara fitur sistem dan alur eksekusinya dapat dipetakan secara satu-ke-satu.

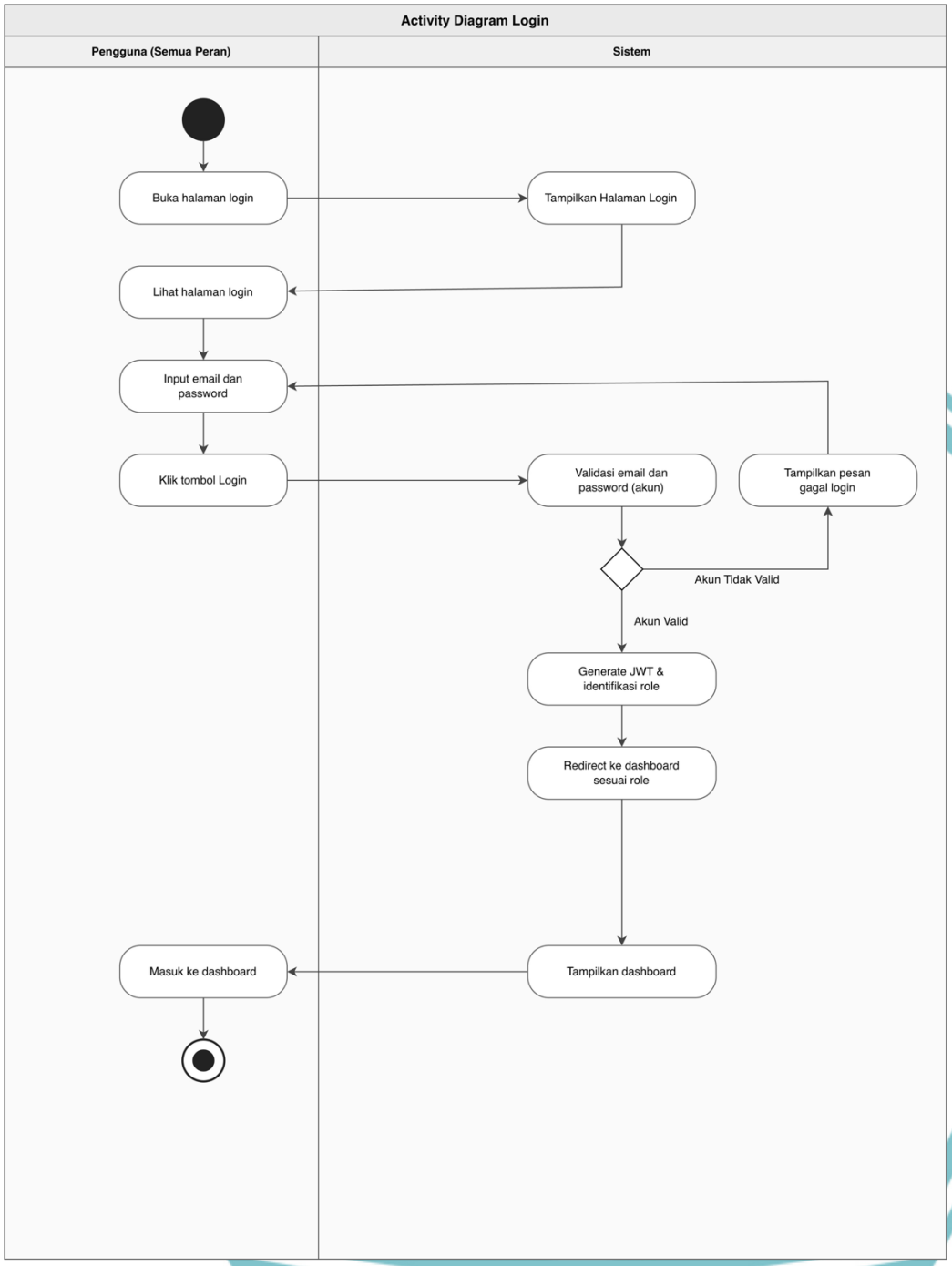
Berdasarkan delapan belas *use case* utama yang telah dirumuskan pada *Use Case Diagram*, disusun delapan belas *Activity Diagram* yang masing-masing merepresentasikan alur eksekusi satu *use case*. Pendekatan satu *Activity Diagram* untuk setiap *use case* dipilih agar seluruh fitur sistem memiliki dokumentasi alur yang lengkap dan konsisten, sehingga tim pengembang dapat mengacu pada satu sumber referensi yang seragam pada tahap implementasi. *Activity Diagram* untuk *sub-use-case* yang ter-*include* secara otomatis oleh sistem tidak disajikan secara terpisah, melainkan menjadi bagian dari *Activity Diagram use case* induknya. Setiap *Activity Diagram* disusun dengan dua partisi *swimlane*, yaitu partisi Pengguna yang merepresentasikan aksi aktor dan partisi Sistem yang merepresentasikan respon otomatis sistem.

Delapan belas *Activity Diagram* dikelompokkan mengikuti kategori *use case* pada Gambar 4.3. Pertama, kategori autentikasi mencakup satu *Activity Diagram* yaitu *Login* yang menggambarkan alur masuk pengguna ke sistem termasuk validasi akun. Kedua, kategori pengelolaan proyek mencakup empat *Activity Diagram* yaitu Menugaskan PM ke Project, Menugaskan Consultant, Menyunting Roster Consultant, dan Mengelola Status Project yang mencakup tiga aksi pengelolaan status berupa menjeda, melanjutkan, dan membatalkan proyek. Ketiga, kategori penyelesaian proyek mencakup satu *Activity Diagram* yaitu Menyelesaikan Project yang menggambarkan validasi seluruh *milestone* dan pemicuan *invoice final* secara otomatis oleh sistem. Keempat, kategori eksekusi *milestone* mencakup dua *Activity Diagram* yaitu Memperbarui Status *Milestone* dan Menilai Kualitas *Milestone*. Kelima, kategori pengelolaan KPI mencakup tiga *Activity Diagram* yaitu Mengatur KPI, Memfinalisasi KPI Periode, dan Menghitung Ulang KPI. Keenam, kategori *monitoring* mencakup lima *Activity Diagram* yaitu Melihat Daftar Project, Melihat Detail Project, Melihat Riwayat *Lifecycle*, Melihat *Dashboard* per Peran, dan Melihat *Dashboard* KPI. Terakhir, kategori administrasi sistem mencakup dua *Activity Diagram* yaitu Mengelola Pengguna dan Mengonfigurasi Sistem.



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

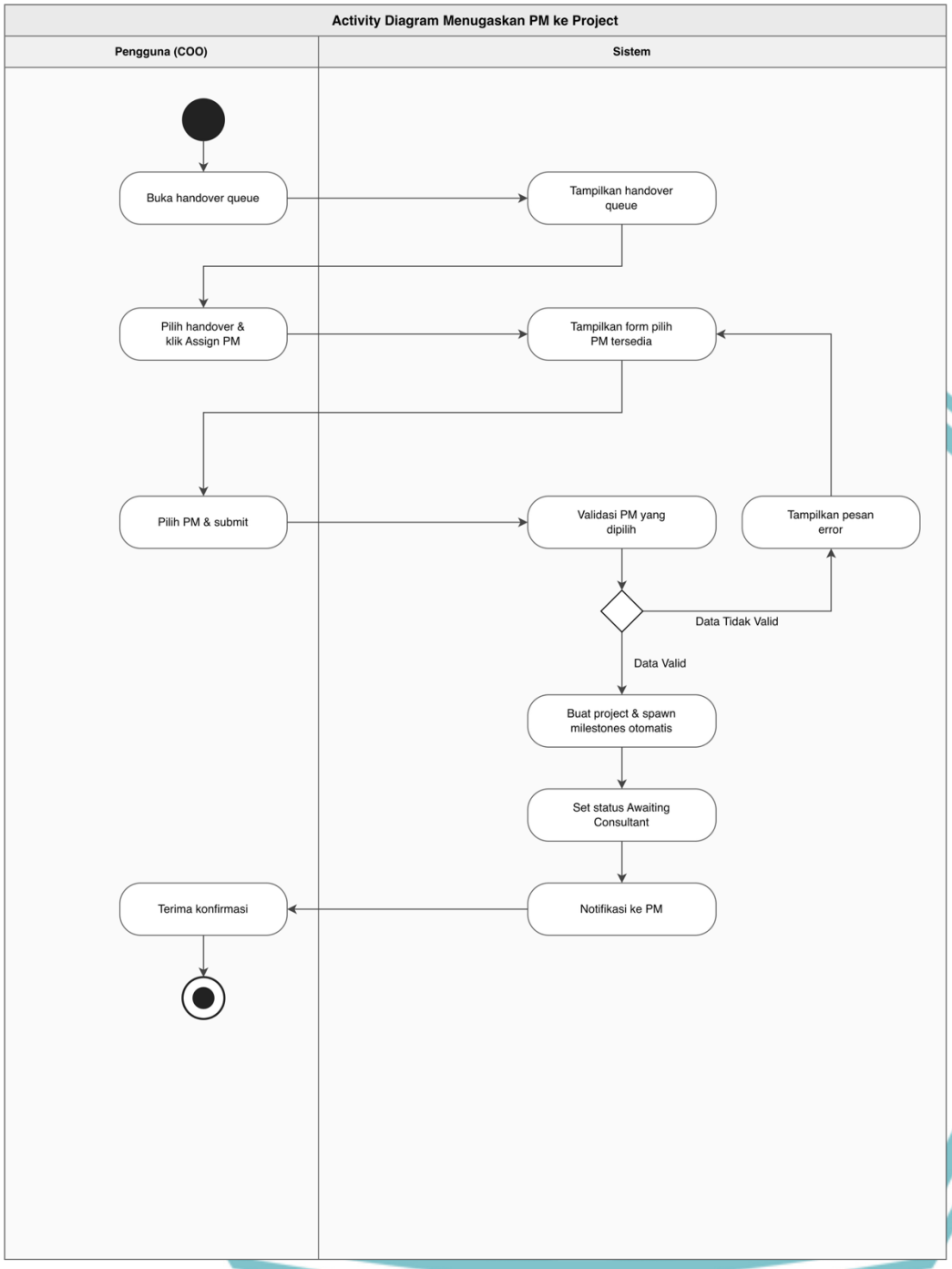



Gambar 4. 4 *Activity Diagram Login*

Sumber: Olahan Peneliti

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



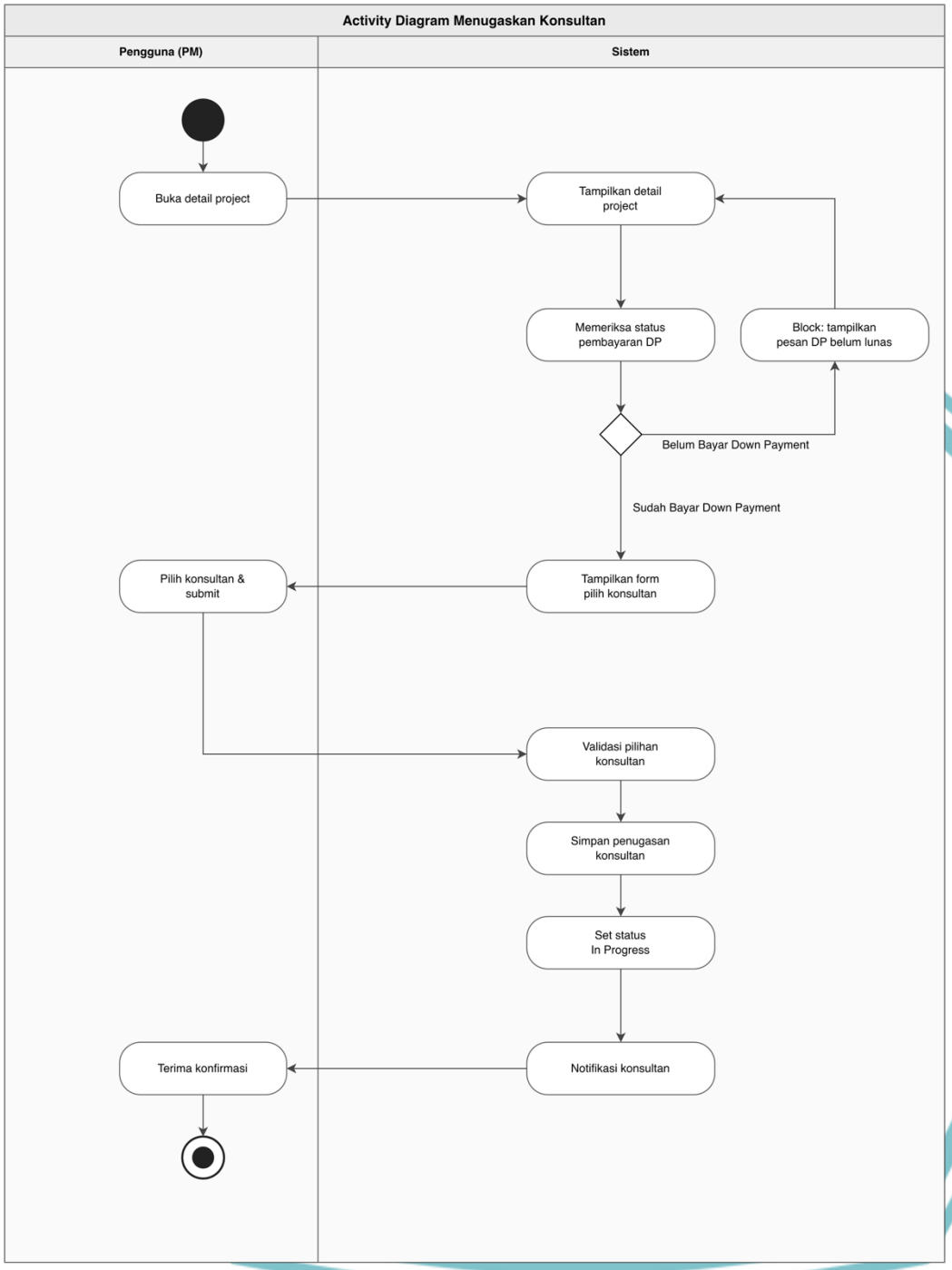

Gambar 4. 5 *Activity Diagram* Menugaskan PM ke Project

Sumber: Olahan Peneliti



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



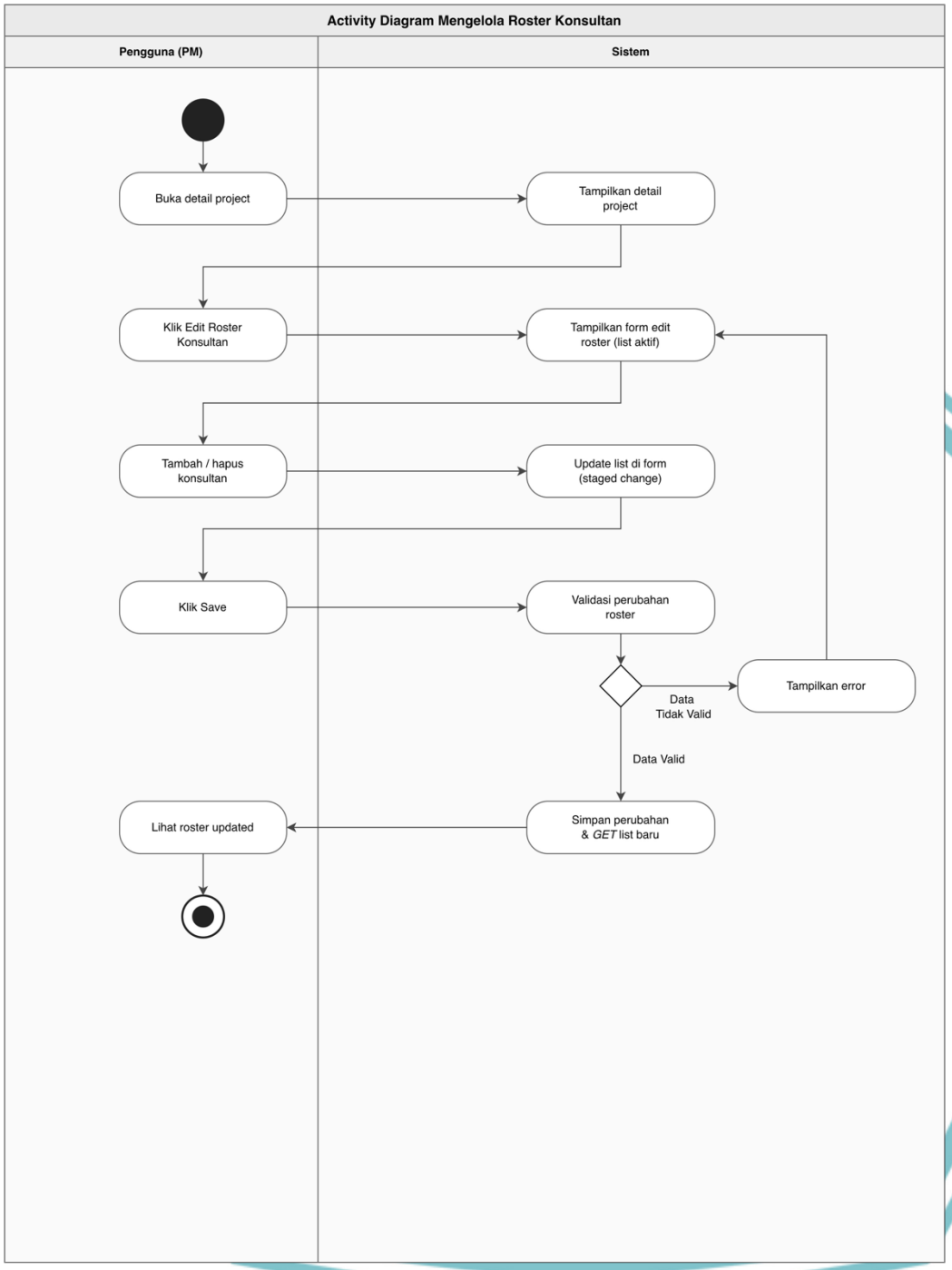

Gambar 4. 6 *Activity Diagram* Menugaskan Consultant

Sumber: Olahan Peneliti



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

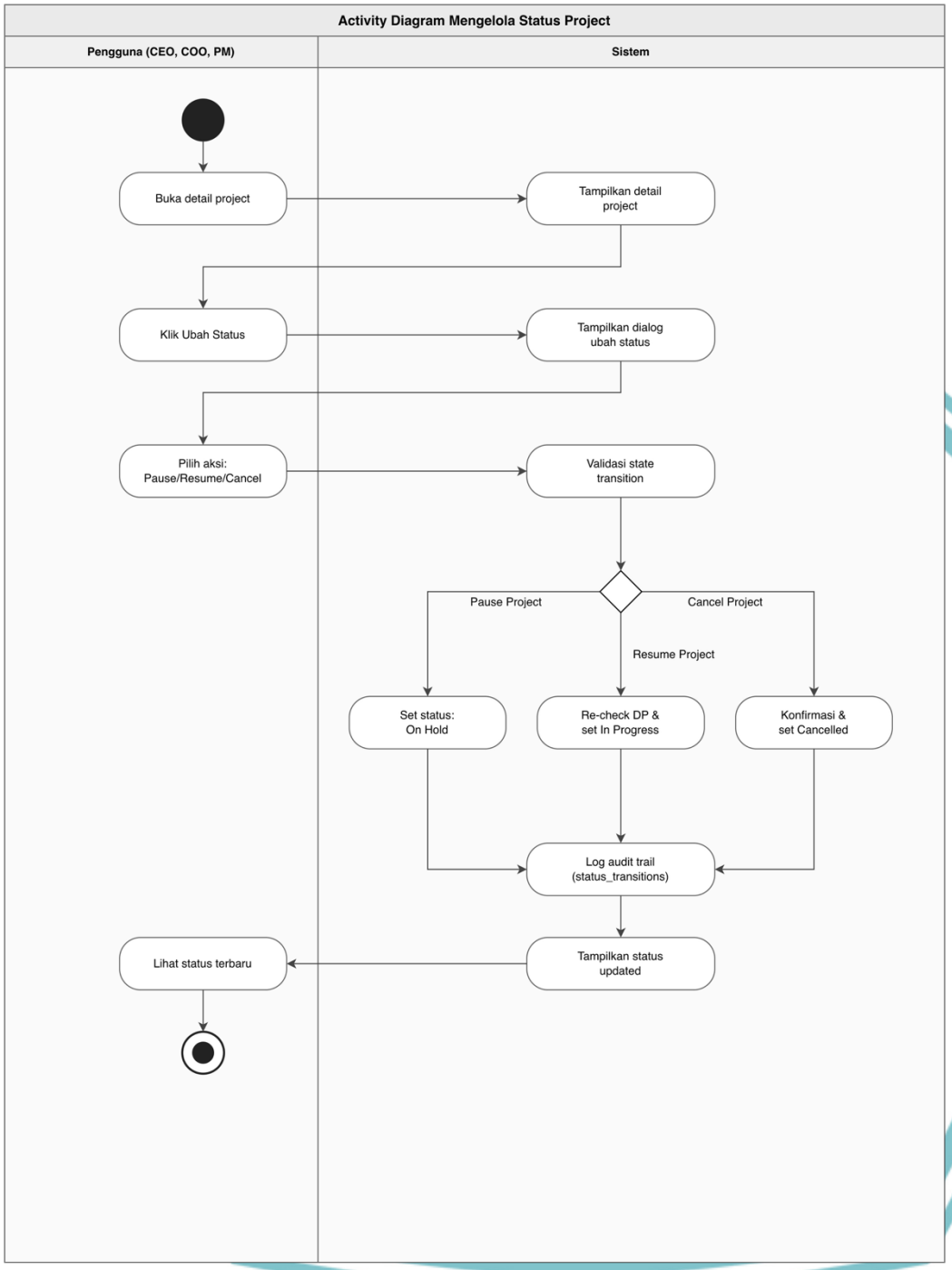



Gambar 4. 7 *Activity Diagram* Mengelola Roster Consultant

Sumber: Olahan Peneliti

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



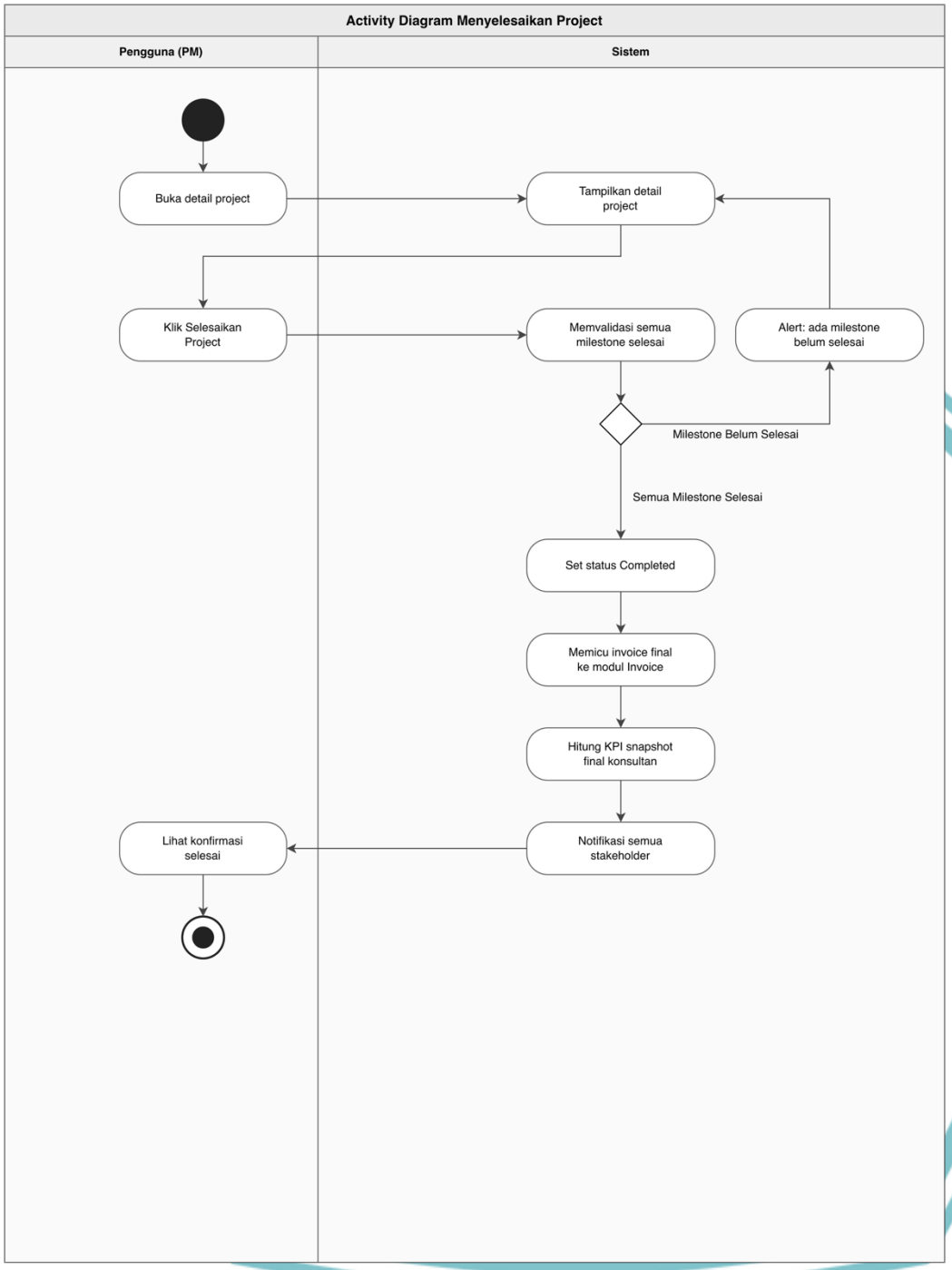

Gambar 4. 8 *Activity Diagram* Mengelola Status Project

Sumber: Olahan Peneliti



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

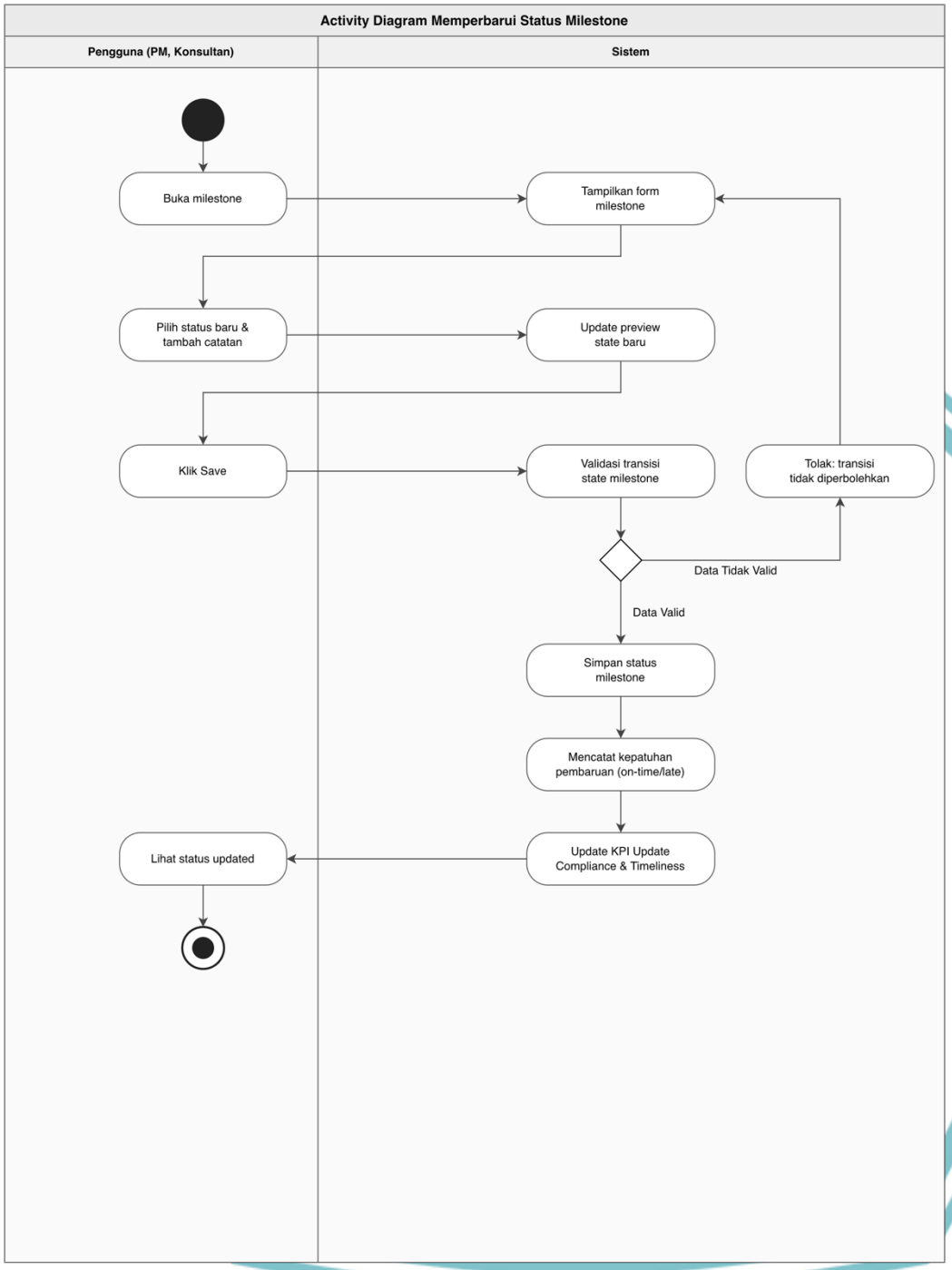



Gambar 4. 9 *Activity Diagram* Menyelesaikan Project

Sumber: Olahan Peneliti

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

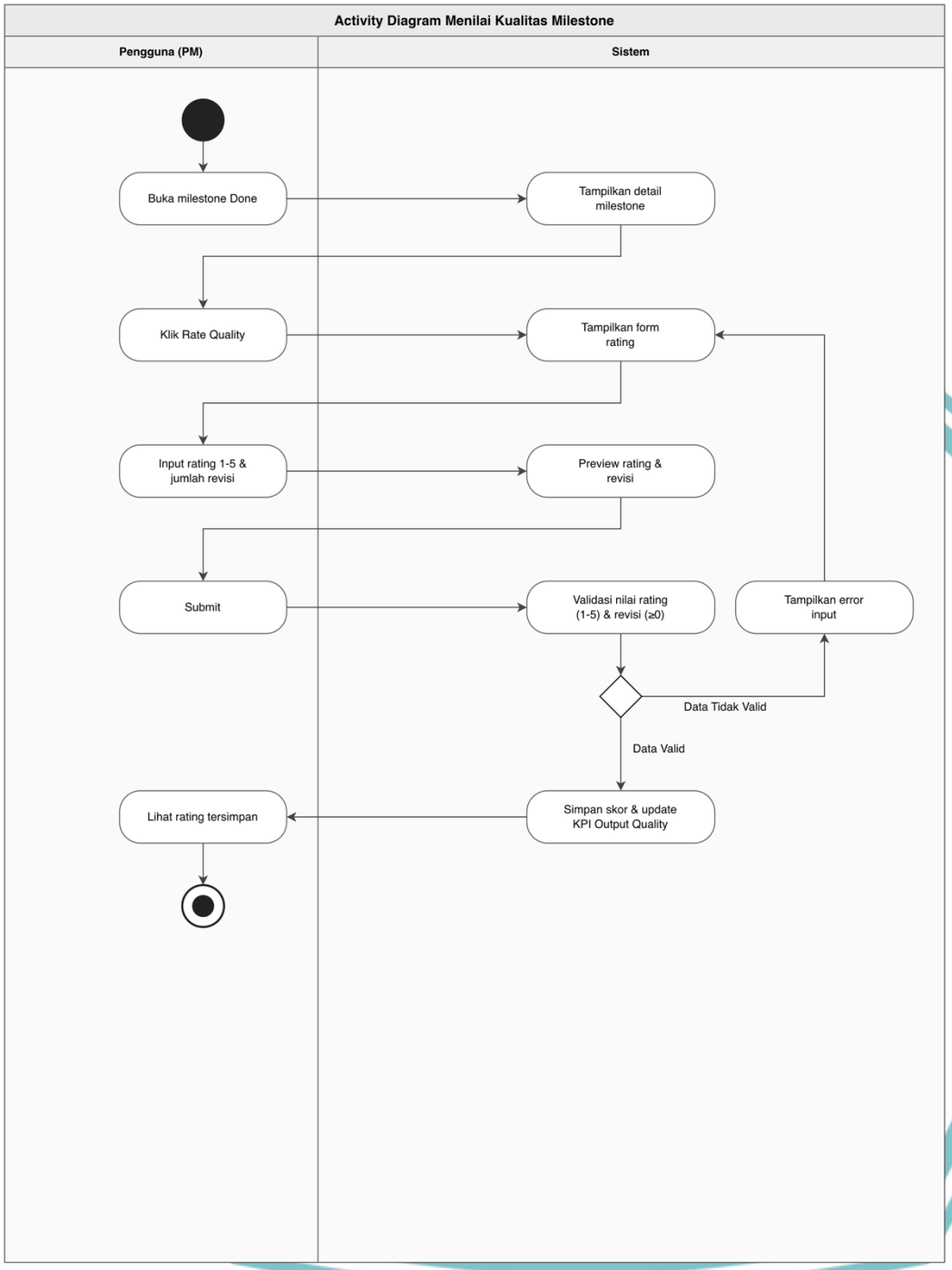



Gambar 4. 10 *Activity Diagram* Memperbarui Status *Milestone*

Sumber: Olahan Peneliti

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



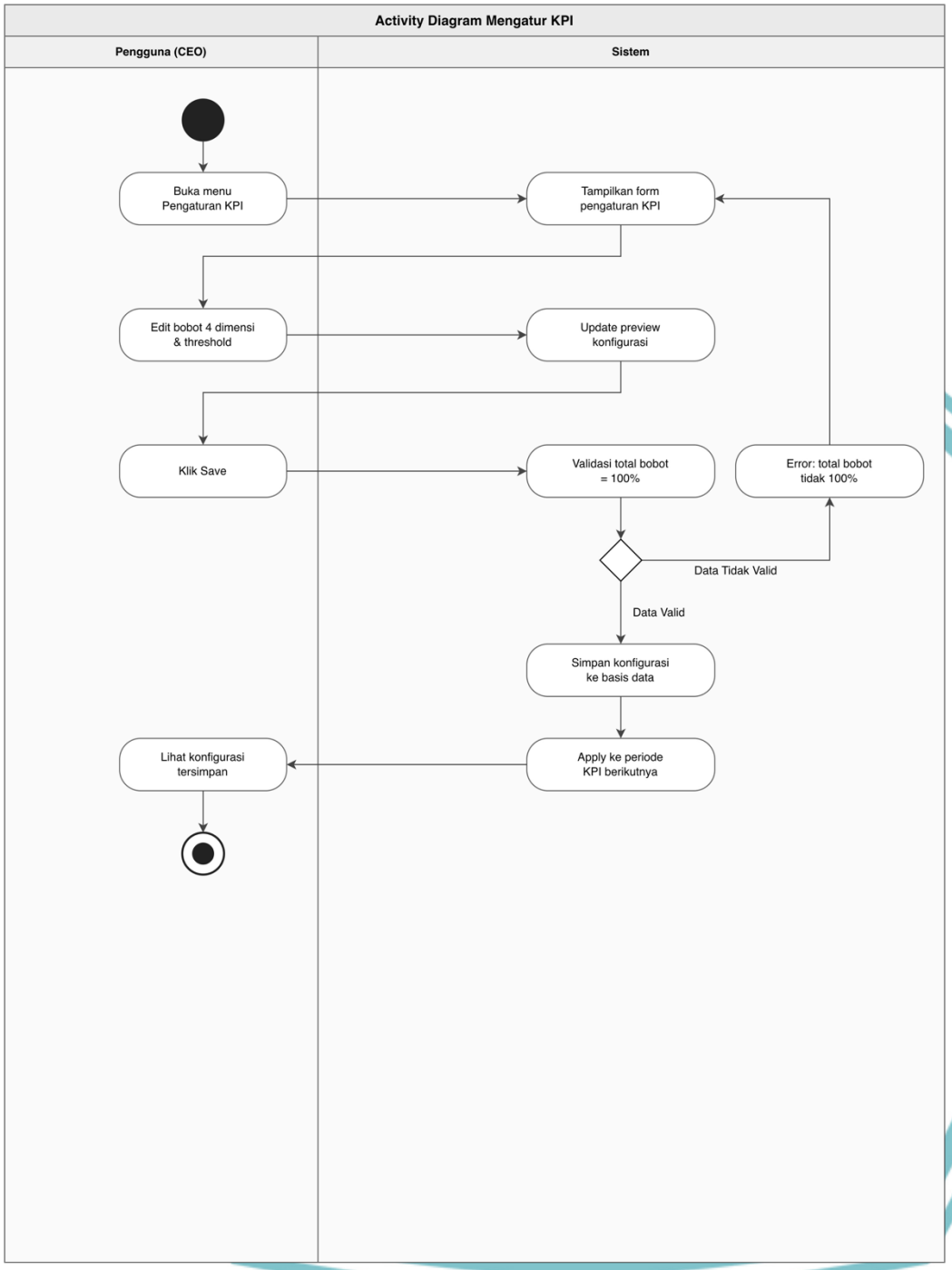

Gambar 4. 11 *Activity Diagram* Menilai Kualitas *Milestone*

Sumber: Olahan Peneliti



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



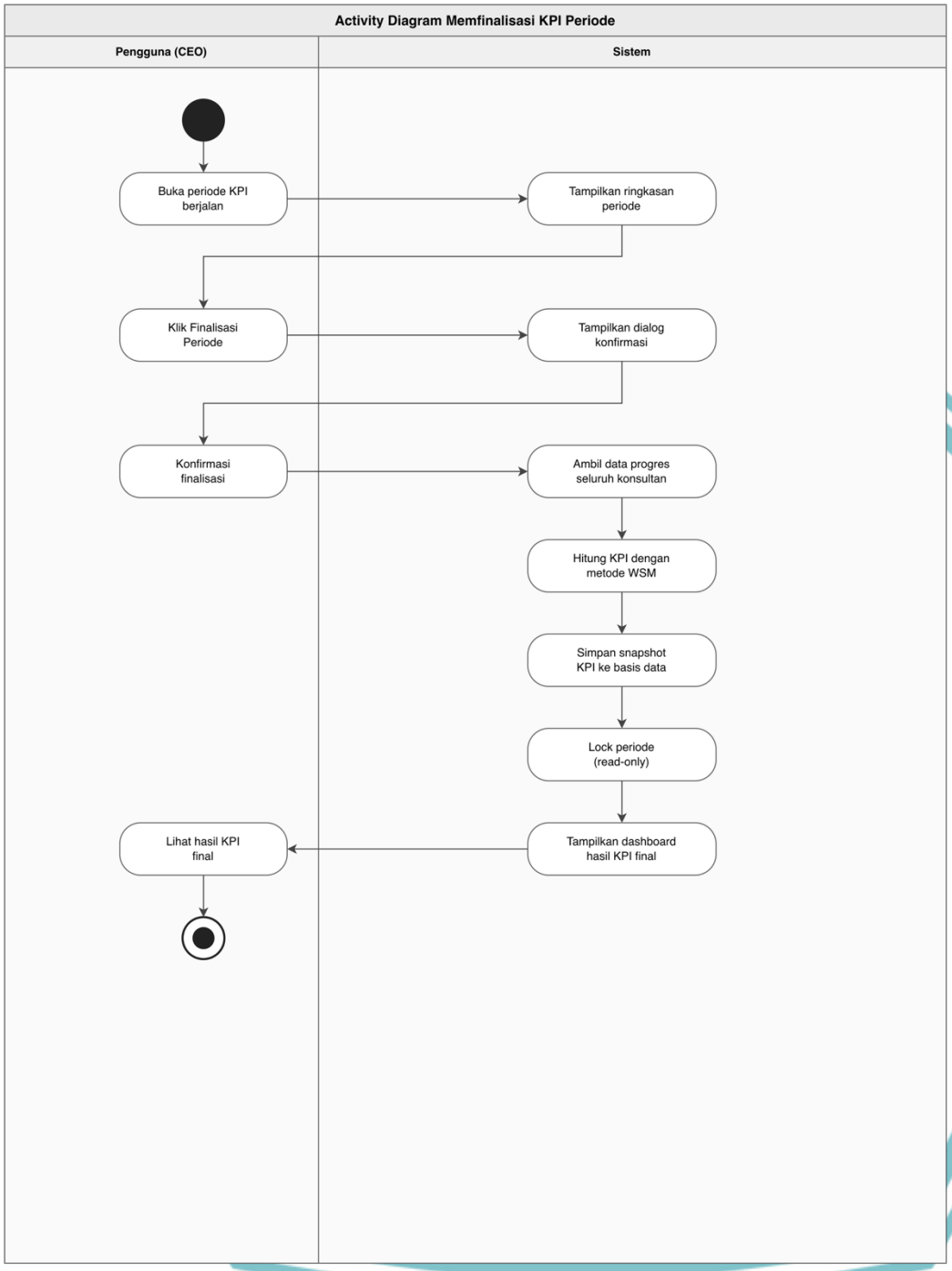

Gambar 4. 12 *Activity Diagram* Mengatur KPI

Sumber: Olahan Peneliti



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



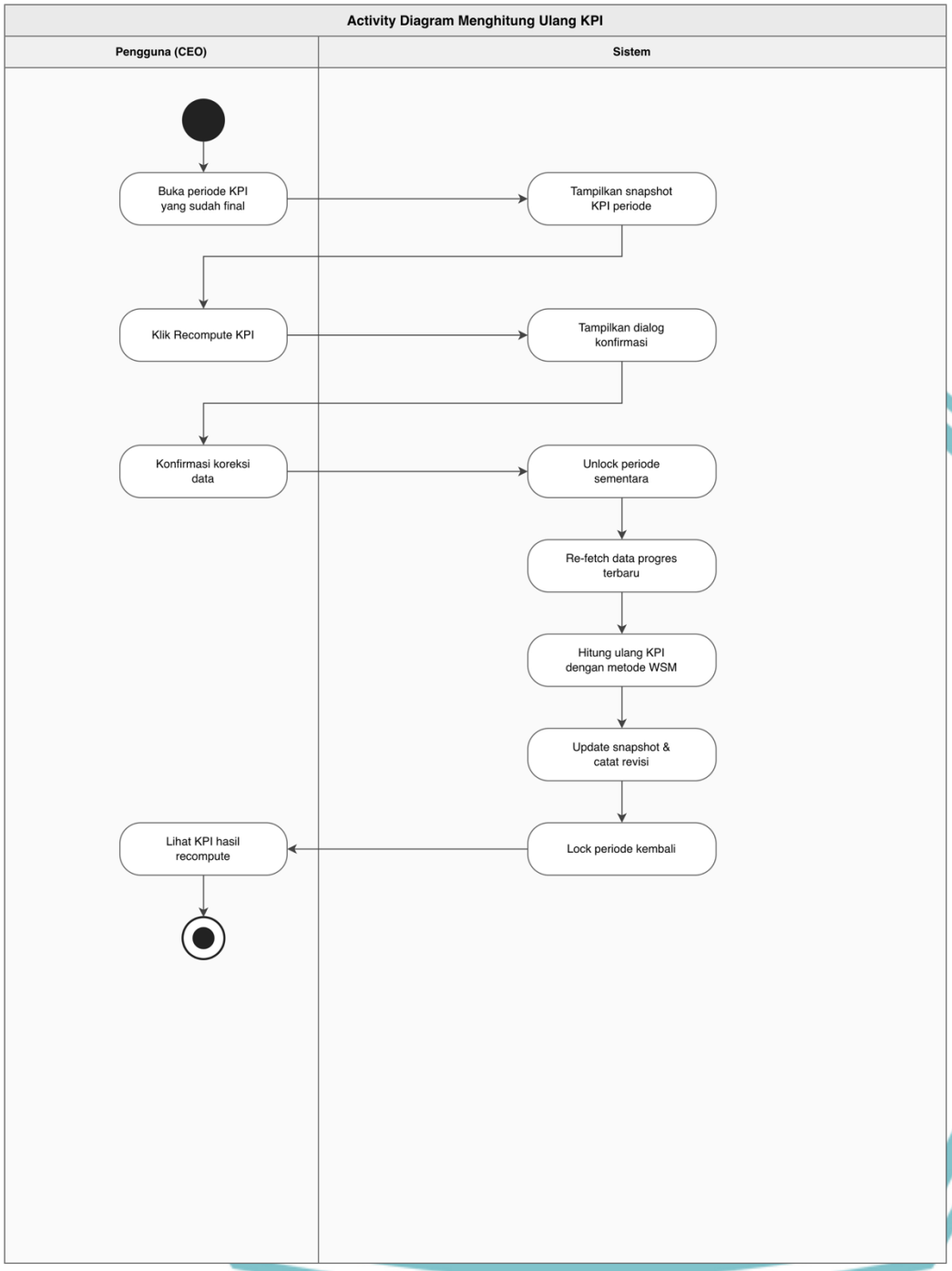

Gambar 4. 13 *Activity Diagram* Memfinalisasi KPI Periode

Sumber: Olahan Peneliti



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



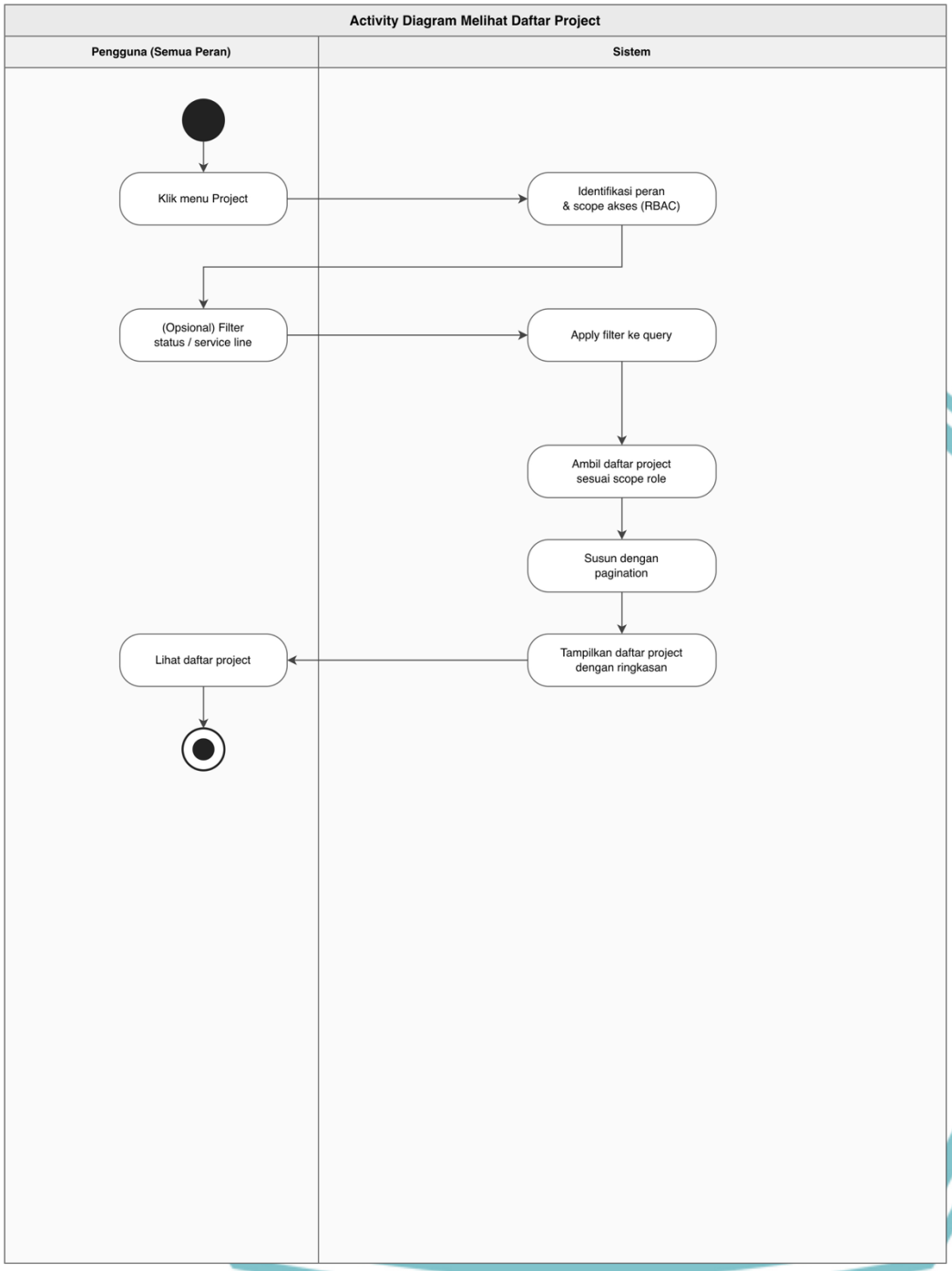

Gambar 4. 14 *Activity Diagram* Menghitung Ulang KPI

Sumber: Olahan Peneliti



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

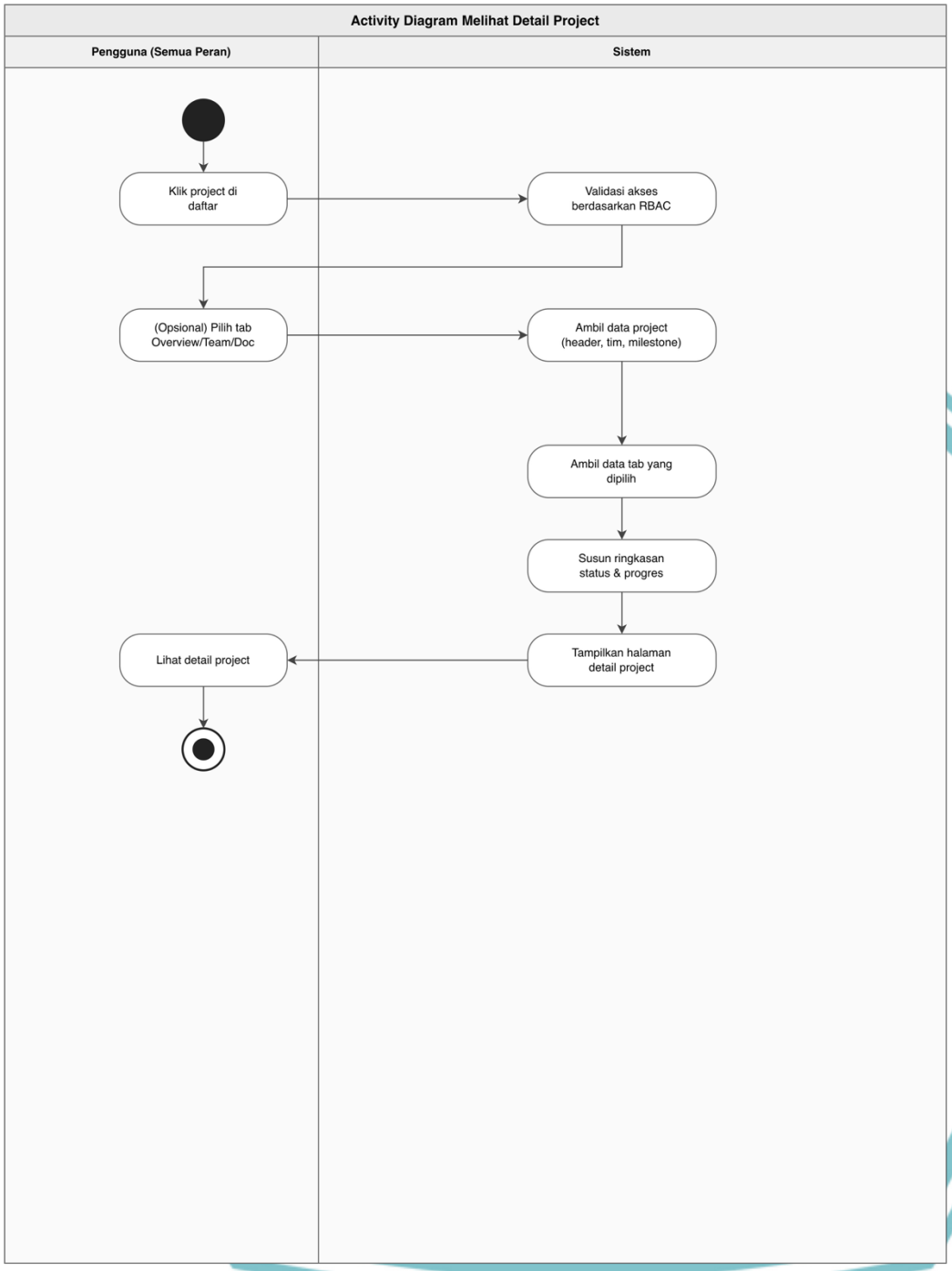



Gambar 4. 15 *Activity Diagram* Melihat Daftar Project

Sumber: Olahan Peneliti

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



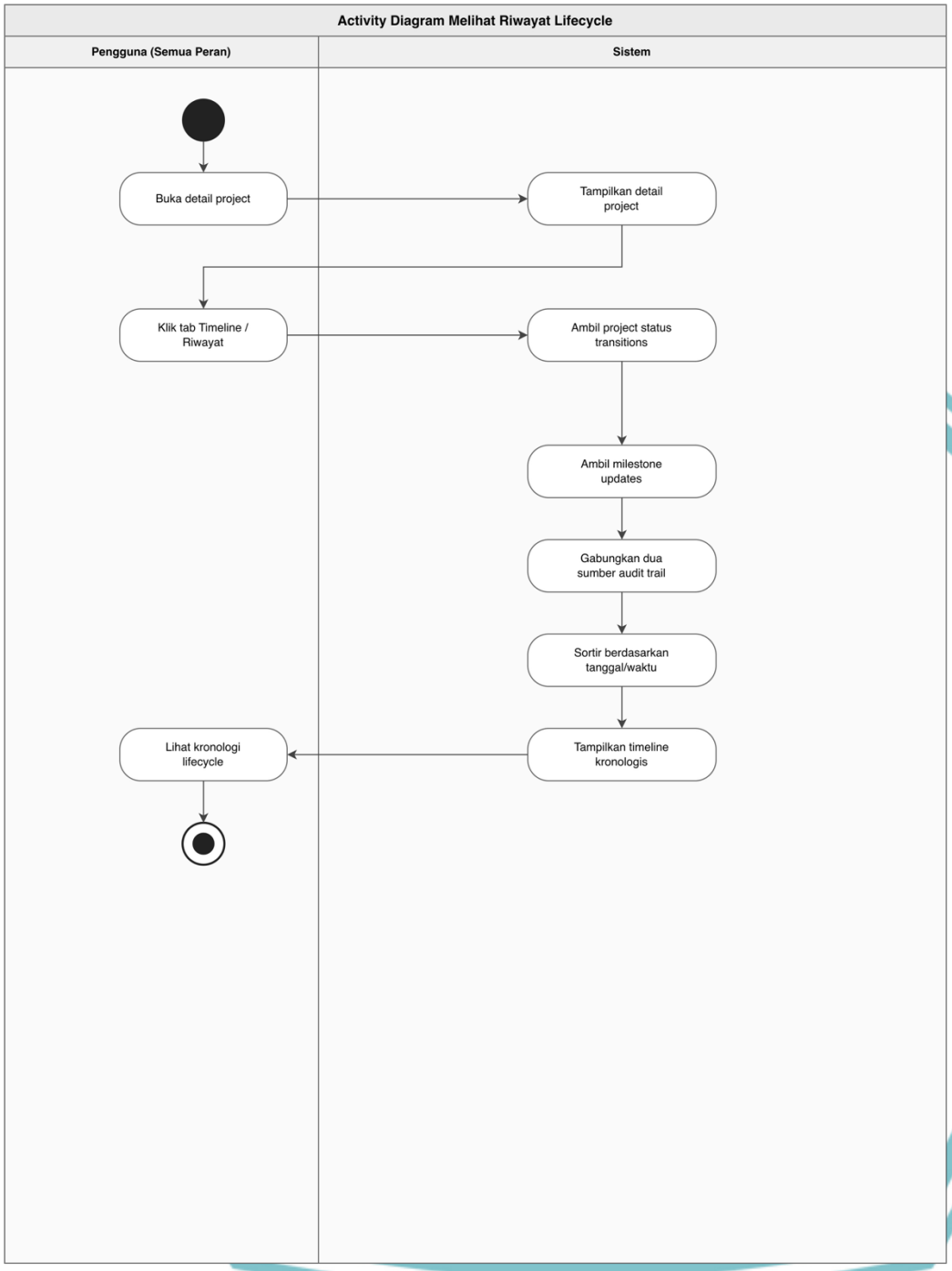

Gambar 4. 16 *Activity Diagram* Melihat Detail Project

Sumber: Olahan Peneliti



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



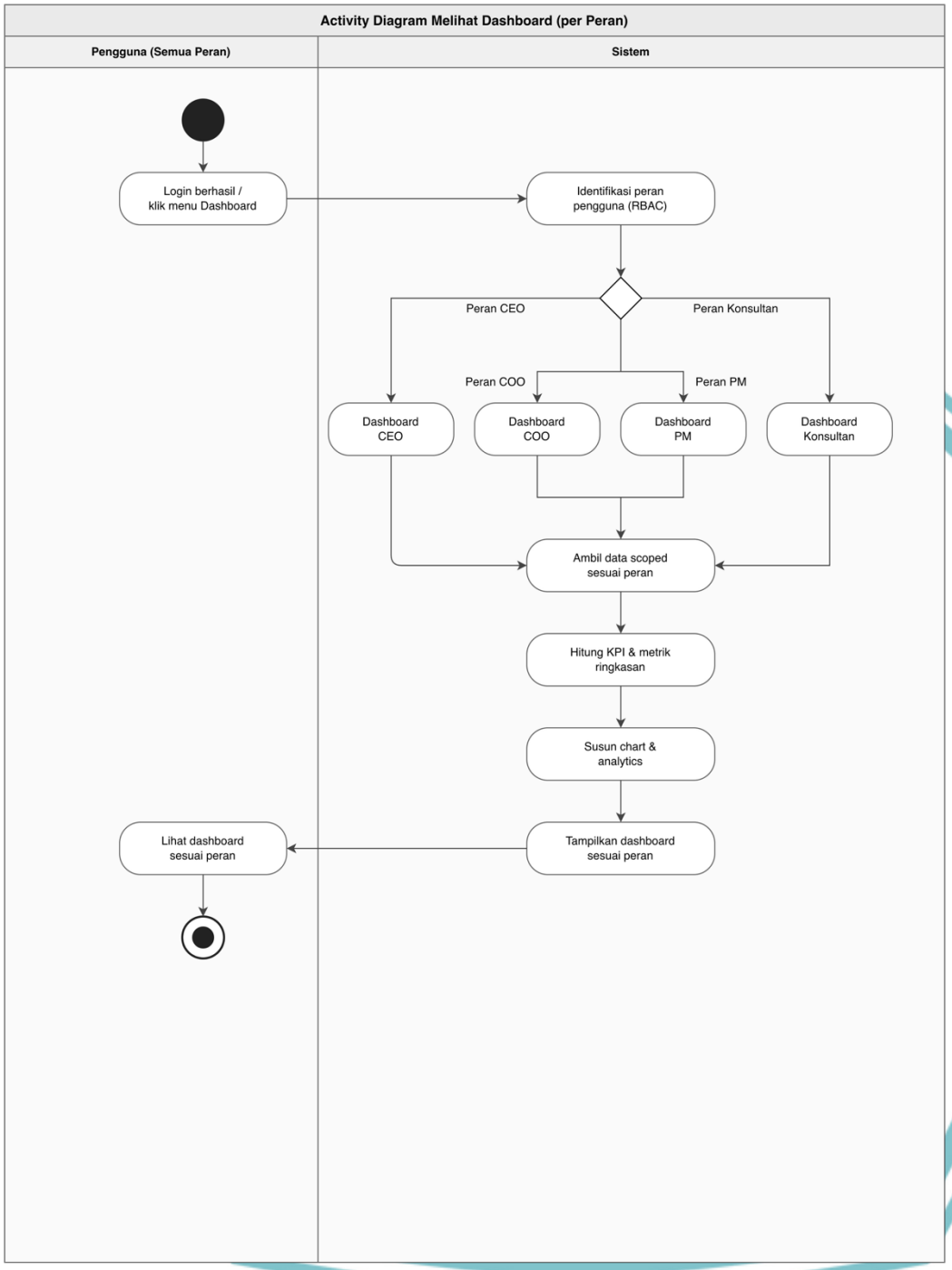

Gambar 4. 17 *Activity Diagram* Melihat Riwayat *Lifecycle*

Sumber: Olahan Peneliti



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



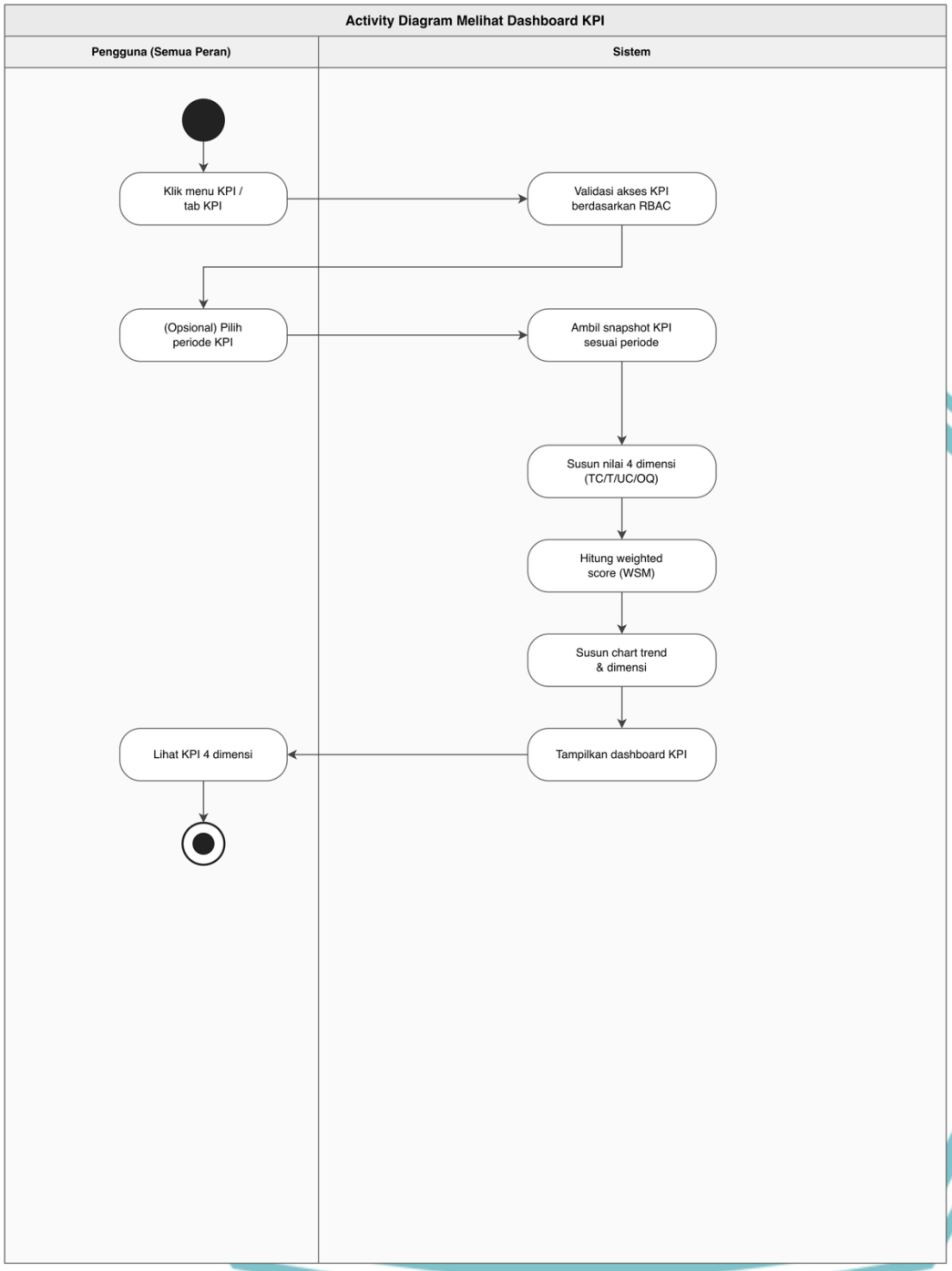

Gambar 4. 18 *Activity Diagram* Melihat *Dashboard* (per Peran)

Sumber: Olahan Peneliti



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



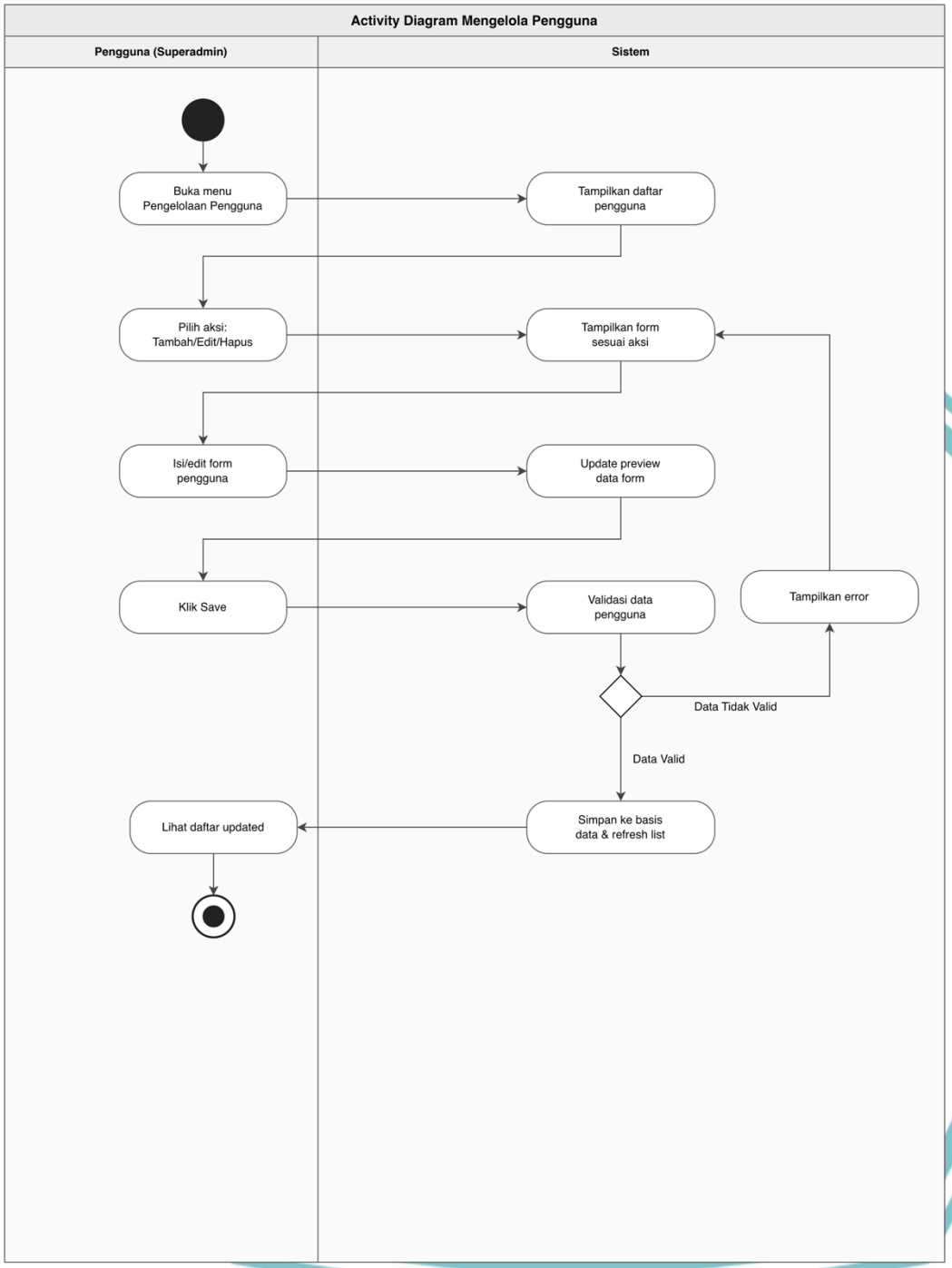

Gambar 4. 19 *Activity Diagram* Melihat *Dashboard* KPI

Sumber: Olahan Peneliti



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

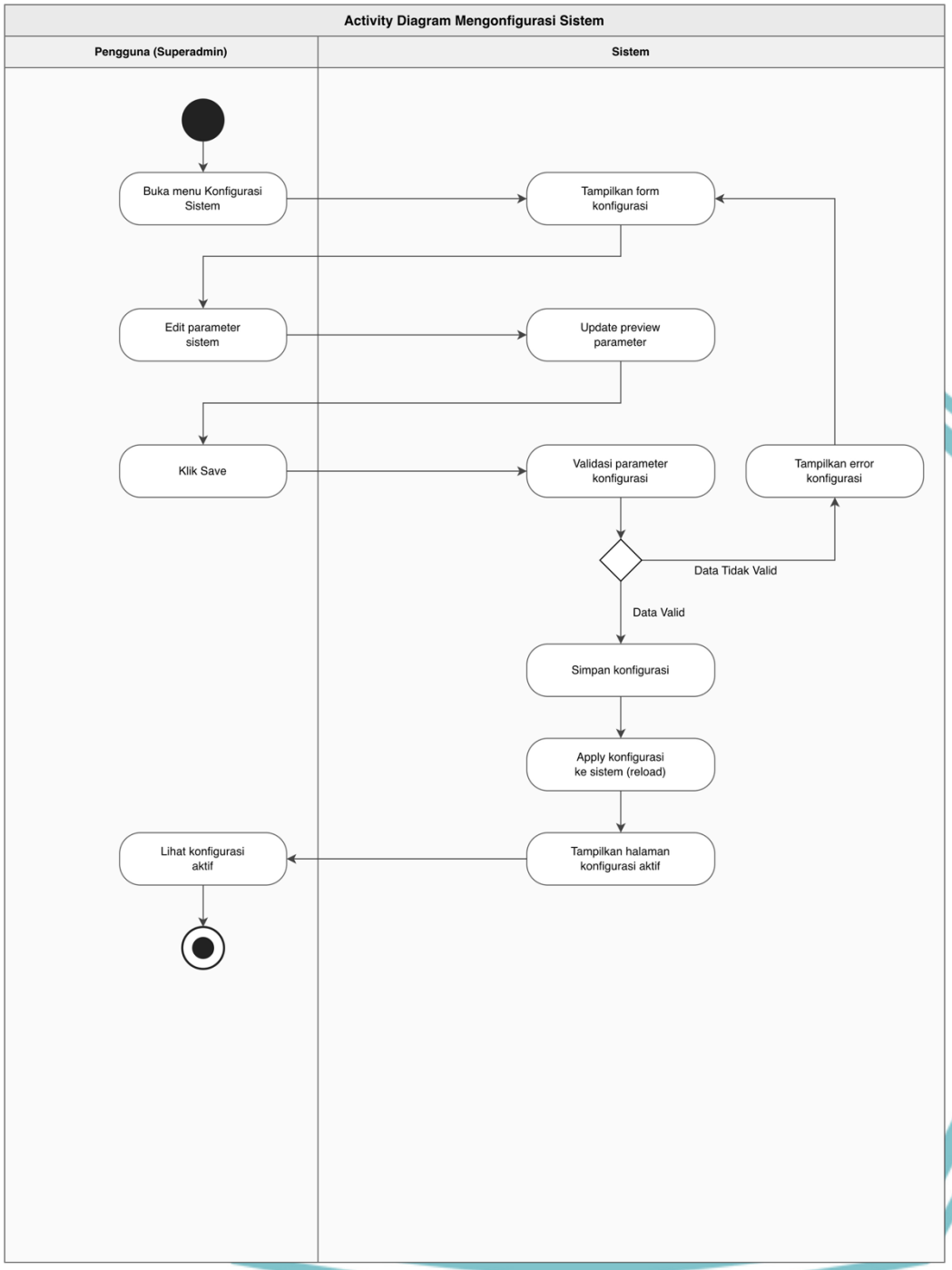



Gambar 4. 20 *Activity Diagram* Mengelola Pengguna

Sumber: Olahan Peneliti

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta




Gambar 4. 21 *Activity Diagram* Mengonfigurasi Sistem

Sumber: Olahan Peneliti

Pada Gambar 4.4 sampai dengan Gambar 4.21 ditampilkan delapan belas *Activity Diagram* yang merepresentasikan alur eksekusi seluruh *use case* pada sistem *monitoring* proyek dan penilaian KPI. Setiap diagram disusun dengan partisi



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

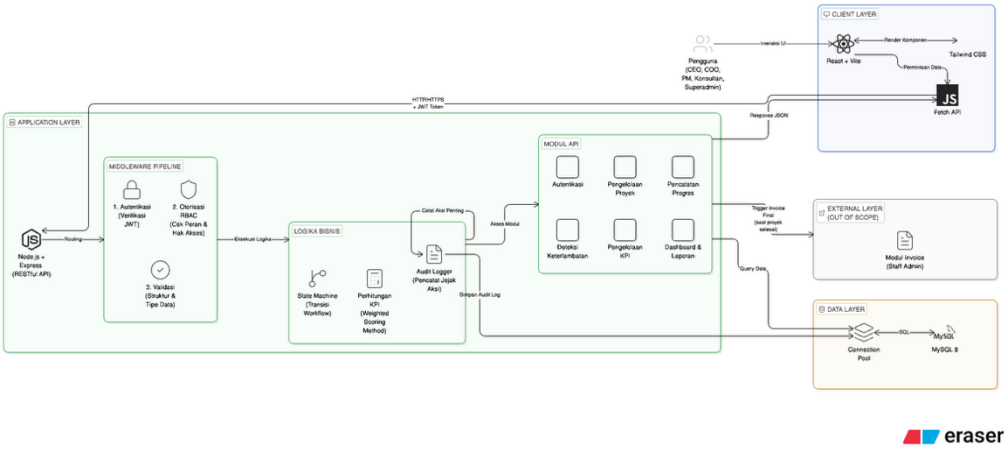
*swimlane* yang memisahkan tanggung jawab pengguna dan sistem, sehingga setiap aksi pengguna dan *respon* otomatis sistem dapat ditelusuri secara jelas. Visualisasi ini melengkapi *Use Case Diagram* sebelumnya dengan menjawab pertanyaan bagaimana setiap fitur sistem dieksekusi, mencakup percabangan keputusan, integrasi dengan *sub-use-case* melalui relasi *include*, serta penanganan kasus alternatif. Hasil pemodelan pada *Activity Diagram* menjadi dasar perancangan logika modul dan basis data pada tahap perancangan sistem.

## 4.2 Perancangan Sistem

Perancangan sistem merupakan tahap yang menerjemahkan kebutuhan fungsional dan non-fungsional yang telah dirumuskan ke dalam desain teknis yang siap diimplementasikan. Perancangan dilakukan secara bertahap dari level arsitektur hingga level antarmuka pengguna, dengan prinsip pemisahan tanggung jawab antar lapisan dan keselarasan dengan praktik pengembangan perangkat lunak modern. Lima aspek perancangan yang dibahas mencakup arsitektur sistem yang menetapkan struktur komunikasi antar komponen, perancangan basis data yang menjabarkan struktur entitas dan relasinya, perancangan *workflow* dan *state machine* yang mengatur transisi status pekerjaan, perancangan antarmuka API yang menghubungkan klien dengan *server*, serta perancangan antarmuka pengguna yang memvisualisasikan fungsionalitas sistem kepada pengguna akhir.

### 4.2.1 Perancangan Arsitektur Sistem

Perancangan arsitektur sistem bertujuan menetapkan kerangka struktural yang akan menjadi acuan dalam tahap implementasi. Arsitektur dirancang untuk mengakomodasi kebutuhan fungsional dan non-fungsional yang telah dirumuskan, dengan penekanan pada modularitas, skalabilitas, dan keamanan akses berbasis peran. Sistem dirancang menggunakan pola arsitektur *client-server multi-tier* yang memisahkan logika presentasi, logika bisnis, dan kekonsistenan data ke dalam lapisan yang independen, sehingga setiap lapisan dapat dikembangkan, diuji, dan dipelihara secara terpisah tanpa mengganggu lapisan lainnya.




Gambar 4. 22 Arsitektur Sistem *Monitoring* Proyek dan Penilaian KPI

Sumber: Olahan Peneliti

Pada Gambar 4.22 ditampilkan arsitektur sistem yang terdiri atas empat lapisan utama yang saling berkomunikasi melalui protokol HTTP/HTTPS. Lapisan pertama adalah *Client Layer* yang berjalan di peramban pengguna dan bertanggung jawab atas seluruh logika presentasi dan interaksi antarmuka. Lapisan kedua adalah *Application Layer* yang berjalan di sisi *server* dan menangani seluruh logika bisnis, validasi, otorisasi, serta orkestrasi alur kerja sistem. Lapisan ketiga adalah Data Layer yang menyimpan seluruh data operasional sistem dalam basis data relasional. Lapisan keempat adalah *External Layer* yang merepresentasikan layanan pihak ketiga yang berinteraksi dengan sistem, dalam penelitian ini berupa modul penagihan *invoice* yang berada di luar ruang lingkup pengembangan.

*Client Layer* dibangun menggunakan *framework* React.js versi 19 yang dipasangkan dengan *tooling Vite* untuk mendukung kecepatan proses pengembangan dan TypeScript untuk menjamin keamanan tipe data sejak tahap penulisan kode. Antarmuka pengguna disusun dari komponen-komponen modular yang dapat digunakan kembali di berbagai halaman, dengan penataan tampilan menggunakan *utility-first framework* Tailwind *CSS*. Lapisan ini hanya bertanggung jawab atas penyajian data dan penangkapan interaksi pengguna, sedangkan seluruh keputusan otorisasi dan validasi bisnis dialihkan ke *Application Layer* melalui pemanggilan API. Komunikasi dengan *Application Layer* dilakukan menggunakan



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Fetch API sebagai HTTP *client native browser* yang dibungkus oleh modul kustom *api-client* untuk menyediakan *helper apiGet, apiPost, apiPut*, dan *apiDelete* dengan *auto-injection* JSON *Web Token* dari *localStorage* secara terpusat.

*Application Layer* dibangun menggunakan *runtime* Node.js dengan *framework* Express.js versi 5 yang menyediakan struktur *middleware* yang fleksibel untuk menangani permintaan HTTP. Lapisan ini mengekspos sekumpulan *endpoint* RESTful API yang dikelompokkan berdasarkan modul fungsionalitas, yaitu modul autentikasi, pengelolaan proyek, pencatatan progres, deteksi keterlambatan, pengelolaan KPI, serta *dashboard* dan laporan. Setiap permintaan yang masuk akan melewati tiga lapisan *middleware* secara berurutan, yaitu *middleware* autentikasi yang memverifikasi keabsahan JSON *Web Token*, *middleware* otorisasi berbasis peran yang membatasi akses sesuai kewenangan pengguna, dan *middleware* validasi yang memastikan struktur dan tipe data permintaan sesuai dengan kontrak API. Logika bisnis seperti perhitungan KPI menggunakan metode WSM dan transisi *state workflow* dieksekusi sepenuhnya pada lapisan ini untuk memastikan integritas data tidak dapat dimanipulasi dari sisi klien.

Data Layer menggunakan sistem manajemen basis data relasional MySQL versi 8 yang menyimpan seluruh entitas operasional sistem, mencakup data pengguna, proyek, *milestone*, transisi status *workflow*, parameter KPI, hasil perhitungan KPI, serta jejak audit perubahan data penting. Struktur basis data dirancang dalam bentuk skema relasional yang dinormalisasi hingga bentuk normal ketiga untuk menghindari pemborosan dan menjaga konsistensi data. Akses ke basis data hanya dapat dilakukan oleh *Application Layer* melalui koneksi *pool* yang dikelola secara terpusat, sehingga klien tidak memiliki akses langsung ke basis data dan seluruh operasi data dapat diaudit pada satu titik.

Alur komunikasi antar lapisan mengikuti pola *request-response* yang sinkron. Permintaan diawali dari interaksi pengguna pada *Client Layer* yang dikirim ke *Application Layer* melalui *endpoint* API tertentu beserta *payload* dan *token* autentikasi. *Application Layer* memvalidasi permintaan melalui rangkaian *middleware*, menjalankan logika bisnis yang sesuai, melakukan operasi baca atau tulis ke Data Layer, kemudian mengembalikan hasil dalam format JSON kepada

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

*Client Layer*. Untuk operasi yang berdampak pada data utama seperti perubahan status proyek dan finalisasi KPI, sistem mencatat aktivitas ke dalam tabel audit log secara otomatis sehingga setiap perubahan dapat ditelusuri kembali beserta pelaku dan waktu kejadiannya.

Pemilihan pola arsitektur *client-server multi-tier* didasarkan pada beberapa pertimbangan teknis. Pertama, pemisahan logika presentasi dari logika bisnis memungkinkan tim pengembang melakukan perubahan tampilan tanpa berisiko mempengaruhi integritas data. Kedua, sentralisasi logika bisnis di *Application Layer* memastikan aturan bisnis seperti perhitungan KPI dan validasi transisi *workflow* konsisten untuk seluruh pengguna, terlepas dari peramban atau perangkat yang digunakan. Ketiga, arsitektur ini selaras dengan skala operasional perusahaan yang berukuran menengah dengan jumlah pengguna aktif terbatas, sehingga pola monolitik yang lebih sederhana lebih sesuai dibandingkan pola *microservices* yang membutuhkan kompleksitas infrastruktur tinggi tanpa keuntungan proporsional pada skala saat ini.

### 4.2.2 Perancangan Hak Akses Pengguna

Perancangan hak akses pengguna bertujuan mendefinisikan secara eksplisit kewenangan yang dimiliki setiap peran dalam sistem sehingga implementasi otorisasi dapat dilakukan secara konsisten dan terverifikasi. Perancangan ini mengacu pada hasil analisis pengguna yang telah dilakukan, di mana lima peran telah diidentifikasi beserta kebutuhan informasi dan kewenangannya masing-masing. Mekanisme kontrol akses yang diterapkan adalah *Role-Based Access Control* (RBAC) dengan pemisahan tiga konsep utama yaitu pengguna, peran, dan hak akses *granular*. Pemisahan ini memungkinkan penyesuaian hak akses pada level tindakan tanpa perlu mengubah struktur kode aplikasi ketika terjadi perubahan kebijakan akses.

Hak akses pada sistem didefinisikan dalam satuan *permission* yang merepresentasikan satu tindakan spesifik terhadap satu sumber daya tertentu. Setiap *permission* dipetakan ke satu atau lebih peran melalui tabel persimpangan *role_permissions* pada basis data, sehingga evaluasi otorisasi dilakukan berdasarkan daftar *permission* yang dimiliki pengguna dan bukan berdasarkan

nama peran secara langsung. Pendekatan ini memberikan fleksibilitas yang lebih tinggi dibandingkan pengecekan peran secara langsung karena satu *permission* dapat diberikan ke beberapa peran tanpa perubahan logika pada kode aplikasi.

Tabel 4. 8 Matriks Hak Akses Pengguna per Peran

| Kode Permission | Deskripsi Hak Akses | CEO | COO | PM | Consultant | Superadmin |
|---|---|---|---|---|---|---|
| PROJECT_VIEW | Melihat daftar dan detail proyek | ✓ | ✓ | ✓ | ✓ | ✓ |
| PROJECT_CREATE | Membuat proyek baru dari *handover* | ✗ | ✓ | ✗ | ✗ | ✗ |
| PROJECT_ASSIGN_PM | Menugaskan PM ke proyek | ✗ | ✓ | ✗ | ✗ | ✗ |
| PROJECT_ASSIGN_CONSULTANT | Menugaskan Consultant ke proyek | ✗ | ✓ | ✓ | ✗ | ✗ |
| PROJECT_*UPDATE*_STATUS | Mengubah status proyek | ✗ | ✓ | ✓ | ✗ | ✗ |
| PROJECT_COMPLETE | Menyelesaikan proyek (*Mark Project Completed*) | ✗ | ✓ | ✓ | ✗ | ✗ |
| *MILESTONE*_VIEW | Melihat daftar dan detail *milestone* | ✓ | ✓ | ✓ | ✓ | ✓ |
| *MILESTONE_UPDATE*_STATUS | Memperbarui status *milestone* | ✗ | ✓ | ✓ | ✓ | ✗ |
| *MILESTONE*_RATE | Memberikan rating kualitas *milestone* | ✗ | ✗ | ✓ | ✗ | ✗ |
| KPI_VIEW | Melihat *dashboard* dan *snapshot* KPI | ✓ | ✓ | ✓ | ✓ | ✗ |
| KPI_CONFIG | Mengonfigurasi bobot dan parameter KPI | ✓ | ✗ | ✗ | ✗ | ✗ |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| Kode Permission | Deskripsi Hak Akses | CEO | COO | PM | Consultant | Superadmin |
|---|---|---|---|---|---|---|
| KPI_EXPORT | Mengekspor laporan KPI dalam format PDF | ✓ | ✓ | ✗ | ✗ | ✗ |
| *DASHBOARD*_CEO_VIEW | Mengakses *dashboard* ringkasan organisasi CEO | ✓ | ✗ | ✗ | ✗ | ✗ |
| *USER*_MANAGE | Mengelola akun pengguna (CRUD) | ✗ | ✗ | ✗ | ✗ | ✓ |
| NOTIFICATION_VIEW | Melihat notifikasi keterlambatan *milestone* | ✓ | ✓ | ✓ | ✓ | ✗ |

Sumber: Olahan Peneliti

Berdasarkan matriks pada Tabel 4.8, terdapat dua prinsip utama yang diterapkan dalam perancangan hak akses sistem. Prinsip pertama adalah pemisahan kewenangan berdasarkan peran fungsional, di mana hak akses yang bersifat strategis seperti KPI_CONFIG dan *DASHBOARD*_CEO_VIEW hanya diberikan kepada CEO, hak akses operasional seperti PROJECT_CREATE dan PROJECT_ASSIGN_PM diberikan kepada COO, serta hak akses administratif teknis seperti *USER*_MANAGE hanya diberikan kepada Superadmin. Prinsip kedua adalah pembatasan lingkup data berdasarkan kepemilikan, di mana peran PM hanya dapat mengakses data proyek yang dipimpinnya, Consultant hanya dapat mengakses *milestone* dan KPI pribadinya, dan COO memiliki akses penuh terhadap seluruh data operasional. Kombinasi kedua prinsip ini memastikan bahwa setiap pengguna hanya dapat melakukan tindakan yang sesuai dengan tanggung jawabnya dan hanya terhadap data yang menjadi kewenangannya, sehingga kerahasiaan dan integritas data operasional perusahaan terjaga secara sistematis.

### 4.2.3 Perancangan Basis Data

Perancangan basis data bertujuan menetapkan struktur penyimpanan data yang menjadi fondasi operasional sistem. Basis data dirancang menggunakan pendekatan

relasional untuk mengakomodasi seluruh entitas yang terlibat dalam proses *monitoring* proyek dan penilaian kinerja karyawan, mulai dari data identitas pengguna dan hak akses hingga data hasil perhitungan KPI dan jejak transisi status. Struktur tabel dirancang dengan normalisasi hingga bentuk normal ketiga untuk menghilangkan pemborosan data dan menjaga konsistensi nilai antar tabel. Beberapa atribut secara sengaja digabungkan pada tabel-tabel yang berfungsi sebagai jejak audit untuk menjaga integritas historis ketika data referensi berubah di masa depan.

Berdasarkan analisis kebutuhan fungsional dan *Use Case Diagram*, dirumuskan delapan belas tabel yang dikelompokkan ke dalam empat modul logis. Modul pertama adalah *Identity and Access Management* yang menangani *autentikasi* pengguna dan otorisasi berbasis peran, terdiri dari enam tabel yaitu *users*, *roles*, *permissions*, *role_permissions*, *departments*, dan *user_departments*. Modul kedua adalah *Monitoring* Proyek sebagai inti penelitian, terdiri dari lima tabel yaitu *projects*, *project_consultants*, *project_status_transitions*, *project_milestones*, dan *project_milestone_updates*. Modul ketiga adalah Konfigurasi dan Penilaian KPI, terdiri dari empat tabel yaitu *task_templates*, *task_template_tasks*, *kpi_period_config*, dan *kpi_snapshots*. Modul keempat adalah tabel integrasi yang berada di luar ruang lingkup pengembangan tetapi memiliki keterkaitan dengan sistem, yaitu *handovers*, *handover_milestones*, dan *invoice_terms*.

Tabel 4. 9 Daftar Entitas Basis Data Sistem

| No | Modul | Entitas | Deskripsi |
|---|---|---|---|
| 1 | *Identity and Access* | *users* | Identitas pengguna sistem beserta kredensial dan status aktif |
| 2 | *Identity and Access* | *roles* | Daftar peran sistem (CEO, COO, PM, Consultant, Superadmin) |



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| No | Modul | Entitas | Deskripsi |
|---|---|---|---|
| 3 | *Identity and Access* | *permissions* | Daftar hak akses *granular* yang tersedia di sistem |
| 4 | *Identity and Access* | *role_permissions* | Pemetaan *many-to-many* antara peran dan hak akses |
| 5 | *Identity and Access* | *departments* | Daftar divisi atau departemen organisasi |
| 6 | *Identity and Access* | *User_departments* | Pemetaan *many-to-many* pengguna dengan departemen |
| 7 | *Monitoring* Proyek | *projects* | Data proyek beserta status dan tim yang ditugaskan |
| 8 | *Monitoring* Proyek | *projects_consultants* | Pemetaan *many-to-many* proyek dengan Consultant pelaksana |
| 9 | *Monitoring* Proyek | *project_status_transitions* | Jejak transisi status proyek beserta pelaku dan alasannya |
| 10 | *Monitoring* Proyek | *project_milestones* | *Milestone* per proyek beserta status, target, dan rating kualitas |
| 11 | *Monitoring* Proyek | *project_milestone_updates* | Jejak transisi status *milestone* beserta pelaku dan *timestamp* |

| No | Modul | Entitas | Deskripsi |
|---|---|---|---|
| 12 | Konfigurasi dan KPI | *task_templates* | *Template tasks* standar per *service line* untuk akselerasi setup proyek |
| 13 | Konfigurasi dan KPI | *task_template_tasks* | Detail *task* per *template* beserta urutan dan bobot |
| 14 | Konfigurasi dan KPI | *kpi_period_config* | Konfigurasi bobot KPI, toleransi, dan skala penilaian per periode |
| 15 | Konfigurasi dan KPI | *kpi_snapshots* | Hasil perhitungan KPI per Consultant per periode beserta status finalisasi |
| 16 | Integrasi (Eksternal) | *handovers* | Data *handover* proyek dari modul Business Development |
| 17 | Integrasi (Eksternal) | *handover_milestones* | *Milestone* awal dari *handover* yang menjadi referensi proyek |
| 18 | Integrasi (Eksternal) | *invoice_terms* | Termin *invoice* yang dipicu oleh peristiwa proyek |

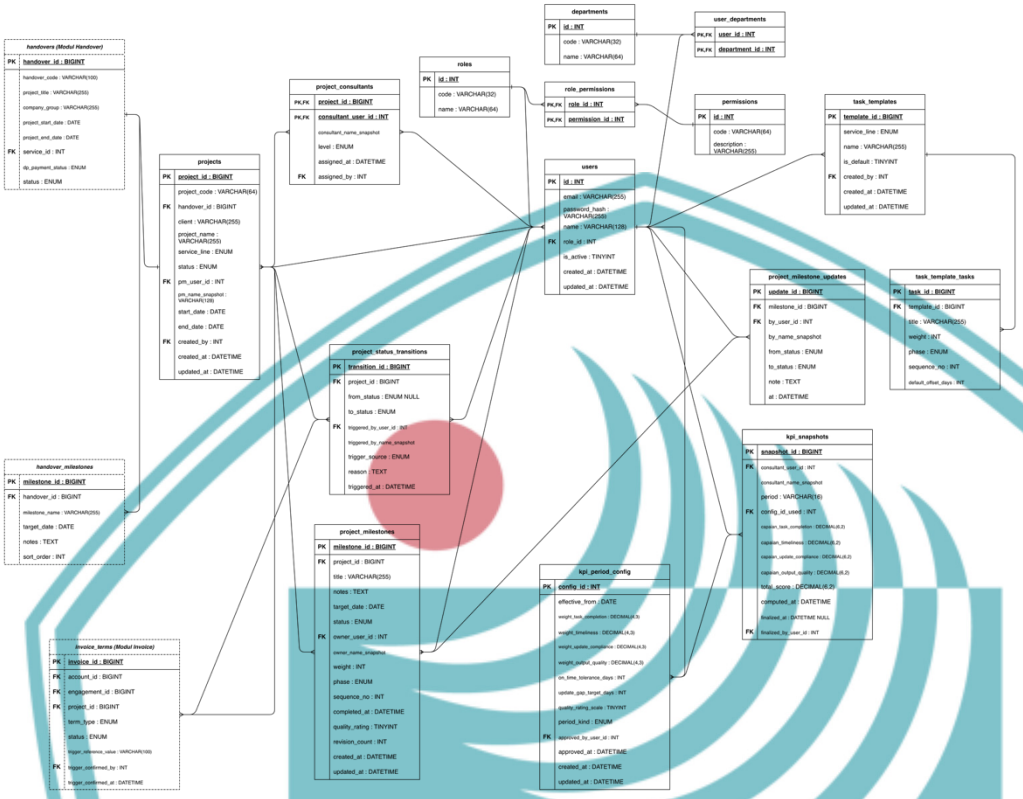
Sumber: Olahan Peneliti

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta




Gambar 4. 23 Tabel Relasi Database Sistem

Sumber: Olahan Peneliti

Pada Gambar 4.23 menyajikan diagram relasi antar tabel sistem dalam bentuk tabel relasional yang memvisualisasikan seluruh struktur kolom beserta *primary key*, *foreign key*, dan relasi antar tabel. Tabel-tabel yang dikembangkan dalam ruang lingkup penelitian digambarkan dengan garis tegas, sedangkan tabel integrasi yang berada di luar ruang lingkup digambarkan dengan garis putus-putus. Pemisahan ini memberikan kejelasan visual mengenai batas tanggung jawab pengembangan, sekaligus menunjukkan titik integrasi sistem dengan modul lain dalam ekosistem perusahaan.

Modul *Identity and Access Management* mengimplementasikan pola RBAC dengan pemisahan tiga konsep utama yaitu pengguna, peran, dan hak akses. Tabel *users* menyimpan identitas dan kredensial pengguna dengan kolom *role_id* sebagai kunci asing ke tabel *roles*, yang berarti setiap pengguna memiliki tepat satu peran pada satu waktu. Tabel *permissions* menyimpan daftar hak akses *granular* sebagai entitas

tersendiri, dan dipetakan ke peran melalui tabel persimpangan *role_permissions* yang mengimplementasikan relasi *many-to-many* antara *roles* dan *permissions*. Pendekatan ini memberikan fleksibilitas penyesuaian hak akses tanpa perlu mengubah struktur kode aplikasi ketika ada perubahan kebijakan akses. Tabel *departments* dan *user_departments* melengkapi modul ini dengan informasi struktur organisasi, di mana tabel *user_departments* mengimplementasikan relasi *many-to-many* antara *users* dan *departments* dengan asumsi seorang pengguna dapat terdaftar pada lebih dari satu departemen sesuai konteks penugasannya.

Modul *Monitoring* Proyek menjadi inti dari basis data penelitian. Tabel *projects* menyimpan informasi proyek beserta status *workflow*, referensi ke *handover_id* sebagai sumber awal proyek, dan kolom *pm_user_id* sebagai kunci asing ke *users* untuk penugasan PM. Kolom *pm_name_snapshot* disimpan berdampingan dengan *pm_user_id* sebagai rekaman nama PM pada saat penugasan dilakukan untuk keperluan integritas historis. Penugasan Consultant ke proyek dimodelkan melalui tabel *project_consultants* sebagai relasi *many-to-many* antara *projects* dan *users*, karena satu proyek dapat melibatkan beberapa Consultant dan seorang Consultant dapat terlibat di beberapa proyek secara bersamaan. Setiap perubahan status proyek dicatat dalam tabel *project_status_transitions* dengan kolom *from_status* yang dapat bernilai *NULL* untuk merepresentasikan kejadian inisiasi proyek, serta kolom *trigger*_source yang membedakan apakah transisi dipicu oleh pengguna atau oleh sistem. Pola yang serupa diterapkan pada level *milestone* melalui tabel *project_milestones* yang menyimpan kolom *quality_rating* sebagai masukan PM saat validasi *milestone*, dan tabel *project_milestone_updates* yang merekam jejak setiap perubahan status *milestone* beserta pelaku dan catatan perubahan. Pemisahan jejak transisi ke tabel terpisah memungkinkan rekonstruksi penuh histori proyek dan menjadi dasar perhitungan dimensi KPI *Update Compliance* dan *Timeliness*.

Modul Konfigurasi dan Penilaian KPI memisahkan parameter konfigurasi dari data hasil perhitungan untuk memastikan *auditability* skor KPI. Tabel *task_templates* dan *task_template_tasks* menyimpan *template task* standar per *service line* yang berfungsi sebagai acuan saat pembuatan proyek baru melalui kolom *service_line*, *phase*, *sequence_no*, dan *default_offset_days* pada *task_template_tasks*. Tabel

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

*kpi_period_config* menyimpan seluruh parameter perhitungan KPI mencakup bobot empat dimensi dalam tipe DECIMAL(4,3), toleransi ketepatan waktu (*on_time_tolerance_days*), target frekuensi pembaruan (*update_gap_target_days*), skala *rating* kualitas (*quality_rating_scale*), jenis periode (*period_kind*), serta kolom *approved_by_user_id* dan *approved_at* untuk mencatat siapa yang menyetujui konfigurasi. Setiap perubahan konfigurasi oleh CEO menghasilkan baris baru dengan kolom *effective_from*, sehingga konfigurasi historis tetap tersimpan untuk keperluan audit. Tabel *kpi_snapshots* menyimpan hasil perhitungan KPI per Consultant per periode dengan empat kolom capaian dimensi bertipe DECIMAL(6,2), kolom *total_score* sebagai hasil akhir *Weighted Scoring Method*, kolom *config_id_used* sebagai kunci asing ke *kpi_period_config* yang digunakan saat perhitungan dilakukan, serta kolom *finalized_at* yang bernilai NULL untuk *snapshot* berstatus *preliminary* dan bernilai *timestamp* untuk *snapshot* yang sudah difinalisasi oleh CEO.

Tabel *project_consultants*, *project_milestones*, *project_status_transitions*, *project_milestone_updates*, dan *kpi_snapshots* menyimpan atribut bersufiks _name_*snapshot* yang merekam nama pengguna pada saat aksi dilakukan. Pola denormalisasi yang sengaja diterapkan ini bertujuan menjaga integritas histori ketika data referensi pada tabel users berubah, seperti perubahan nama atau penghapusan akun pengguna. Tanpa pola ini, riwayat historis akan kehilangan konteks pelaku setelah data referensi berubah, yang dapat mengaburkan jejak audit. Pendekatan ini merupakan trade-off yang disadari antara prinsip normalisasi murni dan kebutuhan audit yang menjadi salah satu kebutuhan nonfungsional sistem..

Tiga tabel yaitu *handovers*, *handover_milestones*, dan *invoice_terms* ditampilkan pada Gambar 4.23 dengan garis putus-putus sebagai penanda bahwa tabel tersebut berada di luar ruang lingkup pengembangan penelitian ini. Tabel *handovers* memiliki kolom *dp_payment_status* yang menjadi kondisi prasyarat yang diperiksa sistem sebelum mengizinkan penugasan Consultant, sedangkan *handover_milestones* menyimpan *milestone* awal yang menjadi referensi pembuatan *milestone* proyek. Tabel *invoice_terms* memiliki kunci asing *project_id* ke tabel *projects* dan relasi ke *project_status_transitions* yang menunjukkan bahwa

pemicu *invoice* dikaitkan dengan kejadian transisi proyek tertentu. Sistem yang dikembangkan hanya membaca data dari *handovers* sebagai masukan pembuatan proyek dan menulis pemicu ke *invoice_terms* ketika proyek selesai, tanpa terlibat dalam logika bisnis internal kedua modul tersebut.

Pemilihan MySQL versi 8 sebagai sistem manajemen basis data didasarkan pada empat pertimbangan utama. Pertama, karakteristik data sistem bersifat sangat terstruktur dengan relasi yang kompleks antar tabel, sehingga lebih sesuai dengan paradigma relasional dibandingkan paradigma dokumen atau *key-value*. Kedua, dukungan terhadap transaksi ACID yang menjadi krusial untuk operasi seperti transisi status proyek dan finalisasi KPI yang harus dijamin konsistensinya. Ketiga, dukungan terhadap *constraint* integritas referensial melalui *foreign key* yang mempermudah penegakan aturan bisnis pada level basis data. Keempat, MySQL merupakan basis data yang matang dengan ekosistem pengembangan yang luas dan tersedia secara gratis, sehingga sesuai dengan skala operasional perusahaan saat ini.

### 4.2.4 Perancangan *Workflow*

Perancangan *workflow* dan *state machine* bertujuan menetapkan aturan transisi status yang akan dijalankan secara konsisten oleh sistem untuk seluruh proses operasional proyek. *State machine* dirancang dalam dua level yaitu level proyek yang mengatur *lifecycle* keseluruhan proyek dan level *milestone* yang mengatur status pekerjaan per tahapan. Kedua level dirancang sebagai *state machine* yang independen secara teknis namun terikat secara semantik, di mana transisi *milestone* hanya diperbolehkan ketika proyek induknya masih berada pada *state* non-terminal. Penilaian kualitas hasil pekerjaan tidak dirancang sebagai *state* melainkan sebagai atribut data yang dapat diberikan kapan saja setelah *milestone* mencapai *state* selesai, sehingga pemisahan ini memberikan fleksibilitas pada Project Manager dalam memberikan penilaian tanpa mengganggu alur kerja Consultant.



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tabel 4. 10 Daftar *State* Proyek pada Sistem

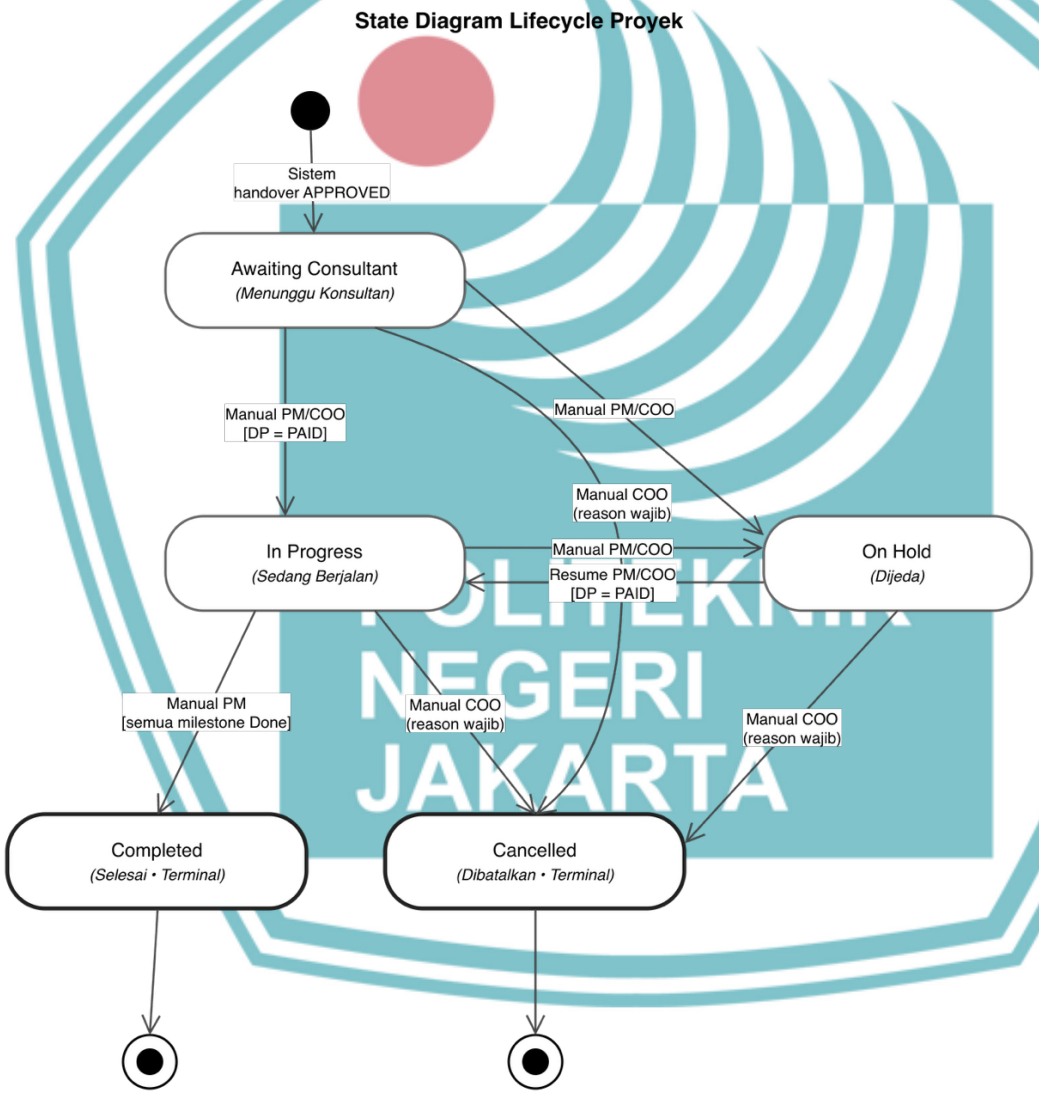
| No | Kode *State* | Nama Tampilan | Deskripsi | Tipe |
|---|---|---|---|---|
| 1 | *Awaiting Consultant* | Menunggu Consultant | Proyek baru dibuat dari *handover* oleh COO dengan Project Manager telah ditugaskan, sedang menunggu penugasan Consultant dan verifikasi pembayaran DP | *Initial* |
| 2 | *In Progress* | Sedang Berjalan | Proyek sedang dikerjakan setelah Consultant ditugaskan dan DP telah terverifikasi sebagai PAID | *Active* |
| 3 | *On Hold* | Dijeda | Proyek dijeda sementara karena alasan eksternal seperti permintaan klien atau kendala sumber daya | *Pause* |
| 4 | *Completed* | Selesai | Seluruh *milestone* telah berstatus *Done* dan invoice final telah dipicu | *Terminal* |
| 5 | *Cancelled* | Dibatalkan | Proyek dibatalkan dengan alasan tercatat, | *Terminal* |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| No | Kode *State* | Nama Tampilan | Deskripsi | Tipe |
|---|---|---|---|---|
| | | | tidak memicu pembayaran | |

Sumber: Olahan Peneliti




Gambar 4. 24 *State Diagram Lifecycle* Proyek

Sumber: Olahan Peneliti



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Pada Gambar 4.26 ditampilkan *State Diagram lifecycle* proyek yang memvisualisasikan lima *state* dan sembilan transisi yang valid pada sistem. Notasi panah menunjukkan arah transisi yang diperbolehkan, dengan label transisi yang mencantumkan pemicu transisi dan *precondition* apabila ada. *State Completed* dan *Cancelled* digambarkan sebagai *state* terminal yang tidak memiliki transisi keluar, sehingga proyek yang telah mencapai salah satu dari kedua *state* tersebut tidak dapat lagi diubah statusnya untuk menjaga integritas histori.

Tabel 4. 11 Matrix Transisi Valid pada *State Machine* Proyek

| No | Dari | Ke | Pemicu | Pelaku | *Precondition* |
|---|---|---|---|---|---|
| 1 | *(initial)* | *Awaiting Consultant* | Otomatis | Sistem | *Handover* berstatus *APPROVED* |
| 2 | *Awaiting Consultant* | *In Progress* | Manual | PM / COO / Superadmin | *handovers.dp_payment_status = PAID* |
| 3 | *Awaiting Consultant* | *On Hold* | Manual | PM / COO / Superadmin | - |
| 4 | *Awaiting Consultant* | *Cancelled* | Manual | COO / Superadmin | Alasan wajib diisi |
| 5 | *In Progress* | *On Hold* | Manual | PM / COO / Superadmin | - |
| 6 | *In Progress* | *Completed* | Manual | PM / Superadmin | Seluruh *milestone* berstatus *Done* |
| 7 | *In Progress* | *Cancelled* | Manual | COO / Superadmin | Alasan wajib diisi |
| 8 | *On Hold* | *In Progress (Resume)* | Manual | PM / COO / Superadmin | *handovers.dp_payment_status = PAID* |
| 9 | *On Hold* | *Cancelled* | Manual | COO / Superadmin | Alasan wajib diisi |

Sumber: Olahan Peneliti

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

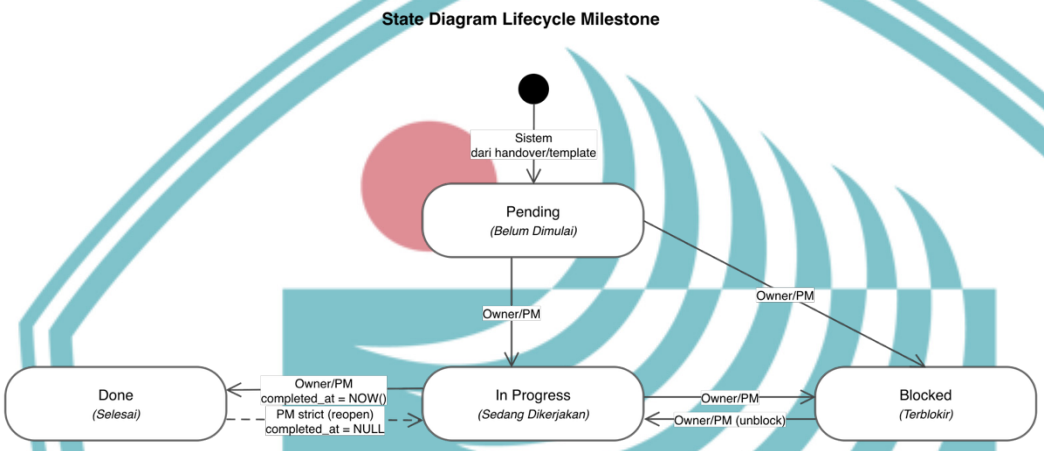
Sembilan transisi pada Tabel 4.11 dapat dikelompokkan menjadi tiga kategori berdasarkan karakteristik pemicunya. Kategori pertama adalah transisi inisialisasi yang dipicu otomatis oleh sistem ketika *handover* dari modul Business Development disetujui, menghasilkan proyek baru pada *state Awaiting Consultant* beserta *milestone* awal yang disalin dari *handover*. Kategori kedua adalah transisi operasional yang dipicu secara manual oleh Project Manager atau COO untuk mengubah status proyek selama siklus hidupnya, mencakup transisi menuju *In Progress, On Hold,* dan *Resume* kembali ke *In Progress*. Kategori ketiga adalah transisi terminal yang menyebabkan proyek mencapai *state* akhir, yaitu transisi menuju *Completed* yang hanya dapat dilakukan oleh Project Manager dengan *precondition* seluruh *milestone* telah selesai, dan transisi menuju *Cancelled* yang hanya dapat dilakukan oleh COO dengan alasan pembatalan wajib didokumentasikan untuk keperluan audit dan akuntabilitas.

Tabel 4. 12 Daftar *State Milestone* pada Sistem

| No | Kode *State* | Nama Display | Deskripsi | Tipe |
|---|---|---|---|---|
| 1 | *Pending* | Belum Dimulai | *Milestone* telah dibuat dari *handover* atau *task template*, belum disentuh Consultant | *Initial* |
| 2 | *In Progress* | Sedang Dikerjakan | Consultant atau PM menandai *milestone* sedang aktif dikerjakan | *Active* |
| 3 | *Done* | Selesai | *Milestone* telah selesai dikerjakan, kolom *completed_at* otomatis terisi waktu sistem saat ini | *Completed* |
| 4 | *Blocked* | Terblokir | *Milestone* tidak dapat dilanjutkan karena ada | *Suspended* |

| | | | | |
|---|---|---|---|---|
| | | | hambatan seperti dependensi yang belum terpenuhi | |

Sumber: Olahan Peneliti




Gambar 4. 25 *State Diagram Lifecycle Milestone*

Sumber: Olahan Peneliti

Pada Gambar 4.27 ditampilkan *State Diagram lifecycle milestone* yang terdiri atas empat *state* dengan enam transisi valid. Berbeda dengan *state machine* proyek, *milestone* tidak memiliki *state* terminal karena seorang Project Manager dapat membuka kembali *milestone* yang telah berstatus *Done* apabila ditemukan kebutuhan revisi setelah validasi. Ketika *milestone* dibuka kembali, kolom '*completed_at*' direset ke nilai NULL untuk memastikan perhitungan KPI *Timeliness* tetap akurat pada saat *milestone* diselesaikan kembali.

Tabel 4. 13 Matrix Transisi Valid pada *State Machine Milestone*

| No | Dari | Ke | Pelaku | *Side Effect* |
|---|---|---|---|---|
| 1 | *Pending* | *In Progress* | Owner / PM / Superadmin | - |



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| | | | | |
|---|---|---|---|---|
| 2 | *Pending* | *Blocked* | Owner / PM / Superadmin | - |
| 3 | *In Progress* | *Done* | Owner / PM / Superadmin | *completed_at* = waktu sistem |
| 4 | *In Progress* | *Blocked* | Owner / PM / Superadmin | - |
| 5 | *Blocked* | *In Progress* | Owner / PM / Superadmin | - |
| 6 | *Done* | *In Progress* | PM / Superadmin | *completed_at* = waktu sistem |

Sumber: Olahan Peneliti

Sistem menerapkan tujuh kategori *gate* yang harus dilalui sebelum transisi dieksekusi. *Gate* pertama adalah validasi *matrix* transisi yang memastikan kombinasi *state* asal dan *state* tujuan terdaftar pada matrix transisi yang valid. *Gate* kedua adalah validasi otorisasi yang memeriksa apakah pelaku transisi memiliki peran dan kepemilikan yang sesuai. *Gate* ketiga khusus untuk transisi proyek menuju *In Progress* atau Resume yang mensyaratkan status pembayaran DP pada *handover* terkait telah *PAID*. *Gate* keempat khusus untuk transisi proyek menuju *Completed* yang mensyaratkan seluruh *milestone* telah berstatus *Done*. *Gate* kelima berlaku untuk seluruh transisi *milestone*, yang mensyaratkan proyek induk berada pada *state* non-terminal. *Gate* keenam adalah validasi kewajiban alasan pada transisi menuju *Cancelled*. *Gate* ketujuh adalah validasi sistem pengulangan yang mendeteksi apabila *state* asal sama dengan *state* tujuan, sistem akan mengembalikan respons *no-operation* tanpa mencatat audit log untuk menghindari log palsu.

Setiap transisi dapat memicu *side effect* yang dieksekusi sebagai satu kesatuan transaksi tak terpisahkan. Transisi paling kompleks terjadi pada saat proyek



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

mencapai *state Completed*, yang memicu empat *side effect* berurutan. Pertama, kolom 'end_date' pada tabel projects diperbarui dengan tanggal saat ini. Kedua, baris pada tabel '*invoice_terms*' dengan tipe FINAL dan status DRAFT diperbarui dengan referensi pemicu, identitas pelaku konfirmasi, dan *timestamp* konfirmasi, kemudian dipromosikan ke status READY_TO_ISSUE melalui pemanggilan fungsi sinkronisasi siklus hidup *invoice*. Ketiga, *snapshot* KPI *preliminary* dihitung secara otomatis untuk seluruh Consultant yang terlibat dan disimpan ke tabel '*kpi_snapshots*' dengan kolom `*finalized_at*` dibiarkan NULL karena finalisasi tetap menjadi kewenangan CEO. Keempat, audit log transisi proyek dicatat ke tabel '*project_status_transitions*'. Apabila salah satu *side effect* gagal, seluruh transaksi dibatalkan melalui mekanisme *rollback* sehingga integritas data tetap terjaga.

Implementasi *state machine* dilakukan menggunakan pendekatan *custom service module* dengan *transition matrix* manual yang dipadukan dengan *middleware* permission pada level *routing*. Pendekatan ini dipilih dibandingkan dengan penggunaan pustaka *state machine* seperti XState karena memberikan kontrol penuh atas logika transisi yang spesifik dengan kebutuhan domain, sekaligus menghindari kompleksitas *dependency* yang tidak proporsional dengan skala sistem. Seluruh modul *state machine* ditempatkan pada direktori '*services/WfMS*' yang terdiri atas enam berkas, yaitu '*state-definitions.js*' sebagai sumber kebenaran definisi *state* dan *matrix* transisi, '*transition-service.js*' sebagai mesin aturan utama yang menjalankan transisi, '*preconditions.js*' yang berisi fungsi-fungsi validasi *gate*, '*authorization.js*' yang berisi validasi peran dan kepemilikan, '*audit-logger.js*' yang menulis log transisi secara *append-only*, dan '*WfMS-error.js*' yang berisi kelas error kustom beserta *mapping* kode HTTP. Setiap transisi dieksekusi melalui tujuh langkah berurutan, yaitu penguncian baris data dengan SELECT FOR *UPDATE* untuk mencegah *race condition*, validasi *matrix* transisi, validasi otorisasi, validasi *precondition*, pembaruan kolom status, pencatatan audit log, dan eksekusi *side effect*. Seluruh langkah dibungkus dalam satu transaksi basis data untuk menjamin konsistensi.

Sistem mengembalikan respons error dalam format JSON yang konsisten dengan struktur kode error, pesan, dan HTTP status yang sesuai. Pemetaan kode error

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

terhadap HTTP status disajikan pada Tabel 4.14. Pesan error dirancang untuk memberikan informasi yang cukup bagi antarmuka pengguna untuk menampilkan notifikasi yang jelas, tanpa mengekspos detail implementasi internal sistem.

Tabel 4. 14 Pemetaan Kode *Error State Machine*

| **Kode Error** | **HTTP Status** | **Skenario** |
|---|---|---|
| NOT_FOUND | 404 | Identitas proyek atau *milestone* tidak ditemukan |
| UNAUTHORIZED_ROLE | 403 | Peran atau kepemilikan pelaku tidak memenuhi syarat |
| INVALID_TRANSITION | 409 | Kombinasi *state* asal dan tujuan tidak terdaftar pada *matrix* |
| *PRECONDITION*_FAILED | 409 | *Precondition* umum tidak terpenuhi |
| DP_UNPAID | 409 | Status pembayaran DP belum PAID saat memulai atau melanjutkan proyek |
| *MILESTONES*_INCOMPLETE | 409 | Terdapat *milestone* yang belum berstatus *Done* saat menyelesaikan proyek |
| PROJECT_TERMINAL | 409 | Upaya mengubah *milestone* pada proyek yang telah mencapai *state* terminal |

Sumber: Olahan Peneliti

### 4.2.5 Perancangan API/*Endpoint*

Perancangan API bertujuan menetapkan kontrak komunikasi antara *Client Layer* dan *Application Layer*. API dirancang mengikuti arsitektur REST dengan

pertukaran data dalam format JSON melalui protokol HTTP. Pada ruang lingkup penelitian, terdapat empat puluh tujuh *endpoint* yang dikelompokkan ke dalam delapan modul fungsional yang berkaitan langsung dengan proses *monitoring* proyek dan penilaian kinerja karyawan berbasis KPI. Sistem mengadopsi pendekatan *progressive standardization*, yaitu menetapkan standar yang ketat pada area inti seperti *state machine*, RBAC, dan *audit trail*, sementara untuk *cross-cutting concerns* seperti *pagination* dan *validation library* dilakukan secara bertahap mengikuti kebutuhan nyata operasional.

Sistem mengadopsi prinsip desain RESTful dengan beberapa penyesuaian yang spesifik dengan kebutuhan domain. Pemetaan HTTP verb terhadap operasi sistem disusun sebagai berikut. GET digunakan untuk operasi pembacaan *resource* baik dalam bentuk daftar maupun detail. POST digunakan untuk pembuatan *resource* baru dan untuk *endpoint* aksi yang merepresentasikan transisi *state*. PATCH digunakan untuk pembaruan parsial pada *sub-resource*. PUT digunakan untuk penggantian penuh *state* dari *resource* yang bersifat *singleton*. DELETE tidak digunakan pada ruang lingkup penelitian karena seluruh penghapusan logis ditangani melalui transisi *state* ke *Cancelled* atau penonaktifan flag is_active, untuk menjaga integritas *audit trail*. Pendekatan ini memastikan bahwa data historis tidak pernah benar-benar hilang dari basis data, sehingga jejak audit tetap dapat ditelusuri sepanjang siklus hidup sistem.

Tabel 4. 15 Konvensi Penamaan *Endpoint* pada Sistem

| **Aspek** | **Konvensi** | **Contoh** |
|---|---|---|
| Base Path | /api/ | /api/projects |
| *Resource* Naming | Plural untuk collection | /projects, /users, /*handovers* |
| Multi-word Path | Kebab-case | /task-templates, /audit-trail |
| Sub-*resource* | Nested path | /projects/:id/*milestones*/:mId/status |



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| Aspek | Konvensi | Contoh |
|---|---|---|
| *State* Transition | Per-action POST | /projects/:id/pause, /projects/:id/complete |
| *Request Body* Field | camelCase | pmUserId, qualityRating |
| Database Column | snake_case (di-mapping di controller) | *pm_user_id*, *quality_rating* |

Sumber: Olahan Peneliti

Tabel 4. 16 Struktur Standar *Request* dan *Response*

| Komponen | Spesifikasi |
|---|---|
| *Headers* Wajib | Authorization: Bearer <jwt>, Content-Type: application/json |
| Format Sukses | { “success”: true, “data”: {...}, “message”: “...” } |
| Format Error | { “success”: false, “code”: “...”, “message”: “...” } |
| HTTP 200 OK | *Read* sukses dan aksi *idempotent* sukses |
| HTTP 201 Created | Pembuatan *resource* baru |
| HTTP 400 Bad *Request* | Validasi format atau *field* wajib |
| HTTP 401 Unauthorized | JWT hilang, kedaluwarsa, atau invalid |
| HTTP 403 Forbidden | *Permission* atau *role* tidak memenuhi (RBAC dan WfMS) |
| HTTP 404 Not Found | *Resource* tidak ditemukan |
| HTTP 409 Conflict | Pelanggaran aturan *state machine* atau *business rule* |
| HTTP 500 Internal | *Uncaught exception* atau *error* basis data |

Sumber: Olahan Peneliti



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Pemisahan antara HTTP 400 dan HTTP 409 menjadi konvensi yang penting pada sistem. Status 400 digunakan ketika *request* memiliki masalah format atau *syntax* seperti tipe data yang salah atau *field* wajib yang kosong. Status 409 digunakan ketika *request* memiliki format yang benar tetapi melanggar aturan *business state*, seperti mencoba menyelesaikan proyek yang masih memiliki *milestone* belum selesai, atau memulai proyek yang DP-nya belum lunas. Pemisahan ini memberikan kejelasan pada *Client Layer* untuk membedakan antara error yang perlu diperbaiki di sisi pengguna (format *input*) dengan error yang merupakan kondisi sistem yang valid namun tidak diperbolehkan (*state* conflict).

Tabel 4. 17 Ringkasan *Endpoint* per Modul Fungsional

| No | Modul | Base Path | Jumlah *Endpoint* | Fungsi Utama |
|---|---|---|---|---|
| 1 | Autentikasi | /api/auth | 3 | *Login*, profil pengguna, logout |
| 2 | Manajemen Pengguna | /api/users | 6 | CRUD pengguna dan lookup Consultant |
| 3 | Departemen | /api/departments | 5 | Master data struktur organisasi |
| 4 | Master Data Lookup | /api/lookup | 3 | Master data services, roles, service lines |
| 5 | Manajemen Proyek | /api/projects | 13 | CRUD proyek, assign Consultant, transisi *state* |
| 6 | Manajemen KPI | /api/kpi | 6 | Konfigurasi bobot, perhitungan, finalisasi *snapshot* |
| 7 | Task Template | /api/task-templates | 4 | Template task per service line |
| 8 | *Dashboard* | /api/*dashboard* | 4 | *Dashboard* tersegmentasi per peran |

| | |
|---|---|
| Total *Endpoint* | 47 |

Sumber: Olahan Peneliti

Sistem juga berinteraksi dengan dua modul boundary yang berada di luar ruang lingkup pengembangan, yaitu modul *Handover* yang dibaca sebagai sumber pembuatan proyek dan verifikasi pembayaran DP, dan modul Invoice yang ditulis oleh sistem ketika proyek mencapai *state* Completed untuk memicu penerbitan invoice final.

*Endpoint* pertama yang menjadi titik masuk *lifecycle* proyek adalah *endpoint* pembuatan proyek dari *handover*. *Endpoint* ini dipanggil oleh COO setelah *handover* dari modul Business Development disetujui, dengan tujuan mengonversi *handover* menjadi proyek yang siap dikelola oleh Project Manager.

Method : POST

Path : /api/projects/from-*handover*/:handoverId

Auth : JWT dan Permission PROJECT_ASSIGN_PM

Aktor : COO atau Superadmin

*Request body* memuat dua field, yaitu pmUserId sebagai identitas Project Manager yang akan ditugaskan (wajib) dan note sebagai catatan untuk *audit trail* (opsional). Pada respons sukses 201, sistem mengembalikan objek proyek lengkap dengan *state* Awaiting Consultant beserta *milestone* awal yang otomatis disalin dari *handover_milestones*. *Endpoint* ini menjalankan empat *side effect* dalam satu transaksi atomik, yaitu pembuatan baris pada tabel projects, penyalinan *milestone* dari *handover*, pembaruan status *handover* menjadi ASSIGNED_TO_PM, dan pencatatan transisi *state* proyek dengan *trigger*_source bernilai SYSTEM.

*Endpoint* kedua yang merepresentasikan pola umum transisi *state* adalah *endpoint* pause project. Pola yang sama berlaku untuk *endpoint* resume, complete, dan cancel dengan perbedaan pada *precondition* dan *side effect* masing-masing.

Method : POST



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Path : /api/projects/:projectId/pause

Auth : JWT dan Permission PROJECT_MANAGE_STATUS

Aktor : PM-of-project, COO, atau Superadmin

*Request body* memuat satu field opsional yaitu reason sebagai catatan alasan jeda. Pada respons sukses 200, sistem mengembalikan objek proyek dengan status yang telah diperbarui beserta transition_id sebagai referensi ke audit log. *Endpoint* ini menjadi titik integrasi terhadap *state machine*, sehingga seluruh validasi seperti matrix transisi, otorisasi, dan *precondition* dieksekusi di balik layar melalui modul *services/WfMS*. Apabila pengguna mencoba memause proyek yang sudah berstatus Completed, sistem akan mengembalikan respons HTTP 409 dengan kode INVALID_TRANSITION.

*Endpoint* ketiga yang menjadi sumber data dimensi *Output Quality* pada perhitungan KPI adalah *endpoint* penilaian kualitas *milestone*.

Method : PATCH

Path : /api/projects/:projectId/*milestones*/:milestoneId/rate

Auth : JWT dan Permission KPI_RATE_TASK

Aktor : PM-of-project strict

Berbeda dengan *endpoint* transisi *state*, *endpoint* ini menggunakan verb PATCH karena rating diperlakukan sebagai atribut data *milestone*, bukan sebagai transisi *state*. *Request body* memuat field qualityRating dengan rentang nilai satu sampai lima yang dapat dikonfigurasi melalui *kpi_period_config*, dan field opsional revisionCount sebagai jumlah revisi sebelum *milestone* dinyatakan selesai. Penilaian ini tidak memicu transisi *state* dan tidak secara otomatis menghitung ulang *snapshot* KPI, perhitungan KPI dilakukan secara on-demand ketika *dashboard* di-*query* atau ketika CEO melakukan finalisasi *snapshot* melalui *endpoint* POST /api/kpi/*snapshots*.

Mekanisme autentikasi sistem menggunakan JSON Web *Token* (JWT) dengan algoritma penandatanganan HS256 dan masa berlaku delapan jam per *token*. *Token*

dikirim oleh klien melalui *header* Authorization dengan skema Bearer dan disimpan pada localStorage peramban. Sistem tidak mengimplementasikan refresh *token*, sehingga pengguna diharuskan *login* ulang setelah *token* kedaluwarsa, sebuah trade-off yang dapat diterima pada konteks aplikasi internal single-tenant. Logout pada sistem bersifat *stateless* tanpa mekanisme blacklist *token*, sehingga proses logout hanya menghapus *token* dari sisi klien.

Otorisasi diterapkan dalam dua lapisan yang saling melengkapi. Lapisan pertama berupa *middleware* permission pada level *routing* yang memverifikasi keberadaan kode permission tertentu pada array permissions di *payload* JWT. Lapisan kedua berupa otorisasi WFMS pada level service yang melakukan pemeriksaan kepemilikan (ownership) dan peran spesifik per transisi. Akses efektif terhadap suatu *endpoint* merupakan irisan dari kedua lapisan tersebut. Permissions di JWT bersifat *snapshot* pada saat *login*, sehingga ketika administrator mengubah hak akses pengguna di basis data, pengguna harus *login* ulang agar perubahan tersebut berlaku.

Setiap *request* HTTP yang masuk ke Application Layer melewati rangkaian *middleware* yang dieksekusi secara berurutan. *Middleware* pertama adalah CORS yang mengontrol asal *request* yang diizinkan. *Middleware* kedua dan ketiga adalah *body* parser yang mendekode *payload* JSON dan urlencoded. *Middleware* keempat adalah static *file* handler yang melayani berkas yang telah diunggah ke direktori uploads. Setelah melewati *middleware* tingkat aplikasi, *request* diteruskan ke *middleware* tingkat route yang spesifik per *endpoint*, yaitu *middleware* autentikasi yang memverifikasi JWT dan mengisi objek req.*user*, *middleware* permission yang memeriksa kode permission yang dibutuhkan *endpoint*, dan *middleware* role atau *dashboard* variant pada *endpoint* yang memerlukannya. Setelah seluruh *middleware* lolos, controller mengeksekusi logika bisnis yang untuk *endpoint* WFMS akan memanggil modul *services/WfMS* untuk validasi *state machine*. Sistem belum mengimplementasikan global error handler, rate limiting, *request* logger terpusat, dan correlation ID untuk tracing antar *request*, yang menjadi area pengembangan lanjutan.



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Pada tahap pengembangan saat ini, sistem belum mengimplementasikan beberapa konvensi REST API yang umumnya diterapkan pada aplikasi skala besar. Pertama, sistem belum menggunakan versioning URL seperti /api/v1/ karena merupakan aplikasi internal single-tenant yang tidak menghadap konsumen eksternal. Kedua, sistem belum menerapkan pagination yang seragam pada *endpoint* pembacaan collection karena volume data masih dalam skala yang dapat ditangani dengan full-list retrieval. Ketiga, sistem menggunakan validasi manual ad-hoc pada controller alih-alih menggunakan pustaka validasi berbasis skema seperti Joi atau Zod. Keempat, sistem belum mengimplementasikan rate limiting dan correlation ID untuk tracing.

Pilihan-pilihan tersebut diambil berdasarkan prinsip progressive standardization, di mana standardisasi diterapkan pada area inti yang berdampak langsung pada integritas data dan akuntabilitas operasional, sedangkan cross-cutting concerns yang belum memberikan nilai proporsional pada skala saat ini ditunda pengembangannya sesuai prinsip YAGNI. Roadmap pengembangan lanjutan mencakup penambahan offset-based pagination, validasi berbasis skema, rate limiting, dan correlation ID untuk distributed tracing.

### 4.2.6 Perancangan Antarmuka Pengguna

Perancangan antarmuka pengguna bertujuan menetapkan bentuk visual dan struktur interaksi yang menjembatani pengguna dengan logika sistem yang telah dirancang sebelumnya. Antarmuka dikembangkan sebagai Single Page Application dengan kerangka kerja React 19 yang dipadukan dengan Tailwind CSS versi 4 untuk penataan tampilan. Perancangan ini berfokus pada tujuh belas halaman yang masuk dalam ruang lingkup penelitian, mencakup halaman autentikasi, *dashboard* tersegmentasi per peran, manajemen proyek, manajemen KPI, dan halaman konfigurasi sistem. Visualisasi rancangan disajikan dalam bentuk *wireframe* untuk menunjukkan struktur informasi dan hierarki konten, sementara detail visual seperti pemilihan warna dan tipografi.

Perancangan antarmuka sistem berlandaskan pada tiga prinsip desain utama yang saling melengkapi. Prinsip pertama adalah *role-based* segmentation, yaitu konten yang ditampilkan pada halaman tertentu disesuaikan dengan peran pengguna yang

sedang *login*. Implementasi prinsip ini paling jelas terlihat pada halaman *Dashboard* yang memiliki tujuh varian konten berbeda untuk peran CEO, COO, Project Manager, Consultant, Business Development, MEO, dan Staff Admin, namun seluruhnya dirutekan melalui URL tunggal /*dashboard*.

Prinsip kedua adalah *workflow*-driven UI, yaitu antarmuka dirancang mengikuti alur *state machine*. Hal ini diwujudkan melalui status badge berwarna sebagai indikator *state* pada setiap entitas dan tombol aksi transisi *state* yang muncul atau hilang secara kondisional sesuai *state* saat ini. Prinsip ketiga adalah permission-aware *components*, yaitu elemen antarmuka seperti tombol aksi, item menu, dan section pada halaman dirender secara kondisional berdasarkan permission yang dimiliki pengguna pada *payload* JWT. Ketiga prinsip ini dieksekusi secara konsisten dengan memanfaatkan custom *hook* useAuth yang menyediakan helper can(permission) untuk pemeriksaan permission di seluruh komponen.

Sistem mengadopsi pola layout yang terdiri atas tiga area utama, yaitu sidebar persisten pada sisi kiri, *header* pada bagian atas, dan area konten utama (PageContainer) yang mengambil sisa ruang. Sidebar dirancang sebagai komponen collapsible yang dapat diciutkan ke lebar 80 piksel atau dikembangkan ke lebar 256 piksel dengan animasi transisi 300 milidetik. Item menu pada sidebar dikelompokkan ke dalam tiga kategori, yaitu Main untuk fungsi operasional sehari-hari, System untuk konfigurasi sistem, dan Administration untuk fungsi administratif. *Header* bagian atas memiliki tinggi tetap 64 piksel dan memuat dua komponen utama, yaitu kolom pencarian (HeaderSearch) di sisi kiri dan menu pengguna (UserMenu) di sisi kanan yang berisi avatar dan dropdown profil. Area konten utama menggunakan lebar maksimum 2000 piksel yang terpusat dengan padding 24 piksel pada layar desktop dan 32 piksel pada layar lebar.

Tabel 4. 18 Design *Tokens* Sistem ERP

| **Kategori** | ***Token*** | **Nilai** | **Penggunaan** |
|---|---|---|---|
| Warna Primer | Blue | #003c90 | Brand utama, aksi primer |



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| Kategori | *Token* | Nilai | Penggunaan |
|---|---|---|---|
| | Blue Light | #0f52ba | Hover *state* |
| Warna Semantik | Success | #006544 | *State* sukses, *state* Completed/Done |
| | Warning | #a16207 | *State* menunggu (Awaiting Consultant) |
| | Danger | #c2410c | *State* terminal negatif, blocked |
| | Info | #003c90 | *State* aktif (In Progress) |
| | Neutral | #434653 | *State* pasif (On Hold, Pending) |
| Warna Netral | Text Primary | #191c1e | Teks judul dan *body* utama |
| | Text Subtle | #737784 | Teks sekunder, helper |
| | Surface | #ffffff | Background card |
| | Page Background | #fafaf9 | Background halaman |
| | Border Subtle | #eceef0 | Garis pemisah |
| Tipografi | Font Family | System sans-serif | *Default* Tailwind |
| | Heading | text-xl, text-2xl | font-semibold/font-bold |
| | *Body* | text-sm (14px) | *Default body* |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| Kategori | *Token* | Nilai | Penggunaan |
|---|---|---|---|
| | Caption | text-xs (12px) | Badge dan helper text |
| Spacing | Grid System | 4 piksel | *Default* Tailwind |
| | Container Padding | p-6 md:p-8 | 24-32 piksel |
| Border Radius | Badge | rounded-full | 9999 piksel |
| | Button | rounded-lg | 8 piksel |
| | Card | rounded-xl | 12 piksel |
| Shadow | Card | shadow-sm dan ring-1 | Subtle elevation |
| Icon *Library* | Lucide React | v1.8.0 | Ikon vektor utama |

Sumber: Olahan Peneliti

Pada Tabel 4.18 menunjukkan Daftar lengkap designs *tokens*. Design system sistem ditetapkan berdasarkan kebutuhan brand internal perusahaan dengan warna primer Blue (#003c90) sebagai identitas utama, didukung oleh palet warna semantik untuk merepresentasikan kondisi sistem. Tipografi menggunakan keluarga sans-serif *default* dari sistem operasi melalui Tailwind CSS, dengan pertimbangan untuk meminimalkan dependensi eksternal dan mempercepat waktu muat halaman. Sistem spacing mengikuti grid empat piksel sebagaimana *default* Tailwind, dan radius border diatur konsisten untuk tiga kategori elemen, yaitu badge menggunakan rounded-full, button menggunakan radius delapan piksel, dan card menggunakan radius dua belas piksel. Komponen antarmuka dibangun secara kustom dari scratch tanpa menggunakan pustaka komponen pihak ketiga seperti shadcn/ui atau Headless UI, dengan pengecualian pada *library* ikon yang menggunakan Lucide React.



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tabel 4. 19 Pemetaan *State* terhadap *Token* Warna Status Badge

| **Entitas** | ***State*** | **Background** | **Text** | **Semantik Visual** |
|---|---|---|---|---|
| Project | Awaiting Consultant | bg-amber-100 | #a16207 | Kuning - menunggu |
| | In Progress | #d5e3fc | #003c90 | Biru muda - aktif |
| | On Hold | #e0e3e5 | #434653 | Abu - pasif |
| | Completed | #006544/15 | #006544 | Hijau - sukses |
| | Cancelled | bg-orange-100 | #c2410c | Oranye - terminal negatif |
| *Milestone* | Pending | #e0e3e5 | #434653 | Abu - belum mulai |
| | In Progress | #d5e3fc | #003c90 | Biru muda - aktif |
| | Done | #006544/15 | #006544 | Hijau - sukses |
| | Blocked | bg-orange-100 | #c2410c | Oranye - masalah |

Sumber: Olahan Peneliti

Pada Tabel 4.19 menyajikan pemetaan lengkap antara *state* dengan *token* warna. Pemetaan status menjadi warna merupakan elemen kritis pada sistem karena menjadi titik temu antara *state machine* yang telah dirancang dengan visualisasi pada antarmuka. Setiap *state* proyek dan *milestone* memiliki pemetaan warna yang konsisten dan diimplementasikan melalui komponen ProjectStatusBadge dan MilestoneStatusBadge sebagai single source of truth pada keseluruhan sistem. Komponen ini direnderasi sebagai pill bulat dengan padding horizontal kecil dan

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

tipografi xs semi-bold, untuk memastikan visibilitas yang baik pada konteks tabel maupun card. Pemilihan warna mengikuti konvensi industri di mana hijau merepresentasikan penyelesaian sukses, biru merepresentasikan aktivitas berjalan, abu merepresentasikan kondisi pasif atau pending, kuning merepresentasikan kondisi menunggu, dan oranye merepresentasikan kondisi terminal negatif atau pemblokiran. Konsistensi pemetaan antara level proyek dan *milestone* disengaja, yaitu *state* In Progress pada kedua level menggunakan warna yang sama untuk memberikan asosiasi visual yang seragam. Demikian pula *state* Completed pada proyek dan Done pada *milestone* menggunakan nuansa hijau yang sama untuk menunjukkan konsep penyelesaian. Pemilihan oranye alih-alih merah untuk *state* Cancelled didasarkan pada pertimbangan psikologi warna, merah dicadangkan untuk konteks error destruktif pada dialog konfirmasi, sehingga *state* pasif seperti Cancelled menggunakan oranye yang lebih netral.

Tabel 4. 20 Inventaris Halaman dalam Ruang Lingkup Penelitian

| No | Modul | Path | Halaman | Aktor Utama |
|---|---|---|---|---|
| 1 | Autentikasi | */login* | *Login* Page | Semua peran |
| 2 | *Dashboard* | */dashboard* | *Dashboard* Page (router internal) | Semua peran |
| | | | Variant CEO | CEO |
| | | | Variant COO | COO |
| | | | Variant PM | PM |
| | | | Variant Consultant | Consultant |
| 3 | Proyek | /projects | Projects List | CEO, COO, PM, Consultant |

| No | Modul | Path | Halaman | Aktor Utama |
|---|---|---|---|---|
| | | /projects/:id/overview | Project Detail - Overview | CEO, COO, PM, Consultant |
| | | /projects/:id/*timeline* | Project Detail - *Timeline* | CEO, COO, PM, Consultant |
| | | /projects/:id/team | Project Detail - Team | CEO, COO, PM, Consultant |
| | | /projects/:id/documents | Project Detail - Documents | CEO, COO, PM, Consultant |
| | | /projects/:id/financials | Project Detail - Financials | CEO, COO, PM |
| 4 | KPI | /kpi | KPI Center | CEO, COO, PM |
| | | /kpi/consultant/:id | KPI Detail per Consultant | CEO, COO, PM, Consultant |
| 5 | Settings | /settings | Settings Landing | CEO, COO, Superadmin |
| | | /settings/kpi-config | KPI Configuration | CEO |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

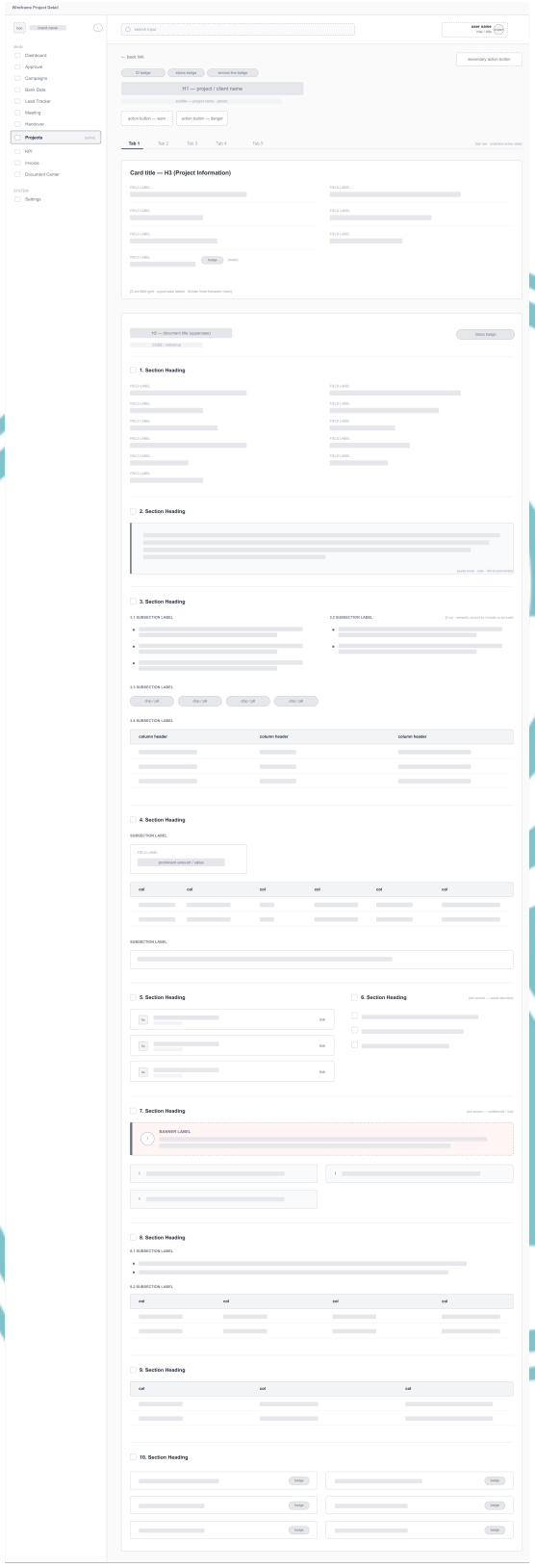
| No | Modul | Path | Halaman | Aktor Utama |
|---|---|---|---|---|
| | | /settings/task-templates | Task Templates Management | COO |
| 6 | Administration | /admin/users | *User* Management | Superadmin |
| | | /admin/departments | Department Management | Superadmin |
| | | /admin/system-settings | System Settings | Superadmin |
| Total | | | 17 Halaman | |

Sumber: Olahan Peneliti

Pada Tabel 4.20 menyajikan Manajemen *milestone* tidak memiliki halaman terpisah karena diintegrasikan secara inline pada tab Overview halaman Project Detail melalui komponen tabel *milestone* dan modal *trigger* untuk penilaian kualitas. Pendekatan ini dipilih untuk menjaga konteks proyek tetap visible saat pengguna mengelola *milestone*, sehingga mengurangi navigasi yang tidak perlu antar halaman.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



Gambar 4. 26 *Wireframe* Halaman Project Detail

Sumber: Olahan Peneliti



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

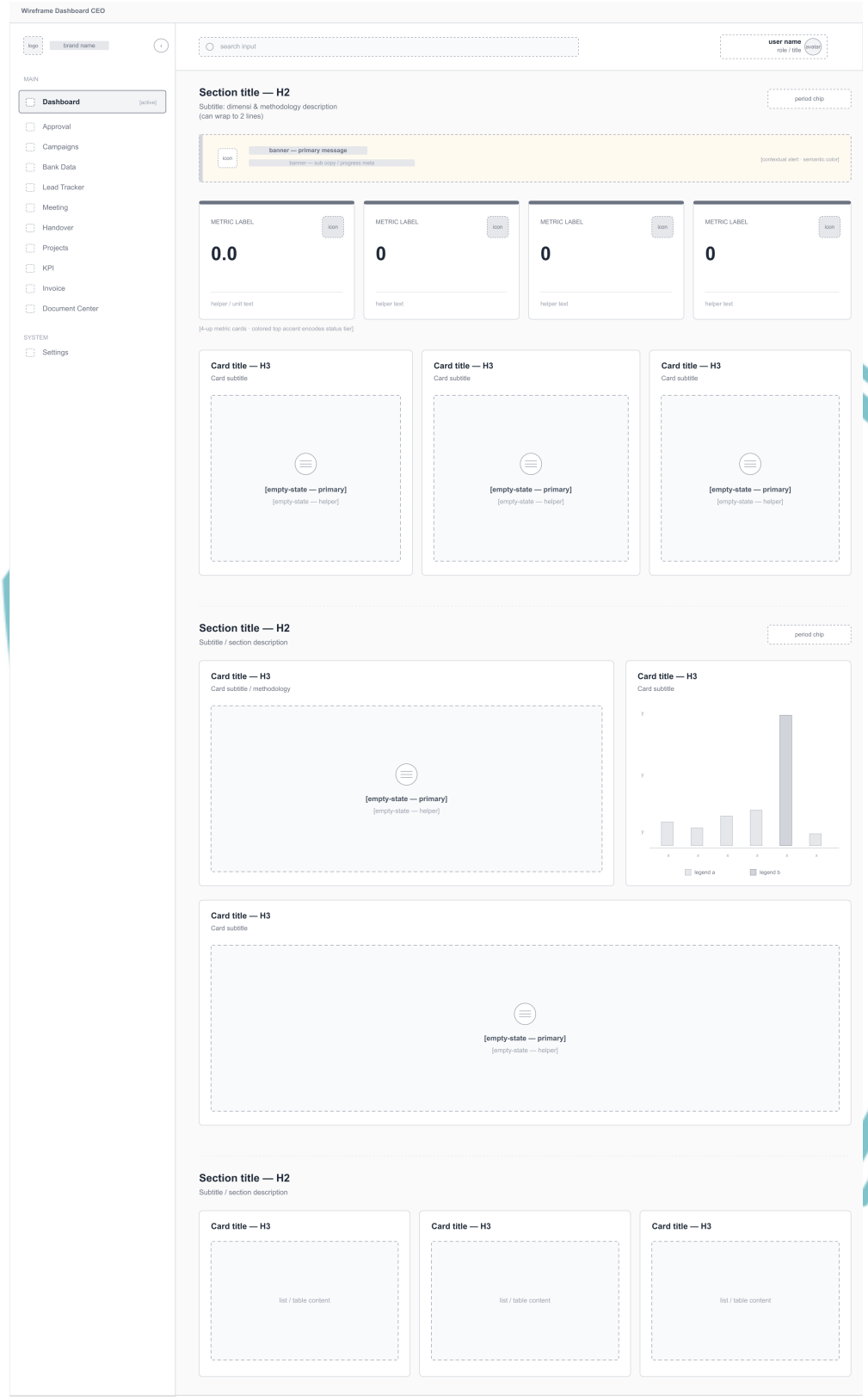
Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
   a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
   b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Pada Gambar 4.28 menunjukan Halaman Project Detail yang merupakan halaman paling kompleks pada sistem karena menjadi titik integrasi visual dari *state machine* WFMS yang telah dirancang. Halaman ini diakses melalui rute /projects/:projectId/overview yang menjadi *default* tab, dan dapat dilihat oleh seluruh aktor utama dengan konten yang disesuaikan berdasarkan permission. Layout halaman tersusun dari empat area utama secara vertikal, yaitu page *header* yang memuat tombol back, status badge, kode proyek, dan nama klien. Baris ProjectLifecycleActions yang menampilkan tombol-tombol transisi *state* secara kondisional, tab navigation untuk lima sub-halaman, dan area konten tab yang berisi project info card, quick stats, dan tabel *milestone*. Tombol aksi pada baris *lifecycle* direnderasi berdasarkan kombinasi antara *state* proyek saat ini dengan permission pengguna yang sedang *login*, sehingga hanya transisi yang valid menurut matrix pada Tabel 4.9 yang ditampilkan kepada pengguna.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta




Gambar 4. 27 *Wireframe* Halaman *Dashboard* CEO

Sumber: Olahan Peneliti



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

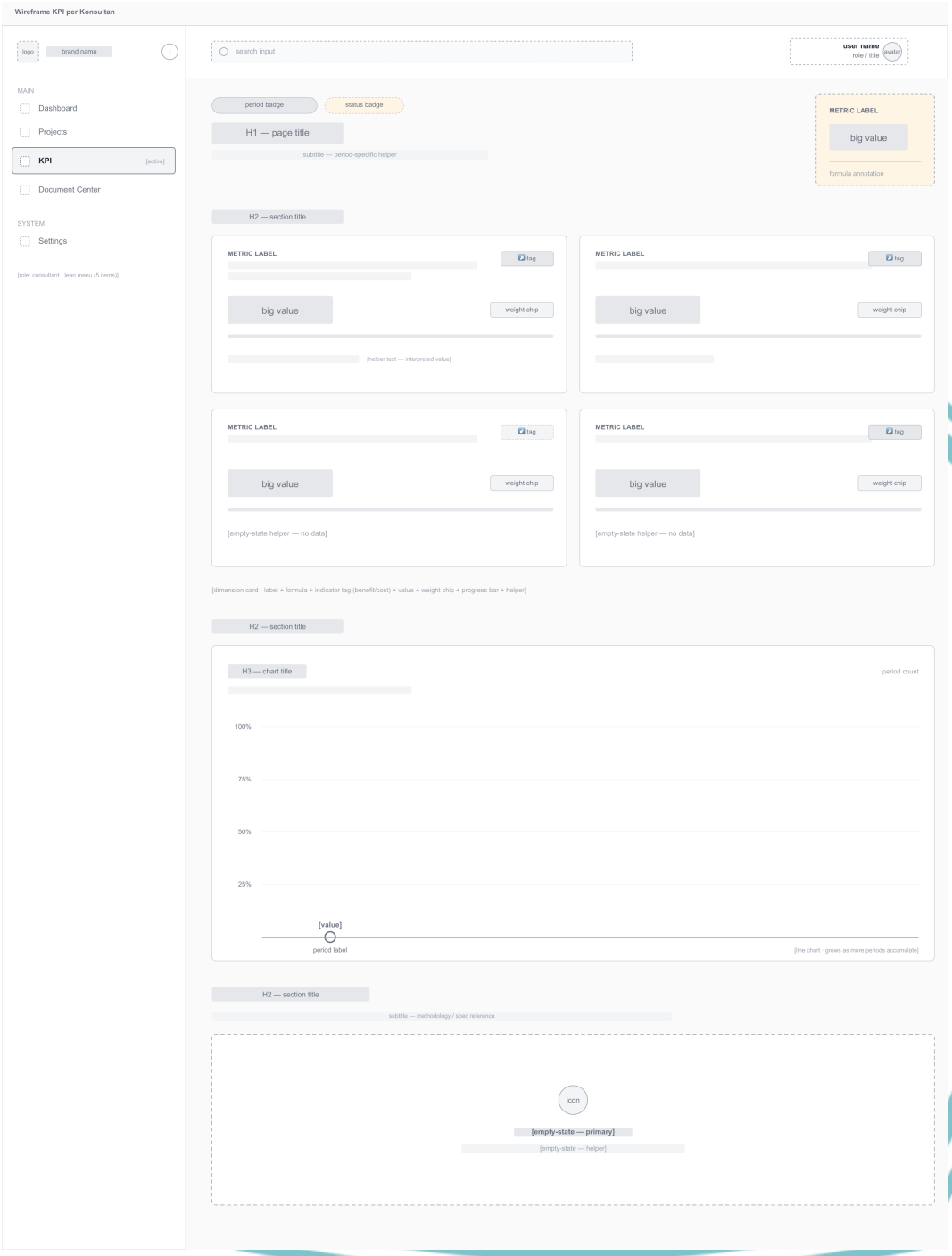
Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
   a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
   b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Pada Gambar 4.29 menunjukan *Wireframe* Halaman *Dashboard* CEO menjadi titik masuk pertama bagi peran CEO setelah berhasil *login*. Halaman ini dirancang untuk memberikan overview cepat terhadap kondisi operasional perusahaan secara keseluruhan, sehingga sebagian besar elemennya bersifat read-only agregat. Layout halaman menggunakan grid dua belas kolom yang membagi konten ke dalam empat baris fungsional. Baris pertama berisi empat KPI summary card yang menampilkan jumlah proyek aktif, proyek selesai tahun berjalan, rata-rata skor KPI, dan periode penilaian yang sedang dibuka. Baris kedua berisi dua chart yaitu KPI Trend dalam bentuk line chart enam bulan dan Project Status Distribution dalam bentuk donut chart. Baris ketiga menampilkan breakdown empat dimensi KPI dalam bentuk radar chart dan tabel top serta bottom Consultant. Baris keempat menampilkan recent activity feed yang menampilkan log aktivitas operasional terbaru pada sistem.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta




Gambar 4. 28 *Wireframe* Halaman KPI *Snapshot* per Consultant

Sumber: Olahan Peneliti

Pada Gambar 4.30 menunjukan Halaman KPI *Snapshot* per Consultant yang menyajikan hasil perhitungan KPI untuk satu Consultant pada satu periode penilaian, dan diakses melalui rute /kpi/consultant/:consultantId. Halaman ini dapat

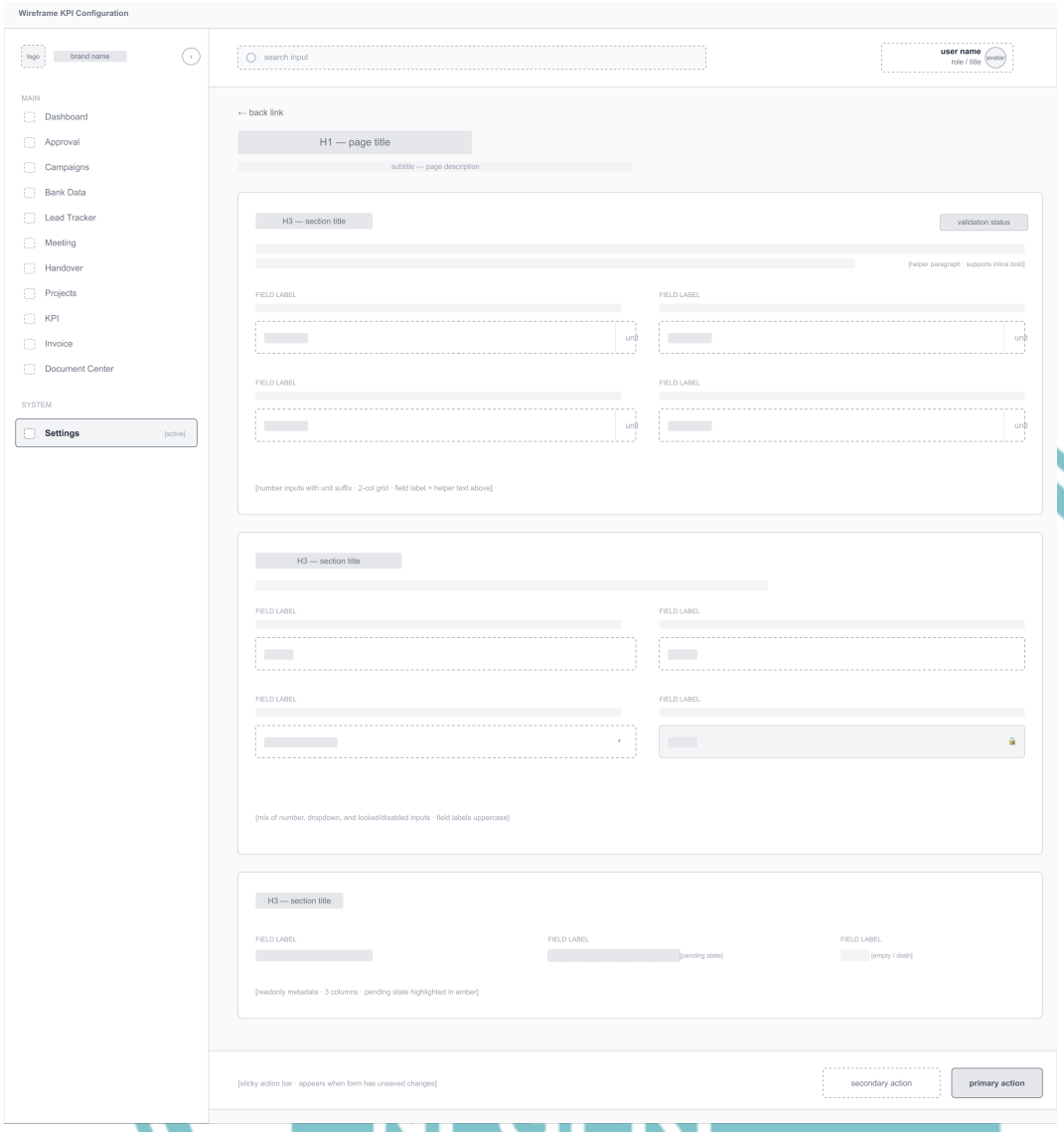


© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

dilihat oleh CEO dan COO untuk seluruh Consultant, oleh PM untuk Consultant di timnya, dan oleh Consultant untuk dirinya sendiri. Layout halaman tersusun dari lima area secara vertikal. Area pertama adalah page *header* yang menampilkan nama Consultant beserta peran dan departemennya. Area kedua adalah period selector berupa dropdown pemilihan bulan. Area ketiga adalah hero KPI Total Score yang menampilkan total skor dalam font besar disertai badge Preliminary atau Finalized untuk menunjukkan status *snapshot*. Area keempat adalah breakdown empat dimensi KPI yang ditampilkan sebagai progress bar dengan bobot masing-masing dimensi. Area kelima adalah tabel *milestone* yang berkontribusi pada perhitungan periode tersebut, sebagai bukti audit perhitungan KPI.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta




Gambar 4. 29 *Wireframe* Halaman KPI Configuration

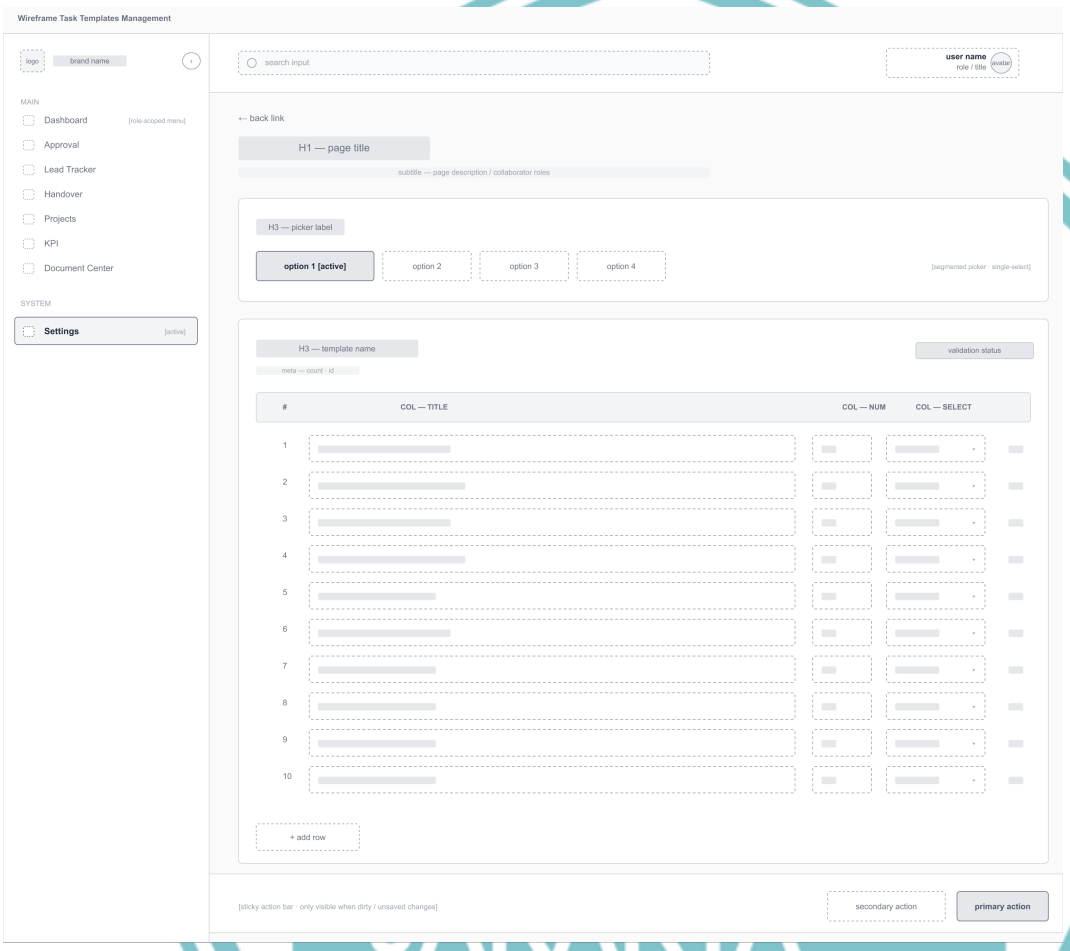
Sumber: Olahan Peneliti

Pada Gambar 4.31 menunjukan Halaman KPI Configuration menjadi titik kontrol bagi CEO untuk menetapkan parameter perhitungan KPI sebagaimana telah dirancang. Halaman ini diakses melalui rute /settings/kpi-config dan hanya dapat dibuka oleh CEO dan Superadmin dengan permission KPI_CONFIGURE. Layout halaman menggunakan pola form single-column dengan side panel untuk preview. Form utama dibagi menjadi dua section, yaitu section Weights yang berisi empat *input* numerik untuk bobot *Task Completion*, *Timeliness*, *Update Compliance*, dan *Output Quality* beserta indikator *real-time* bahwa total bobot harus berjumlah 1.00,



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

dan section Thresholds yang berisi *input* on-time tolerance dalam satuan hari, *update* gap target, skala rating kualitas, dan jenis periode. Side panel di sisi kanan menampilkan simulasi kalkulasi dengan data sampel yang otomatis ter-*update* ketika pengguna mengubah bobot, sehingga CEO dapat memvalidasi dampak perubahan bobot sebelum disimpan.




Gambar 4. 30 *Wireframe* Halaman Task Templates Management

Sumber: Olahan Peneliti

Pada Gambar 4.32 menujukkan Halaman Task Templates Management menjadi titik kontrol bagi COO untuk mengelola template task per service line yang akan otomatis disalin ke proyek baru saat *handover* dikonversi. Halaman ini diakses melalui rute /settings/task-templates dan hanya dapat dibuka oleh COO dan Superadmin dengan permission TASK_TEMPLATE_MANAGE. Layout halaman menggunakan pola split view yang membagi area konten menjadi dua kolom.



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Kolom kiri berisi panel daftar template dengan filter service line di bagian atas dan list card template yang dapat dipilih, masing-masing menampilkan nama template, service line, jumlah task, dan indikator *default*. Kolom kanan berisi editor template yang dapat diakses setelah pengguna memilih satu template, terdiri dari *input* nama template, dropdown service line, toggle *default*, dan daftar task yang dapat diurutkan ulang dengan mekanisme drag and drop berbasis pustaka dnd-kit. Setiap task memiliki atribut sequence_no, weight, phase, dan offset days.

Sistem menerapkan pola yang konsisten untuk penanganan tiga kondisi UI yang umum terjadi, yaitu loading, empty, dan error. Loading *state* diimplementasikan dengan pendekatan minimal melalui teks "Memuat..." atau skeleton block sederhana untuk loading tingkat halaman, dan tombol disabled dengan label loading untuk loading tingkat aksi. Empty *state* diimplementasikan dengan pola ilustrasi ikon Lucide berukuran besar pada kontainer bulat, disertai headline dan teks deskriptif yang menjelaskan kondisi kosong, serta call-to-action button untuk pemulihan, dengan variasi copy yang disesuaikan dengan penyebab empty *state* apakah karena filter atau karena role-scope. Error *state* diimplementasikan melalui toast notification dengan dua varian sukses dan error yang menampilkan pesan natural Bahasa Indonesia dari *backend* secara verbatim, khususnya untuk error WFMS dengan kode 409 seperti *MILESTONES*_INCOMPLETE dan INVALID_TRANSITION. Pola confirmation dialog untuk aksi destruktif seperti Cancel Project menggunakan modal dengan field reason wajib diisi, sedangkan aksi non-destruktif seperti Pause menggunakan modal dengan reason opsional.

Sistem dirancang dengan asumsi desktop dan laptop sebagai perangkat utama, sesuai dengan karakteristik aplikasi ERP internal yang dioperasikan oleh staf perusahaan dengan perangkat kerja yang terstandarisasi. Tablet didukung secara parsial melalui responsive padding pada PageContainer dengan breakpoint Tailwind *default* sm 640 piksel, md 768 piksel, dan lg 1024 piksel. Mobile bukan merupakan target utama, sehingga sidebar dengan lebar tetap akan mendominasi ruang layar pada perangkat berukuran kurang dari 640 piksel. Aksesibilitas penuh sesuai pedoman WCAG belum diaudit secara formal dan menjadi area pengembangan lanjutan, dengan pertimbangan bahwa audiens sistem adalah

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

pengguna internal yang sudah terbiasa dengan antarmuka aplikasi bisnis. Dukungan navigasi keyboard mengandalkan perilaku *default browser* dan semantik HTML standar.

Implementasi antarmuka mengadopsi filosofi minimal external dependency dengan tujuan mempermudah pemeliharaan oleh tim pengembangan internal yang berukuran kecil. *State* management sistem menggunakan React Context API yang dipadukan dengan useState dan useEffect, tanpa menggunakan pustaka eksternal seperti Redux atau Zustand. Pola pengambilan data dilakukan melalui custom *hook* yang membungkus pemanggilan service dan *state* lokal, tanpa *caching* layer global. *Routing* diimplementasikan dengan react-router versi 7 dengan pola Routes dan Route tree yang dilengkapi empat guard, yaitu GuestGuard untuk rute publik, AuthGuard untuk rute terotentikasi, PermissionGuard untuk pemeriksaan permission per rute, dan custom guard untuk modul khusus. Penanganan formulir menggunakan *native* useState tanpa pustaka seperti react-*hook*-form, dan komunikasi HTTP menggunakan *native* fetch API yang dibungkus oleh *client* kustom dengan auto- injection JWT dari localStorage. Pilihan-pilihan ini diambil secara sadar untuk memenuhi kebutuhan MVP dengan trade-off bahwa refactoring menggunakan pustaka yang lebih matang akan menjadi prioritas apabila sistem berkembang ke skala yang lebih besar.

### 4.3 Implementasi Sistem

tahap implementasi sistem perusahaan sebagai realisasi dari rancangan yang telah disusun. Implementasi dilakukan dalam dua kategori besar, yaitu implementasi *backend* yang mencakup logika bisnis, basis data, dan API, serta implementasi *frontend* yang mencakup antarmuka pengguna dan integrasi dengan API. Penyajian implementasi ini difokuskan pada modul-modul yang menjadi kontribusi inti penelitian, yaitu modul *Workflow Management System* (WfMS) sebagai mesin pengelola *lifecycle* proyek dan *milestone*, serta modul KPI Calculation sebagai mesin perhitungan penilaian kinerja Consultant. Detail implementasi disajikan dalam bentuk snippet kode terbatas yang menunjukkan keputusan teknis kunci, disertai narasi penjelasan yang menjembatani antara rancangan dengan kode aktual yang berjalan di sistem. Pembagian sub-bab disusun mengikuti alur dari lingkungan

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

pengembangan, struktur kode, implementasi modul inti, implementasi antarmuka, hingga verifikasi fungsional yang menjadi jembatan menuju tahap pengujian sistem.

### 4.3.1 Lingkungan Pengembangan

Lingkungan pengembangan sistem perusahaan disusun dengan tiga pertimbangan utama, yaitu kestabilan pustaka yang dipilih, kemudahan deployment di lingkungan operasional perusahaan, dan minimalnya dependency yang memudahkan pemeliharaan oleh tim kecil. Sistem dikembangkan sebagai *monorepo* dengan pengelolaan dependency melalui npm workspaces yang membagi kode ke dalam dua *package*, yaitu *backend*-development sebagai REST API *server* dan *frontend*-development sebagai Single Page Application. Pendekatan *monorepo* dipilih karena memudahkan sinkronisasi versi dan koordinasi perubahan antar layer, serta menyederhanakan *workflow* git melalui satu repository terpadu. Daftar pustaka utama beserta versinya disajikan pada Tabel 4.21, sedangkan ringkasan tools pendukung pengembangan disajikan pada Tabel 4.22.

Tabel 4. 21 Daftar Pustaka Utama Sistem

| **Kategori** | **Pustaka** | **Versi** | **Peran** |
|---|---|---|---|
| *Backend Runtime* | Node.js | 22 | *Runtime* JavaScript *server* |
| | Express | ^5.2.1 | *Framework* HTTP *server* |
| *Backend* Database | MySQL | 8.x | Database relasional |
| | mysql2 | ^3.21.0 | Driver MySQL dengan dukungan Promise |
| *Backend Library* | jsonwebtoken | ^9.0.3 | Penerbit dan verifier JWT |

| Kategori | Pustaka | Versi | Peran |
|---|---|---|---|
| | bcryptjs | ^3.0.3 | Hashing password |
| | cors | ^2.8.6 | Cross-origin *resource* sharing |
| | dotenv | ^17.4.1 | Environment variable loader |
| | multer | ^2.1.1 | *Middleware file* upload |
| *Backend* Testing | Jest | ^30.3.0 | Unit testing *framework* |
| | supertest | ^7.2.2 | HTTP assertion *library* |
| | nodemon | ^3.1.14 | Auto-restart pada development |
| *Frontend Framework* | React | ^19.2.4 | *Library* UI |
| | TypeScript | ~6.0.2 | Static typing |
| | Vite | ^8.0.4 | Build tool dan dev *server* |
| *Frontend Library* | Tailwind CSS | ^4.2.2 | Utility-first CSS via @tailwindcss/vite |
| | react-router | ^7.14.0 | *Client*-side *routing* |
| | Lucide React | ^1.8.0 | Icon *library* |
| | @dnd-kit/core | ^6.3.1 | Drag and drop core |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| Kategori | Pustaka | Versi | Peran |
|---|---|---|---|
| | @dnd-kit/sortable | ^10.0.0 | Sortable list extension |
| | @tiptap/react | ^2.27.2 | Rich text editor untuk meeting minutes |
| | html2canvas | ^1.4.1 | *Render* HTML ke canvas untuk PDF |
| | jspdf | ^4.2.1 | Generator PDF untuk invoice |
| *Frontend Tooling* | ESLint | ^9.39.4 | Static code analysis |
| | typescript-eslint | ^8.58.0 | TypeScript-specific lint rules |

Tabel 4. 22 Tools Pendukung Pengembangan

| Kategori | Tool | Penggunaan |
|---|---|---|
| Source Control | Git dan GitHub | Versioning kode dan kolaborasi |
| Code Editor | Visual Studio Code | Editor utama dengan ekstensi TypeScript dan ESLint |
| API Testing | Postman | Pengujian *endpoint* REST API |
| Database *Client* | phpMyAdmin | Manajemen skema dan *query* basis data |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| | | |
|---|---|---|
| Desain Antarmuka | Figma | *Wireframe* dan *mockup* UI |
| Diagram | draw.io dan Eraser.io | Diagram arsitektur, ERD, *state diagram* |
| Project Management | Notion | Dokumentasi spek dan tracking task |

Sumber: Olahan Peneliti

Implementasi basis data dilakukan mengikuti rancangan *Entity Relationship Diagram* yang telah disusun. Skema basis data dibangun melalui mekanisme migration *script* berbasis SQL yang disusun secara sequential dengan format penamaan berprefiks nomor urut dan nama fitur, sehingga setiap perubahan skema dapat ditelusuri kronologinya dan diterapkan ulang pada lingkungan baru secara konsisten. Hingga tahap implementasi saat ini, sistem memiliki tujuh belas migration *file* yang mencakup pembuatan tabel inti, indeks performansi, foreign key constraint, dan beberapa migration evolusi. Lima migration yang paling relevan dengan ruang lingkup penelitian adalah 012 projects-*feature* yang membuat tabel projects beserta enum lima *state*, 013 kpi-*feature* yang membuat tabel *kpi_period_config* dan *kpi_snapshots* beserta *seed* bobot *default*, 014 task-templates-*feature* yang membuat tabel template task, 016 project-status-transitions yang membuat tabel audit *append-only* dengan enum *trigger*_source, dan 017 project-manage-status-permission yang mengkonsolidasikan permission PROJECT_MANAGE_STATUS untuk aksi pause, resume, dan cancel. Migration diterapkan secara manual melalui MySQL CLI atau MySQL Workbench tanpa penggunaan *library* migration eksternal, sehingga developer memiliki kontrol penuh atas urutan eksekusi dan dapat melakukan *rollback* secara manual ketika diperlukan.

Sistem saat ini masih berada pada tahap pengembangan dan dijalankan di lingkungan lokal. *Backend* dijalankan melalui perintah npm run dev:*backend* yang mengaktifkan nodemon untuk auto-restart pada perubahan kode, dengan *server* mendengarkan pada port 3001. *Frontend* dijalankan melalui perintah npm run



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

dev:*frontend* yang mengaktifkan Vite development *server* pada port 5173 dengan fitur Hot Module Replacement untuk mempercepat siklus pengembangan. Komunikasi antara *frontend* dan *backend* pada lingkungan lokal menggunakan protokol HTTP dengan konfigurasi CORS yang mengizinkan origin localhost:5173. Deployment ke lingkungan produksi direncanakan setelah tahap pengujian sistem selesai menggunakan Virtual Private *Server*, dengan konfigurasi yang mencakup process manager untuk *backend*, web *server* reverse proxy untuk *frontend*, dan sertifikat SSL untuk komunikasi terenkripsi.

Sistem mengimplementasikan strategi testing minimal pada tahap pengembangan ini, terbatas pada unit test untuk fungsi utility berbasis Jest sebanyak tiga berkas, yaitu pengujian readiness checker untuk modul invoice term, lead workspace, dan action mapper invoice. Pengujian level integration untuk modul WFMS dan KPI belum diimplementasikan karena keterbatasan waktu pengembangan, dan menjadi salah satu area pengembangan lanjutan. Sistem belum mengimplementasikan pipeline Continuous Integration dan Continuous Deployment serta belum menggunakan containerization melalui Docker, dengan pertimbangan bahwa ruang lingkup penelitian difokuskan pada implementasi fungsional sistem. Penambahan CI/CD pipeline dan containerization direncanakan untuk tahap pengembangan berikutnya bersamaan dengan deployment ke lingkungan produksi.

### 4.3.2 Struktur Kode Project

Struktur kode sistem disusun mengikuti prinsip separation of concerns dengan pendekatan *feature*-based architecture pada *frontend* dan layered architecture pada *backend*. Pendekatan ini dipilih dengan pertimbangan bahwa pengelompokan kode berdasarkan fitur memudahkan pengembang baru memahami konteks bisnis pada satu domain tanpa berpindah antar folder yang terpencar, sedangkan pengelompokan berdasarkan layer pada *backend* memudahkan penerapan *cross-cutting concerns* seperti *middleware* dan *error handling* pada level *routing*. Sistem dikembangkan sebagai *monorepo* dengan npm workspaces yang berisi dua *package*, yaitu *backend-development* dan *frontend-development*. Sinkronisasi tipe data antara *backend* dan *frontend* dilakukan secara manual mengikuti kontrak API yang telah ditetapkan, tanpa *package* shared terpisah karena perbedaan *tooling*

antara *backend* yang menggunakan CommonJS JavaScript dan *frontend* yang menggunakan TypeScript dengan modul ECMAScript. Ketidaksinkronan tipe terdeteksi pada tahap integrasi melalui *error* pemanggilan API yang gagal.

Tabel 4. 23 Struktur Folder *Package backend*-development

| No | Path | Tipe | Deskripsi |
|---|---|---|---|
| 1 | *backend*-development/*package*.json | *File* | Manifest dependency *package backend* |
| 2 | src/app.js | *File* | Express bootstrap (*middleware* dan routes) |
| 3 | src/*server*.js | *File* | Entry point aplikasi (dotenv dan listen port) |
| 4 | src/config/ | Folder | Konfigurasi koneksi pool basis data |
| 5 | src/constants/ | Folder | Shared enums untuk roles dan statuses |
| 6 | src/controllers/ | Folder | HTTP route handler (lebih dari 20 *file*) |
| 7 | src/db/init.sql | *File* | Skema lengkap untuk fresh install |
| 8 | src/db/migrations/ | Folder | Incremental SQL files (17 *file*) |
| 9 | src/db/seeds/ | Folder | Demo data untuk pengembangan |
| 10 | src/*middleware*/ | Folder | *Middleware* autentikasi dan otorisasi |
| 11 | src/repositories/ | Folder | Data access layer per fitur |
| 12 | src/routes/ | Folder | Express Router instance per modul |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| No | Path | Tipe | Deskripsi |
|---|---|---|---|
| 13 | src/*services/kpi/* | Folder | Modul KPI Calculation (3 *file*) |
| 14 | src/*services/WfMS/* | Folder | Modul *Workflow Management System* (7 *file*) |
| 15 | src/utils/ | Folder | Fungsi helper lintas modul (3 *file* unit test) |

Sumber: Olahan Peneliti

Struktur kode pada *package backend*-development disusun mengikuti pola layered architecture yang memisahkan tiga tanggung jawab utama, yaitu routes sebagai entry point HTTP, controllers sebagai penghubung antara HTTP layer dan business logic, dan services sebagai tempat business logic murni yang dapat diuji tanpa dependensi HTTP. Modul *middleware* ditempatkan terpisah karena sifatnya yang cross-cutting dan dipakai oleh berbagai route. Modul *services/WfMS* merupakan modul khusus yang berisi seluruh logika *state machine* sebagaimana telah dirancang, dan dipisahkan dari controllers agar logika transisi *state* tetap bersih dari concern HTTP. Modul *services/kpi* berisi seluruh logika perhitungan KPI. Modul repositories berisi data access layer per fitur, sedangkan modul utils berisi fungsi helper yang dipakai lintas modul.

Tabel 4. 24 Struktur Folder *Package frontend*-development

| No | Path | Tipe | Deskripsi |
|---|---|---|---|
| 1 | *frontend*-development/*package*.json | *File* | Manifest dependency *package frontend* |
| 2 | tsconfig.json | *File* | Konfigurasi compiler TypeScript |
| 3 | vite.config.ts | *File* | Konfigurasi build tool Vite |
| 4 | index.html | *File* | Entry HTML aplikasi |
| 5 | src/main.tsx | *File* | Entry point React |



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| No | Path | Tipe | Deskripsi |
|---|---|---|---|
| 6 | src/app/guards/ | Folder | Auth, Permission, dan Role guards |
| 7 | src/app/layouts/ | Folder | AppShellLayout dan AuthLayout |
| 8 | src/app/navigation/ | Folder | Konfigurasi menu sidebar |
| 9 | src/app/permissions/ | Folder | Enum permissions dan role-permission mapping |
| 10 | src/app/router/ | Folder | React Router tree dan route definitions |
| 11 | src/app/store/ | Folder | Auth store berbasis Context API |
| 12 | src/assets/branding/ | Folder | Logo SVG dan brand colors |
| 13 | src/*components*/shared/ | Folder | Komponen shared app-aware (sidebar, *header*) |
| 14 | src/*components*/ui/ | Folder | Komponen primitive (toast, dialog) |
| 15 | src/*features*/auth/ | Folder | Fitur Autentikasi (*Login*) |
| 16 | src/*features*/*dashboard*/ | Folder | Per-role *dashboard* variants |
| 17 | src/*features*/kpi/ | Folder | Fitur KPI *snapshot* dan config (scope penelitian) |
| 18 | src/*features*/projects/ | Folder | Fitur Project Management (scope penelitian) |
| 19 | src/*features*/settings/ | Folder | Fitur Task Templates (scope penelitian) |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| No | Path | Tipe | Deskripsi |
|---|---|---|---|
| 20 | src/*features*/... | Folder | 14 fitur lain di luar scope penelitian |
| 21 | src/*hooks*/ | Folder | Cross-*feature hooks* (use-toast) |
| 22 | src/services/*api-client.ts* | *File* | HTTP *client wrapper* berbasis Fetch API |
| 23 | src/types/ | Folder | Cross-*feature* TypeScript type definitions |
| 24 | src/utils/ | Folder | Helper functions (date, formatters, classnames) |

Sumber: Olahan Peneliti

Struktur kode pada *package frontend*-development disusun mengikuti pola *feature*-based architecture yang mengelompokkan kode berdasarkan domain fitur, bukan berdasarkan tipe *file*. Pendekatan ini dipilih karena memudahkan pengembang menavigasi kode terkait satu fitur tanpa berpindah antar folder yang terpencar. Sistem memiliki sembilan belas folder fitur dalam direktori *features*, di mana fitur yang masuk ruang lingkup penelitian adalah projects, kpi, *dashboard*, dan settings. Setiap folder fitur memiliki struktur internal yang konsisten terdiri atas pages untuk komponen tingkat halaman, *components* untuk komponen reusable khusus fitur, *hooks* untuk custom React *hooks*, services untuk pemanggilan API, dan types untuk TypeScript type definitions. Komponen yang bersifat generic dan reusable lintas fitur seperti toast dan dialog ditempatkan di folder *components*/ui, sedangkan komponen yang bersifat shared dan app-aware seperti sidebar dan *header* ditempatkan di folder *components*/shared. Folder app berisi konfigurasi tingkat aplikasi seperti router, guards, layouts, permissions, dan global store.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tabel 4. 25 Pemetaan Layer Arsitektur dengan Lokasi Kode

| **Layer** | **Komponen** | **Lokasi Kode** |
|---|---|---|
| *Client Layer* | Single Page Application | *frontend*-development/src/ |
| | UI *Components* | *frontend*/src/*components*/, *features*/*/*components*/ |
| | *Routing* | *frontend*/src/app/router/ |
| | HTTP *Client* | *frontend*/src/services/*api-client.ts* (Fetch API *wrapper*) |
| Application Layer | REST API *Server* | *backend*-development/src/ |
| | HTTP Routes | *backend*/src/routes/ |
| | Controllers | *backend*/src/controllers/ |
| | Business Logic | *backend*/src/services/ |
| | WfMS *Engine* | *backend*/src/*services/WfMS*/ (7 *file*) |
| | KPI *Engine* | *backend*/src/*services/kpi*/ (3 *file*) |
| | *Middleware* | *backend*/src/*middleware*/ |
| Data Layer | MySQL Database | *Server* database (lokal pada tahap pengembangan) |
| | Schema Migrations | *backend*/src/db/migrations/ (17 *file*) |
| | Data Access Layer | *backend*/src/repositories/ |
| External Layer | Modul Invoice (boundary) | Read-only via controller, tidak ada modul terpisah |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| Layer | Komponen | Lokasi Kode |
|---|---|---|
| | Modul *Handover* (boundary) | Read-only via controllers/projects.controller.js |

Sumber: Olahan Peneliti

Pemetaan antara struktur kode dengan arsitektur sistem yang telah dirancang, dirangkum pada Tabel 4.25. Pemetaan ini menunjukkan bahwa keempat layer arsitektur memiliki representasi yang dapat ditelusuri pada struktur folder kode, sehingga seorang pengembang baru dapat dengan mudah mengaitkan diagram arsitektur dengan kode aktual. Karena sistem dikembangkan sebagai *monorepo*, seluruh layer berada dalam satu repository tetapi terpisah secara *package*, sehingga build dan deploy dapat dilakukan secara independen per *package*. Pemisahan modul *services/WfMS* dan *services/kpi* sebagai sub-direktori tersendiri merupakan bukti penerapan single responsibility principle pada level modul, di mana setiap modul memiliki tanggung jawab yang jelas dan tidak tercampur dengan logika modul lain.

### 4.3.3 Implementasi *Backend* dan *Database*

Implementasi *backend* sistem dibangun di atas Node.js 22 dengan *framework* Express 5 yang dikonfigurasi sebagai REST API *server stateless*. Titik masuk sistem terdiri dari dua berkas, yaitu *server*.js yang bertugas sebagai *bootstrap* minimal dengan membaca konfigurasi *environment* dan mengaktifkan *server* pada *port* yang ditentukan melalui variabel process.env.PORT (nilai *default* 3001), serta app.js yang mendefinisikan seluruh *middleware stack* dan registrasi *route group*.

Tabel 4. 26 *Middleware Stack* pada *app.js*

| No. | *Middleware* | Fungsi |
|---|---|---|
| 1 | cors({ origin: *FRONTEND*_ORIGIN, credentials: true }) | Pembatasan *origin* permintaan lintas domain berdasarkan variabel *environment* |

| No. | *Middleware* | Fungsi |
|---|---|---|
| 2 | express.json() | Penguraian *body* JSON pada setiap permintaan masuk |
| 3 | express.urlencoded({ extended: true }) | Penguraian *body* bertipe *form-urlencoded* |
| 4 | express.static('/uploads') | Penyajian berkas yang diunggah melalui *multer disk storage* |
| 5 | GET /api/health | *Endpoint healthcheck* untuk pemantauan ketersediaan *server* |
| 6 | 22 *route group* /api/* | Registrasi seluruh *route* bisnis sistem |
| 7 | *Fallback* 404 /api/* | Penanganan *path* API yang tidak ditemukan dengan respons { error: '*Endpoint* tidak ditemukan', path: req.originalUrl } |

Sumber: Olahan Peneliti

Pada Tabel 4.26 menyajikan *Middleware stack* pada app.js dieksekusi secara berurutan mengikuti tujuh lapisan. Penanganan *error* pada sistem diimplementasikan secara *per-controller* menggunakan blok try/catch di setiap *controller function*, tanpa *global error handler* terpusat pada level aplikasi. Pendekatan ini memberikan kontrol granular atas format respons *error* di setiap *endpoint*, sehingga setiap *controller* dapat menentukan kode status HTTP dan pesan *error* yang paling sesuai dengan konteks operasi yang gagal.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
const pool = mysql.createPool({
  host: process.env.DB_HOST,
  user: process.env.DB_USER,
  password: process.env.DB_PASSWORD,
  database: process.env.DB_NAME,
  waitForConnections: true,
  connectionLimit: 10,
  queueLimit: 0,
  connectTimeout: 15000,
  dateStrings: false
});
```

Gambar 4. 31 Implementasi Connection Pool MySQL pada *db.js*

Sumber: Olahan Peneliti

Pada Gambar 4.33 menyajikan *Snippet* implementasi koneksi basis data. Koneksi ke basis data MySQL dikelola menggunakan *driver* mysql2/promise yang menyediakan antarmuka berbasis Promise sehingga kompatibel secara alami dengan pola *async*/await yang digunakan di seluruh lapisan *controller* dan *service*. Konfigurasi koneksi diimplementasikan pada berkas *backend-development/src/config/db.js* menggunakan mekanisme *connection pool* dengan batas sepuluh koneksi paralel dan antrean permintaan tak terbatas. Parameter connectTimeout ditetapkan selama 15.000 milidetik untuk mengakomodasi latensi jaringan akibat basis data yang dihosting secara *remote* pada IDCloudHosting. *Fungsi* ping() diekspos sebagai *method* terpisah untuk keperluan *healthcheck monitoring*.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tabel 4. 27 *Migration File* yang Relevan dengan Ruang Lingkup Penelitian

| **Nama *Migration*** | **Cakupan** |
|---|---|
| projects-*feature* | Tabel projects, *project_consultants*, *project_milestones*, *project_milestone_updates* |
| kpi-*feature* | Tabel *kpi_period_config* dan *kpi_snapshots* beserta *seed* bobot KPI *default* |
| task-templates-*feature* | Tabel *task_templates* dan *task_template_tasks* |
| project-status-transitions | Tabel audit *append-only* *project_status_transitions* dengan *enum trigger*_source |
| project-manage-status-permission | Konsolidasi permission PROJECT_MANAGE_STATUS untuk aksi *pause, resume, dan cancel* |

Sumber: Olahan Peneliti

Pada Tabel 4.27 menyajikan lima migration yang paling relevan. Pembangunan skema basis data dilakukan melalui mekanisme *migration script* berbasis SQL yang disusun secara *sequential* dengan format penamaan berprefiks tanggal dan nomor urut. Pendekatan ini memastikan setiap perubahan skema dapat ditelusuri kronologinya dan diterapkan ulang pada lingkungan baru secara konsisten. Sistem memiliki tujuh belas *migration file* yang diterapkan secara manual melalui MySQL CLI tanpa *library migration* eksternal.

*Migration* pertama merupakan *migration* terpenting dalam ruang lingkup penelitian karena mendefinisikan struktur tabel inti sistem *monitoring* proyek. Tabel projects dirancang dengan kolom status bertipe ENUM yang membatasi nilai status pada lima *state* yang valid sesuai dengan *state machine* yang telah dirancang, yaitu

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Awaiting Consultant, In Progress, On Hold, Completed, dan Cancelled. Kolom *handover_id* ditetapkan sebagai *unique foreign key* yang mengimplementasikan relasi satu-ke-satu antara *handover* dan proyek sehingga satu *handover* tidak dapat dikonversi menjadi lebih dari satu proyek. Tabel *project_milestones* menyertakan kolom *quality_rating* bertipe TINYINT dengan *constraint* CHECK yang membatasi nilai antara 1 hingga 5 sebagai masukan dimensi *Output Quality* pada perhitungan KPI, serta kolom weight untuk bobot *milestone* yang digunakan dalam perhitungan *Task Completion* berbobot. Seluruh tabel menggunakan *storage engine* InnoDB untuk mendukung transaksi ACID dan *foreign key constraint*, serta *charset* utf8mb4 untuk kompatibilitas penuh dengan karakter Unicode termasuk karakter non-Latin.

Tiga desain keputusan teknis pada lapisan basis data perlu dijelaskan secara eksplisit. Pertama, penggunaan ENUM pada kolom status di tabel projects dan *project_milestones* mengimplementasikan validasi *state* di level basis data, bukan hanya di level aplikasi, sehingga nilai *state* yang tidak valid tidak dapat masuk ke basis data bahkan jika terdapat celah pada validasi *controller*. Kedua, pola *snapshot* pada kolom *pm_name_snapshot* dan consultant_name_*snapshot* menyimpan nama pengguna pada saat penugasan dilakukan, sehingga laporan historis tetap menampilkan nama yang benar meskipun data pengguna pada tabel users telah berubah di kemudian hari. Ketiga, tabel *project_milestone_updates* dirancang sebagai tabel *append-only* tanpa operasi *UPDATE* atau DELETE, sehingga seluruh riwayat perubahan status *milestone* tersimpan secara permanen dan menjadi sumber data yang dapat diaudit untuk perhitungan dimensi KPI *Update Compliance*.

### 4.3.4 Implementasi Autentikasi dan Hak Akses

Sistem mengimplementasikan mekanisme autentikasi berbasis JSON Web *Token* (JWT) yang dipadukan dengan mekanisme otorisasi berbasis *permission* yang dibangun di atas perancangan RBAC yang telah dirumuskan pada sub-bab 4.2.2. Implementasi autentikasi dan hak akses terdiri dari tiga komponen utama, yaitu proses *login* dan penerbitan *token* pada *auth.controller.js*, verifikasi *token* pada *middleware authenticate.js*, dan evaluasi *permission* pada *middleware require-permission.js*.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Proses autentikasi diimplementasikan pada fungsi *login* di *auth.controller.js* yang mengeksekusi enam langkah secara berurutan. Pertama, sistem memvalidasi keberadaan *field email* dan password pada *request body* dan mengembalikan respons 400 apabila salah satu tidak tersedia. Kedua, sistem melakukan *lookup* pengguna berdasarkan *email* pada basis data dan memverifikasi status aktif akun. Ketiga, sistem memverifikasi kecocokan *password* menggunakan fungsi bcrypt.compare terhadap nilai *hash* yang tersimpan dengan *cost factor* 10. Keempat, sistem mengambil daftar *permission* yang berlaku untuk peran pengguna dari basis data dan menggabungkannya dengan *permission* berbasis departemen melalui fungsi *resolveEffectivePermissions*. Kelima, sistem menerbitkan JWT yang memuat *payload* berisi identitas dan *permission snapshot* pengguna. Keenam, sistem mencatat percobaan *login* ke tabel audit dan mengembalikan *token* beserta data profil pengguna.

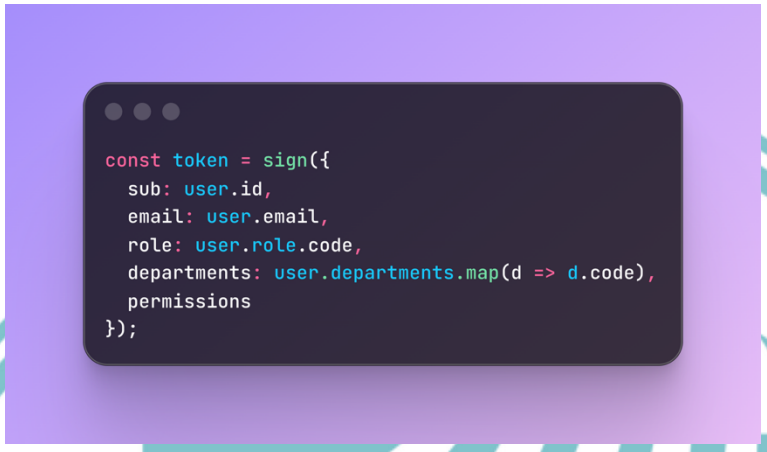
Tabel 4. 28 *Payload* JWT Sistem

| **Field** | **Tipe** | **Isi** |
|---|---|---|
| sub | Integer | ID pengguna (*primary key* tabel users) |
| *email* | String | Alamat *email* pengguna |
| role | String | Kode peran pengguna (contoh: CEO, PM, CONSULTANT) |
| departments | Array | Daftar kode departemen yang dimiliki pengguna |
| permissions | Array | *Snapshot* daftar kode *permission* yang berlaku saat *login* |

Sumber: Olahan Peneliti

Pada Tabel 4.28 menyajikan *Payload* JWT memuat lima *field.* JWT diterbitkan menggunakan *library* jsonwebtoken dengan algoritma HS256 dan *secret* yang dibaca dari variabel *environment* JWT_SECRET. Masa berlaku *token* ditetapkan

selama delapan jam yang merepresentasikan satu hari kerja, dikonfigurasi melalui variabel JWT_EXPIRES_IN.



```
const token = sign({
  sub: user.id,
  email: user.email,
  role: user.role.code,
  departments: user.departments.map(d => d.code),
  permissions
});
```

Gambar 4. 32 Implementasi Penerbitan JWT pada Proses *Login*

Sumber: Olahan Peneliti

Pada Gambar 4.34 menyajikan implementasi penerbitan JWT. Pendekatan *permission snapshot* di dalam JWT memiliki implikasi teknis yang perlu diperhatikan: perubahan *permission* yang dilakukan oleh Superadmin tidak langsung berlaku pada sesi pengguna yang sedang aktif. Perubahan tersebut baru efektif setelah pengguna melakukan *logout* dan *login* kembali untuk memperbarui *token*. Pendekatan ini dipilih untuk menghindari kompleksitas mekanisme *token rotation* dan *refresh token*, dengan pertimbangan bahwa perubahan *permission* pada lingkungan operasional perusahaan bersifat jarang dan dapat dikoordinasikan dengan pengguna yang bersangkutan.

Verifikasi identitas pengguna diimplementasikan sebagai *middleware stateless* pada berkas *authenticate.js* yang dieksekusi sebelum setiap *controller* yang memerlukan autentikasi. *Middleware* ini mengeksekusi tiga langkah secara berurutan. Pertama, *middleware* membaca *header* Authorization dan mengekstrak *token* menggunakan ekspresi reguler ^Bearer\s+(.+)$/i yang memastikan format *Bearer token* terpenuhi. Kedua, *middleware* memanggil fungsi jwt.verify() yang secara otomatis memvalidasi *signature token* dan memeriksa masa berlaku. Ketiga,



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

apabila verifikasi berhasil, *payload token* dilampirkan ke objek req.*user* untuk dikonsumsi oleh *middleware* dan *controller* berikutnya dalam rantai eksekusi.



```
const authenticate = (req, res, next) => {
  const header = req.headers['authorization'] || '';
  const match = /^Bearer\s+(.+)$/i.exec(header);
  if (!match)
    return res.status(401).json({ error: 'Missing Authorization header' });
  try {
    req.user = verify(match[1]);
    return next();
  } catch (e) {
    return res.status(401).json({ error: 'Invalid or expired token', detail:
e.message });
  }
};
```

Gambar 4. 33 Implementasi *Middleware* Autentikasi

Sumber: Olahan Peneliti

Pada Gambar 4.35 menyajikan snippet implementasi *middleware* autentikasi. Mekanisme ini bersifat sepenuhnya *stateless* karena tidak melakukan kueri ke basis data, seluruh informasi identitas dan *permission* pengguna dibaca langsung dari *payload token* yang telah terverifikasi. Pendekatan ini memberikan keuntungan latensi yang rendah per *request* karena tidak ada *round-trip* ke basis data untuk setiap pemeriksaan autentikasi.

Tabel 4. 29 Mode Evaluasi *requirePermission*

| **Mode** | **Logika** | **Penggunaan** |
|---|---|---|
| any (*default*) | Pengguna cukup memiliki salah satu dari *permission* yang dipersyaratkan | Operasi yang dapat dilakukan oleh beberapa peran berbeda |
| all | Pengguna wajib memiliki seluruh *permission* yang dipersyaratkan | Operasi yang memerlukan kombinasi kewenangan |

Sumber: Olahan Peneliti

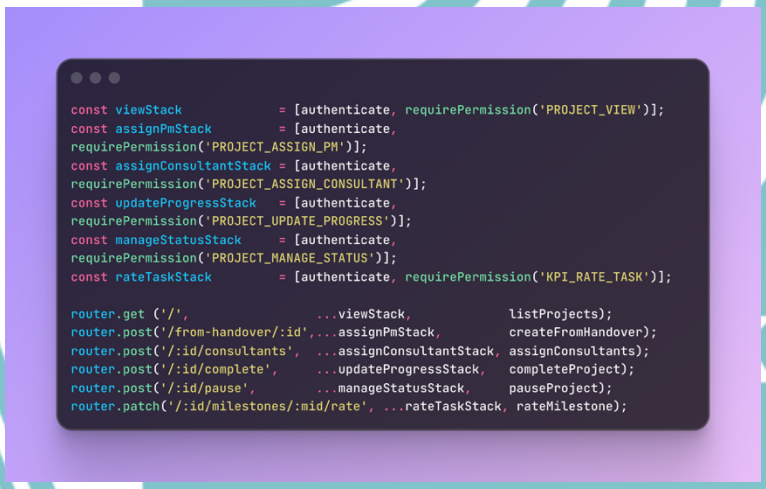


© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Pada Tabel 4.29 menyajikan *Middleware factory* mendukung dua mode evaluasi. Otorisasi pada level *route* diimplementasikan melalui *middleware factory requirePermission* pada berkas *require-permission.js*. *Middleware* ini menerima satu atau lebih kode *permission* beserta parameter mode yang menentukan logika evaluasi, kemudian mengembalikan fungsi *middleware* yang membaca daftar *permission* dari req.*user*.permissions dan mengevaluasinya terhadap *permission* yang dipersyaratkan. Apabila evaluasi *permission* gagal, *middleware* mengembalikan respons HTTP 403 dengan *body* terstruktur yang memuat kode *error* UNAUTHORIZED_ROLE, pesan penolakan, daftar *permission* yang dipersyaratkan, dan mode evaluasi yang digunakan, sehingga *frontend* dapat membedakan kegagalan autentikasi (401) dari kegagalan otorisasi (403) dan memberikan respons antarmuka yang sesuai.




Gambar 4. 34 Penerapan *Middleware* Autentikasi dan Otorisasi pada *Route* Proyek

Sumber: Olahan Peneliti

Pada Gambar 4.36 menyajikan *Snippet* penerapan pada projects.routes.js. Penerapan *middleware* pada *route* dilakukan melalui pola *stack pre-definition* di mana kombinasi authenticate dan *requirePermission* didefinisikan terlebih dahulu sebagai variabel *array* sebelum digunakan pada *route* yang relevan. Pendekatan ini meningkatkan keterbacaan kode karena seluruh persyaratan *permission* per

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

*endpoint* dapat diaudit dari satu berkas *route* tanpa perlu memeriksa implementasi *controller*.

Sistem mengimplementasikan otorisasi dua lapis sesuai dengan perancangan. Lapis pertama adalah pemeriksaan *permission* di level *route* melalui *middleware requirePermission* yang memastikan peran pengguna memiliki kewenangan untuk mengakses *endpoint* yang bersangkutan. Lapis kedua adalah pemeriksaan *ownership* di level *service* yang memastikan pengguna hanya dapat mengoperasikan data yang menjadi lingkup kewenangannya, misalnya PM hanya dapat menyelesaikan proyek yang ia pimpin dan Konsultan hanya dapat memperbarui *milestone* yang ditugaskan kepadanya. Kombinasi dua lapis ini memastikan bahwa hak akses dikendalikan baik pada dimensi tindakan maupun pada dimensi data.

### 4.3.5 Implementasi Modul Utama Sistem

Modul *Workflow Management System* (WfMS) merupakan kontribusi inti pertama dari penelitian ini, yang berfungsi sebagai mesin pengelola *lifecycle* proyek dan *milestone* sebagaimana telah dirancang. Modul ini diimplementasikan dalam direktori *backend*-development/src/*services/WfMS* yang terdiri atas tujuh berkas JavaScript dengan tanggung jawab yang terpisah, sesuai dengan prinsip single responsibility. Enam berkas pertama berisi logika inti yang dikelompokkan berdasarkan tanggung jawabnya, sedangkan berkas ketujuh berupa *index.js* yang berfungsi sebagai barrel re-export untuk menyederhanakan pemanggilan modul dari controller melalui satu pernyataan require tunggal. Implementasi WfMS dilakukan dengan pendekatan custom service module berbasis transition matrix yang dipadukan dengan *middleware* permission pada level *routing*, sebagaimana telah dijustifikasi. Penyajian implementasi dimulai dengan struktur modul, dilanjutkan dengan snippet kode inti yang menunjukkan keputusan teknis pada enam area utama, yaitu definisi *state* dan matrix transisi, fungsi eksekusi transisi, *precondition gate*, otorisasi dua lapis, audit logger, dan custom error class. Setiap snippet kode disertai narasi penjelasan yang menghubungkan implementasi dengan rancangan.

Tabel 4. 30 Struktur Modul *services/WfMS*

| No | Berkas | Tanggungjawab |
|---|---|---|
| 1 | *state-definitions.js* | Sumber kebenaran definisi *state* dan matrix transisi untuk proyek dan *milestone* |
| 2 | *transition-service.js* | Mesin aturan utama yang menjalankan tujuh langkah eksekusi transisi |
| 3 | *preconditions.js* | Fungsi-fungsi validasi *gate* seperti pemeriksaan DP dan *milestone* Done |
| 4 | *authorization.js* | Validasi peran dan kepemilikan pelaku transisi |
| 5 | *audit-logger.js* | Pencatatan transisi ke tabel audit *append-only* |
| 6 | *WfMS-error.js* | Kelas error kustom beserta mapping kode HTTP |
| 7 | *index.js* | Re-export seluruh modul untuk konsumsi dari controller |

Sumber: Olahan Peneliti

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
const PROJECT_STATES = Object.freeze({
  AWAITING_CONSULTANT: 'Awaiting Consultant',
  IN_PROGRESS: 'In Progress',
  ON_HOLD: 'On Hold',
  COMPLETED: 'Completed',
  CANCELLED: 'Cancelled'
});

const PROJECT_TRANSITION_MATRIX = Object.freeze({
  null: ['Awaiting Consultant'],

  'Awaiting Consultant': ['In Progress', 'On Hold', 'Cancelled'],
  'In Progress':         ['On Hold', 'Completed', 'Cancelled'],
  'On Hold':             ['In Progress', 'Cancelled'],
  'Completed':           [],
  'Cancelled':           []
});

const MILESTONE_STATES = Object.freeze({
  PENDING:     'Pending',
  IN_PROGRESS: 'In Progress',
  DONE:        'Done',
  BLOCKED:     'Blocked'
});

const MILESTONE_TRANSITION_MATRIX = Object.freeze({
  'Pending':     ['In Progress', 'Blocked'],
  'In Progress': ['Done', 'Blocked'],
  'Done':        ['In Progress'],
  'Blocked':     ['In Progress']
});

const isAllowedTransition = (entity, fromStatus, toStatus) => {
  const matrix = entity === 'project'
    ? PROJECT_TRANSITION_MATRIX
    : entity === 'milestone'
      ? MILESTONE_TRANSITION_MATRIX
      : null;
  if (!matrix) return false;

  const allowed = matrix[fromStatus === null ? 'null' : fromStatus];
  if (!allowed) return false;
  return allowed.includes(toStatus);
};
```

Gambar 4. 35 Implementasi *State* Definitions dan Transition Matrix

Sumber: Olahan Peneliti

Pada Gambar 4.37 menyajikan Definisi *state* dan matrix transisi diimplementasikan pada berkas *state-definitions.js* sebagai sumber kebenaran tunggal yang menjamin konsistensi *state machine* di seluruh sistem. Konstanta yang dihasilkan dibungkus dengan Object. freeze untuk menegakkan immutability pada *runtime*, sehingga *state machine* tidak dapat dimodifikasi secara dinamis dan satu-satunya cara mengubah *state machine* adalah melalui modifikasi berkas dan redeployment. Pendekatan ini dipilih untuk mencegah modifikasi *state machine* secara tidak sengaja oleh modul lain pada saat aplikasi berjalan, yang dapat menyebabkan

inkonsistensi data dan kesulitan auditing. Fungsi predikat *isAllowedTransition* menerima jenis entitas, *state* asal, dan *state* tujuan, lalu mengembalikan nilai boolean berdasarkan keberadaan kombinasi tersebut pada matrix transisi.

Penanganan kunci null pada matrix transisi proyek merepresentasikan keadaan inisialisasi, yaitu transisi dari kondisi sebelum proyek dibuat menuju *state* Awaiting Consultant. Penanganan ini mengikuti konvensi yang umum dijumpai pada *state machine library* seperti Camunda dan AASM, di mana initial *state* dimodelkan sebagai transisi dari nilai null. Berkas *state-definitions.js* juga mengekspos predikat tambahan seperti isValidStatus dan isTerminalProjectStatus yang dimanfaatkan oleh modul *preconditions* dan authorization untuk memeriksa terminal *state* proyek pada cross-entity *precondition milestone*.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
const transitionProject = async (conn, params) => {
  const {
    projectId,
    toStatus,
    actor,
    reason = null,
    triggerSource = 'USER',
    skipPreconditions = false
  } = params;

  const [[project]] = await conn.query(
    `SELECT
       p.project_id, p.project_code, p.handover_id,
       p.status, p.pm_user_id, p.end_date,
       h.dp_payment_status
     FROM projects p
     INNER JOIN handovers h ON h.handover_id = p.handover_id
     WHERE p.project_id = ?
     FOR UPDATE`,
    [projectId]
  );
  if (!project) {
    throw new WFMSError('NOT_FOUND', 'Project tidak ditemukan.');
  }

  const fromStatus = project.status;
  if (fromStatus === toStatus) {
    return { noop: true, fromStatus, toStatus, project };
  }

  if (!isAllowedTransition('project', fromStatus, toStatus)) {
    throw new WFMSError(
      'INVALID_TRANSITION',
      `Transisi project dari ${fromStatus} ke ${toStatus} tidak diperbolehkan.`
    );
  }

  if (!isAuthorizedForProjectTransition(actor, project, fromStatus, toStatus)) {
    throw new WFMSError(
      'UNAUTHORIZED_ROLE',
      `Role ${actor?.role_code || 'unknown'} tidak boleh transisi project ke
${toStatus}.`
    );
  }

  if (!skipPreconditions) {
    const pre = await checkProjectPreconditions(conn, project, toStatus);
    if (!pre.passed) {
      throw new WFMSError(pre.code || 'PRECONDITION_FAILED', pre.message);
    }
  }

  await conn.query(
    `UPDATE projects SET status = ?, updated_at = NOW() WHERE project_id = ?`,
    [toStatus, projectId]
  );

  const transitionId = await logProjectTransition(conn, {
    projectId, fromStatus, toStatus,
    userId: actor?.id || null,
    userNameSnapshot: actor?.name || null,
    triggerSource, reason
  });

  return {
    noop: false,
    fromStatus,
    toStatus,
    transitionId,
    project: { ...project, status: toStatus }
  };
};
```

Gambar 4. 36 Kode Implementasi Fungsi *transitionProject*

Sumber: Olahan Peneliti

Pada Gambar 4.38 menyajikan Fungsi *transitionProject* pada berkas *transition-service.js* merupakan satu-satunya jalur yang diperbolehkan untuk mengubah status

proyek pada sistem. Fungsi ini mengimplementasikan tujuh langkah eksekusi transisi yang telah dirancang dalam satu transaksi basis data atomik yang dikelola oleh pemanggil, sehingga apabila salah satu langkah gagal, seluruh perubahan dibatalkan melalui mekanisme *rollback* dan integritas data tetap terjaga. Langkah pertama melakukan penguncian baris data dengan klausa SELECT FOR *UPDATE* untuk mencegah *race condition* pada skenario dua pengguna memicu transisi secara bersamaan. Langkah kedua hingga keempat memvalidasi matrix transisi, otorisasi, dan *precondition*. Langkah kelima dan keenam memperbarui kolom status dan mencatat audit log. Langkah ketujuh dialihkan ke pemanggil melalui pola decoupling, sehingga *side effect* spesifik per transisi seperti pemicu invoice final pada transisi Completed tidak mengotori logika WfMS murni.

Tiga aspek implementasi pada Gambar 4.38 yang menjadi kunci kualitas sistem adalah penguncian baris dengan SELECT FOR *UPDATE* pada langkah pertama yang mencegah *race condition*, idempotensi pada baris pemeriksaan fromStatus === toStatus yang mengembalikan respons no-operation tanpa mencatat audit log palsu ketika pemanggil memicu transisi dengan *state* tujuan yang sama dengan *state* saat ini, dan eksplisit decoupling pada langkah ketujuh dengan komentar yang menegaskan bahwa *side effect* ditangani di pemanggil. Pendekatan decoupling ini menjaga modul WfMS tetap fokus pada logika *state machine* murni dan tidak bergantung pada modul lain seperti Invoice atau KPI, sehingga modul WfMS dapat dimodifikasi tanpa risiko memutus integrasi dengan modul-modul tersebut. Pola yang sama diterapkan pada fungsi *transitionMilestone* yang menangani transisi *milestone*, dengan tambahan penanganan otomatis kolom *completed_at* yang diisi pada transisi menuju Done dan direset ke NULL pada transisi reopen dari Done ke In Progress.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
const checkDpPaidForStart = (project) => {
  if (project?.dp_payment_status !== 'PAID') {
    return {
      passed: false,
      code: 'DP_UNPAID',
      message:
        'DP belum dibayar. Project belum boleh mulai — assign consultant ' +
        'akan tersedia setelah pembayaran DP dikonfirmasi.'
    };
  }
  return { passed: true };
};

const checkAllMilestonesDone = async (conn, projectId) => {
  const [rows] = await conn.query(
    `SELECT
       COUNT(*) AS total,
       SUM(CASE WHEN status = 'Done' THEN 1 ELSE 0 END) AS done_count
     FROM project_milestones
     WHERE project_id = ?`,
    [projectId]
  );
  const total = Number(rows[0]?.total ?? 0);
  const doneCount = Number(rows[0]?.done_count ?? 0);

  if (total === 0) {
    return {
      passed: false,
      code: 'PRECONDITION_FAILED',
      message: 'Project tidak punya milestone — tidak bisa dianggap Completed.'
    };
  }
  if (doneCount < total) {
    return {
      passed: false,
      code: 'MILESTONES_INCOMPLETE',
      message: `Masih ada ${total - doneCount} dari ${total} milestone yang belum
Done.`
    };
  }
  return { passed: true };
};

const checkProjectPreconditions = async (conn, project, toStatus) => {
  if (project.status === PROJECT_STATES.AWAITING_CONSULTANT &&
      toStatus === PROJECT_STATES.IN_PROGRESS) {
    return checkDpPaidForStart(project);
  }

  if (project.status === PROJECT_STATES.ON_HOLD &&
      toStatus === PROJECT_STATES.IN_PROGRESS) {
    return checkDpPaidForStart(project);
  }

  if (toStatus === PROJECT_STATES.COMPLETED) {
    return checkAllMilestonesDone(conn, project.project_id);
  }

  return { passed: true };
};
```

Gambar 4. 37 Kode Implementasi *Precondition Gate*

Sumber: Olahan Peneliti

Pada Gambar 4.39 menyajikan Setiap *precondition gate* diimplementasikan sebagai fungsi terpisah pada berkas *preconditions.js* dengan tanda tangan yang konsisten, yaitu menerima koneksi basis data dan parameter yang relevan, lalu mengembalikan objek discriminated union dengan field passed bernilai boolean. Apabila *precondition* gagal, objek tersebut juga memuat field code dan message yang dapat langsung dikonsumsi oleh WfMSError. Pola discriminated union ini

memberikan dua manfaat, yaitu pemisahan kondisi sukses dan gagal yang eksplisit pada level tipe, serta kemudahan komposisi multiple *precondition* tanpa try-catch berlapis. Dua contoh *precondition gate* yang paling sering dieksekusi adalah checkDpPaidForStart yang dipanggil pada transisi menuju In Progress serta Resume, dan checkAllMilestonesDone yang dipanggil pada transisi menuju Completed. Kedua fungsi tersebut diintegrasikan oleh fungsi aggregator checkProjectPreconditions yang dipanggil langsung oleh *transitionProject*.

```
const isPmOf = (user, project) =>
  Number(project?.pm_user_id) === Number(user?.id);

const isOwnerOf = (user, milestone) =>
  Number(milestone?.owner_user_id) === Number(user?.id);

const hasRole = (user, ...roles) =>
  roles.includes(String(user?.role_code || '').toUpperCase());

const isAuthorizedForProjectTransition = (user, project, fromStatus, toStatus) => {
  if (!user) return false;

  if (fromStatus === null && toStatus === PROJECT_STATES.AWAITING_CONSULTANT) {
    return hasRole(user, 'COO', 'SUPERADMIN');
  }

  if (toStatus === PROJECT_STATES.IN_PROGRESS) {
    return isPmOf(user, project) || hasRole(user, 'COO', 'SUPERADMIN');
  }

  if (toStatus === PROJECT_STATES.ON_HOLD) {
    return isPmOf(user, project) || hasRole(user, 'COO', 'SUPERADMIN');
  }

  if (toStatus === PROJECT_STATES.COMPLETED) {
    return isPmOf(user, project) || hasRole(user, 'SUPERADMIN');
  }

  if (toStatus === PROJECT_STATES.CANCELLED) {
    return isPmOf(user, project) || hasRole(user, 'COO', 'SUPERADMIN');
  }

  return false;
};
```

Gambar 4. 38 Kode Implementasi Otorisasi Lapisan Kedua WfMS

Sumber: Olahan Peneliti

Pada Gambar 4.40 menyajikan Otorisasi WfMS diimplementasikan pada berkas *authorization.js* sebagai lapisan kedua dari mekanisme dua lapis otorisasi yang telah dirancang. Lapisan pertama berupa *middleware* permission pada level *routing* yang memeriksa keberadaan kode permission tertentu pada *payload* JWT, sedangkan lapisan kedua pada modul WfMS berfokus pada ownership dan role-spesifik transition yang tidak dapat ditangani oleh permission generic. Fungsi *isAuthorizedForProjectTransition* menerima objek pengguna beserta proyek dan



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

menerapkan enam aturan transisi yang spesifik, mulai dari aturan inisialisasi proyek yang hanya dapat dilakukan oleh COO atau Superadmin, hingga aturan transisi menuju Completed yang memerlukan kepemilikan PM yang ketat. Tiga helper sederhana mendukung pemeriksaan ini, yaitu isPmOf yang mencocokkan id pengguna dengan *pm_user_id* proyek, isOwnerOf yang mencocokkan id pengguna dengan owner_*user*_id *milestone*, dan hasRole yang memeriksa kode peran pengguna terhadap daftar peran yang diperbolehkan.

Pendekatan *default* deny pada akhir fungsi *isAuthorizedForProjectTransition* yang mengembalikan false untuk seluruh kombinasi yang tidak ter-cover oleh aturan eksplisit merupakan praktik keamanan defensif. Pendekatan ini memastikan bahwa ketika terdapat transisi baru yang ditambahkan ke matrix tanpa pembaruan aturan otorisasi, transisi tersebut otomatis akan ditolak hingga aturan otorisasinya didefinisikan secara eksplisit, sehingga mencegah celah keamanan akibat lupa konfigurasi. Pola serupa diterapkan pada isAuthorizedForMilestoneTransition yang membedakan antara transisi normal dengan transisi reopen Done ke In Progress yang khusus diberikan kepada PM atau Superadmin.



```
const logProjectTransition = async (conn, params) => {
  const {
    projectId,
    fromStatus,
    toStatus,
    userId = null,
    userNameSnapshot = null,
    triggerSource = 'USER',
    reason = null
  } = params;

  let snapshotName = userNameSnapshot;
  if (!snapshotName && userId) {
    const [[row]] = await conn.query(
      `SELECT name FROM users WHERE id = ? LIMIT 1`,
      [userId]
    );
    snapshotName = row?.name || null;
  }

  const [result] = await conn.query(
    `INSERT INTO project_status_transitions
      (project_id, from_status, to_status,
       triggered_by_user_id, triggered_by_name_snapshot,
       trigger_source, reason)
     VALUES (?, ?, ?, ?, ?, ?, ?)`,
    [projectId, fromStatus, toStatus, userId, snapshotName, triggerSource, reason]
  );
  return result.insertId;
};
```

Gambar 4. 39 Kode Implementasi Audit Logger

Sumber: Olahan Peneliti



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Pada Gambar 4.41 menyajikan Pencatatan audit transisi *state* diimplementasikan pada berkas *audit-logger.js* melalui fungsi *logProjectTransition* yang menulis satu baris baru ke tabel *project_status_transitions* untuk setiap transisi yang dieksekusi. Fungsi ini dipanggil pada langkah keenam *transitionProject* dalam koneksi basis data yang sama, sehingga apabila audit log gagal ditulis, seluruh transaksi termasuk perubahan status akan di-*rollback*. Pola pengamanan ini memastikan bahwa setiap perubahan status pada basis data pasti memiliki jejak audit yang sesuai. Sifat *append-only* tabel audit diterapkan secara konvensi pada level kode, yaitu tidak ada path eksekusi *UPDATE* atau DELETE terhadap tabel ini di seluruh sistem. Pelanggaran konvensi ini dapat terdeteksi pada tahap code review. Fungsi juga menerapkan pattern *snapshot* dengan menyimpan kolom triggered_by_name_*snapshot*, sehingga riwayat audit tetap menampilkan nama pelaku pada saat aksi dilakukan, meskipun data referensi pada tabel users berubah di kemudian hari.

Berkas *audit-logger.js* juga menyediakan fungsi *logMilestoneTransition* yang mengikuti pola serupa untuk pencatatan transisi *milestone* pada tabel *project_milestone_updates*, beserta fungsi fetchProjectAuditTrail yang menggabungkan dan mengurutkan secara kronologis seluruh transisi project dan *milestone* pada satu proyek untuk kebutuhan halaman *Timeline* pada antarmuka Project Detail. Fungsi gabungan ini menjadi sumber data tab *Timeline* yang menampilkan riwayat *lifecycle* proyek secara lengkap kepada pengguna.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
// services/wfms/wfms-error.js
class WFMSError extends Error {
  constructor(code, message) {
    super(message);
    this.name = 'WFMSError';
    this.code = code;
  }
}

const WFMS_ERROR_HTTP_STATUS = {
  NOT_FOUND: 404,
  UNAUTHORIZED_ROLE: 403,
  INVALID_TRANSITION: 409,
  PRECONDITION_FAILED: 409,
  DP_UNPAID: 409,
  MILESTONES_INCOMPLETE: 409,
  PROJECT_TERMINAL: 409
};

const httpStatusForWFMSError = (err) =>
  WFMS_ERROR_HTTP_STATUS[err?.code] ?? 500;

// controllers/projects.controller.js
const sendWFMSError = (res, err) => {
  if (!(err instanceof wfms.WFMSError)) return false;
  const status = wfms.httpStatusForWFMSError(err);
  res.status(status).json({
    success: false,
    code: err.code,
    message: err.message
  });
  return true;
};
```

Gambar 4. 40 Kode Implementasi WFMSError dan Helper *sendWFMSError*

Sumber: Olahan Peneliti

Pada Gambar 4.42 menyajikan Penanganan error WfMS diimplementasikan melalui custom error class WFMSError pada berkas *WfMS-error.js* yang merupakan ekstensi dari kelas Error bawaan JavaScript dengan tambahan property code untuk mengidentifikasi jenis kesalahan secara deterministik. Pemetaan dari kode error ke HTTP status code diimplementasikan sebagai object literal yang dapat di-extend tanpa modifikasi kelas WFMSError, sesuai prinsip open-closed. Pada controller, helper *sendWFMSError* memeriksa apakah error yang ditangkap merupakan instance dari WFMSError, dan jika ya, mengembalikan respons JSON yang konsisten dengan format yang telah dirancang. Apabila error bukan

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

WFMSError, helper mengembalikan nilai false sehingga controller dapat melempar ulang error tersebut untuk ditangani oleh mekanisme error handler tingkat aplikasi.

Pola penggunaan helper *sendWFMSError* pada controller mengikuti struktur try-catch yang konsisten, di mana setiap pemanggilan modul service WFMS dibungkus dalam blok try yang akan menangkap WFMSError yang dilempar oleh fungsi service. Pada blok catch, helper *sendWFMSError* dipanggil terlebih dahulu, dan apabila helper mengembalikan nilai true yang berarti respons telah dikirim, controller melakukan return tanpa eksekusi lanjutan. Apabila helper mengembalikan false, error tersebut diteruskan ke handler generik untuk pemetaan ke HTTP 500. Pola ini memastikan bahwa seluruh error WFMS ditangani secara seragam pada level controller tanpa perlu menulis ulang logika mapping di setiap *endpoint*.

Modul KPI Calculation merupakan kontribusi inti kedua dari penelitian ini, yang berfungsi sebagai mesin perhitungan skor kinerja Consultant berdasarkan empat dimensi KPI yang telah dianalisis. Modul ini diimplementasikan dalam direktori *backend*-development/src/*services/kpi* yang terdiri atas tiga berkas JavaScript dengan pemisahan tanggung jawab yang jelas. Berkas *kpi-calculations.js* berisi fungsi-fungsi murni perhitungan empat dimensi KPI beserta agregasi *Weighted Scoring Method*, berkas *kpi-compute-service.js* bertanggung jawab atas orkestrasi perhitungan beserta persistensi *snapshot* ke basis data, dan berkas *index.js* berfungsi sebagai barrel re-export untuk konsumsi dari controller. Implementasi mengikuti metode *Weighted Scoring Method* yang telah dibahas, dengan bobot dimensi yang dapat dikonfigurasi secara dinamis melalui tabel *kpi_period_config*. Penyajian implementasi ini dimulai dengan struktur modul, dilanjutkan dengan implementasi empat dimensi KPI secara berurutan, kemudian *Weighted Scoring Method* beserta validasi bobot, persistensi *snapshot*, dan diakhiri dengan integrasi cross-module pada controller *completeProject* yang menunjukkan bagaimana modul KPI dipicu otomatis oleh modul WFMS saat proyek mencapai *state* Completed.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tabel 4. 31 Struktur Modul *services/kpi*

| No | Berkas | Tanggungjawab |
|---|---|---|
| 1 | *kpi-calculations.js* | Fungsi murni perhitungan empat dimensi KPI dan *Weighted Scoring Method* |
| 2 | *kpi-compute-service.js* | Orkestrasi perhitungan, pengambilan konfigurasi, dan persistensi *snapshot* |
| 3 | *index.js* | Re-export seluruh modul untuk konsumsi dari controller |

Sumber: Olahan Peneliti

Empat dimensi KPI yang telah dirumuskan, diimplementasikan sebagai fungsi-fungsi murni pada berkas *kpi-calculations.js*. Setiap fungsi menerima array *milestone* beserta parameter konfigurasi yang relevan, lalu mengembalikan objek dengan field capaian sebagai persentase pencapaian pada skala nol hingga seratus dan rawValue sebagai nilai mentah untuk keperluan audit. Pemisahan capaian dan rawValue memungkinkan *dashboard* menampilkan detail capaian sekaligus mempertahankan nilai mentah yang dapat ditelusuri kembali ke sumber data. Implementasi setiap dimensi dijelaskan secara berurutan dimulai dari *Task Completion* sebagai dimensi dengan bobot terbesar, dilanjutkan *Timeliness*, *Update Compliance*, dan diakhiri *Output Quality*.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
const MS_PER_DAY = 1000 * 60 * 60 * 24;

const computeTaskCompletion = (milestones) => {
  const totalWeight = milestones.reduce(
    (sum, m) => sum + Number(m.weight || 0), 0
  );
  if (totalWeight === 0) return { capaian: 0, rawValue: 0 };

  const doneWeight = milestones
    .filter((m) => m.status === 'Done')
    .reduce((sum, m) => sum + Number(m.weight || 0), 0);

  const ratio = doneWeight / totalWeight;
  return { capaian: ratio * 100, rawValue: ratio };
};
```

Gambar 4. 41 Kode Implementasi Perhitungan *Task Completion*

Sumber: Olahan Peneliti

Pada Gambar 4.43 menyajikan Dimensi *Task Completion* dihitung menggunakan rumus rasio antara total bobot *milestone* yang berstatus Done terhadap total bobot seluruh *milestone* yang ditugaskan kepada Consultant, dikali seratus persen. Pendekatan ini berbeda dari rasio sederhana jumlah Done terhadap jumlah total, karena memberikan kontribusi yang proporsional terhadap ukuran *milestone*. *Milestone* dengan bobot besar memberikan kontribusi yang lebih besar terhadap capaian dimensi ini, sesuai dengan prinsip bahwa tidak semua *milestone* memiliki ukuran kerja yang sama. Apabila total bobot bernilai nol yang berarti Consultant belum memiliki *milestone* sama sekali, fungsi mengembalikan capaian nol untuk menghindari pembagian dengan nol. Konstanta MS_PER_DAY yang didefinisikan pada awal berkas merepresentasikan jumlah milidetik dalam satu hari dan akan dimanfaatkan oleh dimensi *Timeliness* dan *Update Compliance* untuk mengkonversi selisih timestamp menjadi satuan hari. Pemanggilan Number(m.weight || 0) memberikan dua lapisan defensive coding, yaitu *fallback* ke nol apabila kolom weight bernilai null atau undefined, dan konversi eksplisit ke tipe numerik untuk mencegah string concatenation pada operasi penjumlahan. Pola defensive coding ini diterapkan secara konsisten pada keempat dimensi.



```
const computeTimeliness = (milestones, onTimeToleranceDays) => {
  const doneMilestones = milestones.filter(
    (m) => m.status === 'Done' && m.completedAt
  );
  if (doneMilestones.length === 0) return { capaian: 0, rawValue: 0 };

  const onTime = doneMilestones.filter((m) => {
    const target = new Date(m.targetDate).getTime();
    const completed = new Date(m.completedAt).getTime();
    if (Number.isNaN(target) || Number.isNaN(completed)) return false;
    const diffDays = (completed - target) / MS_PER_DAY;
    return diffDays <= onTimeToleranceDays;
  });

  const ratio = onTime.length / doneMilestones.length;
  return { capaian: ratio * 100, rawValue: ratio };
};
```

Gambar 4. 42 Kode Implementasi Perhitungan *Timeliness*

Sumber: Olahan Peneliti

Pada Gambar 4.44 menyajikan Dimensi *Timeliness* dihitung sebagai rasio antara jumlah *milestone* yang diselesaikan tepat waktu terhadap jumlah seluruh *milestone* yang telah berstatus Done. Definisi tepat waktu adalah selisih hari antara tanggal penyelesaian dan tanggal target tidak melebihi toleransi yang dikonfigurasi pada *kpi_period_config*, dengan nilai *default* dua hari. *Milestone* yang belum Done tidak dihitung pada dimensi ini karena belum memiliki tanggal penyelesaian. Apabila Consultant belum memiliki satupun *milestone* berstatus Done, fungsi mengembalikan capaian nol karena belum ada data yang dapat dievaluasi. Pemeriksaan Number.isNaN pada timestamp hasil konversi new Date().getTime() menangani *edge case* ketika data tanggal pada basis data berformat tidak valid, yang akan menyebabkan getTime() mengembalikan NaN. Apabila terjadi, *milestone* tersebut dianggap tidak tepat waktu secara konservatif, bukan dilewati, sehingga kualitas data yang buruk tetap tercermin pada capaian dimensi. Toleransi keterlambatan disengaja menggunakan operator <= yang berarti *milestone* yang selesai tepat pada hari target dengan toleransi positif tetap dianggap tepat waktu.



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
const computeUpdateCompliance = (milestones, targetGapDays, period) => {
  const allGaps = [];
  milestones.forEach((m) => {
    const log = filterUpdateLogByPeriod(m.updateLog || [], period);
    for (let i = 1; i < log.length; i++) {
      const prev = new Date(log[i - 1].at).getTime();
      const curr = new Date(log[i].at).getTime();
      if (!Number.isNaN(prev) && !Number.isNaN(curr) && curr >= prev) {
        allGaps.push((curr - prev) / MS_PER_DAY);
      }
    }
  });
  if (allGaps.length === 0) return { capaian: 100, rawValue: 0 };
  const avgGap = allGaps.reduce((sum, g) => sum + g, 0) / allGaps.length;
  if (avgGap === 0) return { capaian: 100, rawValue: 0 };
  const capaian = Math.min(100, (targetGapDays / avgGap) * 100);
  return { capaian, rawValue: avgGap };
};
```

Gambar 4. 43 Kode Implementasi Perhitungan *Update Compliance*

Sumber: Olahan Peneliti

Pada Gambar 4.45 menyajikan Dimensi *Update Compliance* dihitung sebagai rasio antara target gap hari pembaruan progres terhadap rata-rata gap aktual antar pembaruan, dikali seratus persen, dengan capping pada nilai seratus persen. Target gap *default* adalah tiga hari yang berarti Consultant diharapkan memperbarui progres setiap tiga hari sekali. Pendekatan capping pada seratus persen mencegah Consultant yang sangat rajin memperbarui progres mendapatkan skor melebihi seratus, sehingga tidak terjadi distorsi pada *Weighted Scoring Method* ketika dimensi dijumlahkan dengan bobot. Apabila Consultant belum memiliki pembaruan progres atau hanya memiliki satu kali pembaruan sehingga tidak ada gap yang dapat dihitung, fungsi mengembalikan capaian seratus persen sebagai *default* compliant. Pilihan *default* seratus persen untuk Consultant tanpa data gap defensible secara akademik karena Consultant baru tidak seharusnya dipenalti karena belum cukup data untuk dievaluasi. Filosofi ini konsisten dengan prinsip penanganan *edge case* yang diterapkan pada keempat dimensi, yaitu kondisi tanpa data dikembalikan sebagai *default* yang netral atau compliant alih-alih sebagai nol yang dapat diinterpretasikan sebagai kinerja buruk. Fungsi helper filterUpdateLogByPeriod yang dipanggil pada awal iterasi memastikan hanya pembaruan progres yang jatuh dalam rentang periode yang relevan yang

dimasukkan ke dalam perhitungan, sehingga *snapshot* bulanan tidak terdistorsi oleh pembaruan dari periode sebelumnya.



```
const computeOutputQuality = (milestones, qualityRatingScale = 5) => {
  const rated = milestones.filter((m) => m.qualityRating != null);
  if (rated.length === 0) return { capaian: 0, rawValue: 0 };

  const sum = rated.reduce(
    (acc, m) => acc + Number(m.qualityRating || 0), 0
  );
  const avg = sum / rated.length;

  return {
    capaian: (avg / qualityRatingScale) * 100,
    rawValue: avg
  };
};
```

Gambar 4. 44 Kode Implementasi Perhitungan *Output Quality*

Sumber: Olahan Peneliti

Pada Gambar 4.46 menyajikan Dimensi *Output Quality* dihitung sebagai rata-rata rating kualitas yang diberikan oleh Project Manager pada *milestone* yang telah Done, dibagi dengan skala maksimum yang dikonfigurasi pada *kpi_period_config* dengan nilai *default* lima, lalu dikali seratus persen. *Milestone* dengan qualityRating bernilai null, yaitu yang belum dinilai oleh Project Manager, di-skip dari perhitungan rata-rata alih-alih dianggap nol. Keputusan ini diambil karena nilai null merepresentasikan ketiadaan penilaian, bukan penilaian buruk, sehingga memperlakukan null sebagai nol akan menyebabkan distorsi penilaian yang merugikan Consultant secara tidak adil. Konsekuensinya, apabila hanya satu dari lima *milestone* yang dinilai PM, rata-rata dihitung dari satu data tersebut. Implementasi disajikan pada Gambar 4.40. Penggunaan operator != null pada filter dengan sengaja menggunakan loose equality yang akan mencocokkan baik null maupun undefined, sehingga fungsi tetap robust terhadap dua kemungkinan representasi data kosong pada basis data. Parameter qualityRatingScale diberikan *default* value lima sebagai *fallback* apabila konfigurasi tidak menyediakan skala, mengikuti konvensi skala penilaian lima poin yang umum digunakan pada penilaian kinerja. Berbeda dengan dimensi *Update Compliance* yang mengembalikan seratus persen pada kondisi tanpa data, dimensi *Output Quality* mengembalikan nol karena



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

penilaian kualitas memang menjadi tanggung jawab eksplisit Project Manager dan ketiadaannya tidak dapat diinterpretasikan sebagai compliant.

```
// services/kpi/kpi-calculations.js
const computeKpiTotal = (weights, capaian) => {
  const totalWeight =
    Number(weights.taskCompletion || 0) +
    Number(weights.timeliness || 0) +
    Number(weights.updateCompliance || 0) +
    Number(weights.outputQuality || 0);
  if (totalWeight === 0) return 0;

  const numerator =
    Number(weights.taskCompletion || 0)   * Number(capaian.taskCompletion || 0) +
    Number(weights.timeliness || 0)       * Number(capaian.timeliness || 0) +
    Number(weights.updateCompliance || 0) * Number(capaian.updateCompliance || 0) +
    Number(weights.outputQuality || 0)    * Number(capaian.outputQuality || 0);

  return numerator / totalWeight;
};

// controllers/kpi.controller.js
const inRange = (v) => Number.isFinite(v) && v >= 0 && v <= 1;
if (!inRange(w.tc) || !inRange(w.tm) || !inRange(w.uc) || !inRange(w.oq)) {
  return res.status(400).json({
    success: false, message: 'Setiap bobot harus 0..1.'
  });
}
const sum = w.tc + w.tm + w.uc + w.oq;
if (Math.abs(sum - 1) > 0.01) {
  return res.status(400).json({
    success: false,
    message: `Total bobot harus 1.0 (sekarang ${sum.toFixed(3)}).`
  });
}
```

Gambar 4. 45 Kode Implementasi *Weighted Scoring Method* dan Validasi Bobot

Sumber: Olahan Peneliti

Pada Gambar 4.47 menyajikan Skor total KPI dihitung menggunakan *Weighted Scoring Method* yang menggabungkan capaian empat dimensi dengan bobot yang berasal dari *kpi_period_config*. Implementasi pada fungsi *computeKpiTotal* sengaja melakukan auto-normalisasi dengan membagi numerator terhadap total bobot, alih-alih hardcoding pembagi nilai satu. Pendekatan ini memastikan perhitungan tetap akurat meskipun total bobot yang dikonfigurasi CEO tidak persis bernilai satu, misalnya akibat floating point pada penyimpanan basis data. Setiap operasi pengambilan nilai bobot dan capaian dibungkus dengan Number() dan *fallback* || 0 untuk menjamin perhitungan tetap valid meskipun ada field yang null atau undefined. Validasi total bobot harus sama dengan satu, dengan toleransi nol koma nol satu untuk safety floating point, dilakukan pada controller saat CEO menyimpan konfigurasi baru. Pengambilan konfigurasi dilakukan dengan *query* langsung pada saat perhitungan dieksekusi, mengambil config_id terbaru sebagai konfigurasi



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

aktif. *Snapshot* yang dihasilkan menyimpan *config_id_used* sebagai jejak audit yang memungkinkan rekonstruksi perhitungan dengan parameter yang berlaku pada saat *snapshot* dibuat. Fungsi aggregator *computeFullKpi* pada berkas yang sama mengorkestrasi pemanggilan keempat fungsi dimensi diikuti dengan pemanggilan *computeKpiTotal*, lalu mengembalikan satu objek terstruktur yang berisi capaian per dimensi beserta total skor.

```
const computeAndStorePreliminaryForConsultant = async (conn, params) => {
  const { consultantUserId, consultantName, period, config } = params;

  if (await isSnapshotFinalized(conn, consultantUserId, period)) {
    return { skipped: true, reason: 'ALREADY_FINALIZED' };
  }

  const milestones = await fetchConsultantMilestones(conn, consultantUserId);
  if (milestones.length === 0) {
    return { skipped: true, reason: 'NO_MILESTONES' };
  }

  const result = computeFullKpi(milestones, config, period);

  await conn.query(
    `INSERT INTO kpi_snapshots
      (consultant_user_id, consultant_name_snapshot, period, config_id_used,
       capaian_task_completion, capaian_timeliness,
       capaian_update_compliance, capaian_output_quality,
       total_score, computed_at, finalized_at, finalized_by_user_id)
     VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, NOW(), NULL, NULL)
     ON DUPLICATE KEY UPDATE
       consultant_name_snapshot  = VALUES(consultant_name_snapshot),
       config_id_used            = VALUES(config_id_used),
       capaian_task_completion   = VALUES(capaian_task_completion),
       capaian_timeliness        = VALUES(capaian_timeliness),
       capaian_update_compliance = VALUES(capaian_update_compliance),
       capaian_output_quality    = VALUES(capaian_output_quality),
       total_score               = VALUES(total_score),
       computed_at               = NOW()
       -- finalized_at & finalized_by_user_id NOT touched (preserve NULL state)`,
    [
      consultantUserId, consultantName, period, config.configId,
      Number(result.capaian.taskCompletion.toFixed(2)),
      Number(result.capaian.timeliness.toFixed(2)),
      Number(result.capaian.updateCompliance.toFixed(2)),
      Number(result.capaian.outputQuality.toFixed(2)),
      Number(result.total.toFixed(2))
    ]
  );

  return { skipped: false, snapshot: result };
};
```

Gambar 4. 46 Kode Implementasi Persistensi *Snapshot* KPI

Sumber: Olahan Peneliti

Pada Gambar 4.48 menyajikan Hasil perhitungan KPI dipersistensi ke tabel *kpi_snapshots* melalui fungsi computeAndStorePreliminaryForConsultant pada berkas *kpi-compute-service.js*. Fungsi ini menggunakan pernyataan INSERT ON DUPLICATE KEY *UPDATE* dengan composite unique key pada kolom consultant_*user*_id dan period, sehingga pemanggilan berulang untuk Consultant

dan periode yang sama akan memperbarui baris yang sudah ada alih-alih menghasilkan duplikasi. Sebelum melakukan *upsert*, fungsi memeriksa apakah *snapshot* sudah dalam *state* finalized melalui pemeriksaan kolom *finalized_at*, apabila sudah finalized, fungsi mengembalikan respons skipped untuk mencegah modifikasi *snapshot* yang sudah dikunci oleh CEO. Klausa *UPDATE* pada pernyataan *upsert* secara sengaja tidak menyentuh kolom *finalized_at* dan finalized_by_*user*_id, sehingga *snapshot* yang sudah finalized tidak akan kembali ke *state* preliminary akibat recompute. Invariant ini menjamin bahwa keputusan finalisasi oleh CEO bersifat permanen dan dapat menjadi dasar audit yang dapat dipercaya. Implementasi disajikan pada Gambar 4.42. Komentar SQL pada klausa terakhir yang menegaskan *finalized_at* & finalized_by_*user*_id NOT touched (preserve NULL *state*) bukan sekadar dokumentasi melainkan kontrak desain yang penting. Tanpa baris ini, klausa *UPDATE* bisa salah ditambahkan oleh developer berikutnya yang menyertakan *finalized_at* = VALUES(*finalized_at*) dan secara tidak sengaja menghapus finalisasi CEO. Pembulatan dua desimal pada penyimpanan capaian menggunakan Number(x.toFixed(2)) memberikan presisi yang cukup untuk pelaporan tanpa membuang data mentah, karena rawValue per dimensi tidak dipersistensi dan dihitung ulang dari *milestone* data setiap kali dibutuhkan.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
const completeProject = async (req, res) => {
  const transitionResult = await wfms.transitionProject(conn, {
    projectId, toStatus: 'Completed', actor, reason: note || null
  });
  const project = transitionResult.project;

  const todayIso = new Date().toISOString().slice(0, 10);
  await conn.query(
    `UPDATE projects SET end_date = ? WHERE project_id = ?`,
    [todayIso, projectId]
  );

  const [triggerResult] = await conn.query(
    `UPDATE invoice_terms
     SET trigger_reference_value = 'Project completed',
         trigger_confirmed_by = ?,
         trigger_confirmed_at = NOW(),
         updated_at = CURRENT_TIMESTAMP
     WHERE project_id = ?
       AND term_type = 'FINAL'
       AND status = 'DRAFT'`,
    [actor.id, projectId]
  );
  const triggeredInvoiceTerms = triggerResult.affectedRows ?? 0;

  const [accountRows] = await conn.query(
    `SELECT DISTINCT account_id FROM invoice_terms
     WHERE project_id = ? AND term_type = 'FINAL'`,
    [projectId]
  );
  for (const row of accountRows) {
    await syncInvoiceTermLifecycle(conn, row.account_id);
  }

  let kpiComputeResults = [];
  try {
    const [consultantRows] = await conn.query(
      `SELECT consultant_user_id, consultant_name_snapshot
       FROM project_consultants WHERE project_id = ?`,
      [projectId]
    );
    if (consultantRows.length > 0) {
      const period = kpiService.formatPeriodFromDate(new Date());
      kpiComputeResults = await kpiService.computeAndStorePreliminarySnapshots(conn,
{
        consultants: consultantRows.map((c) => ({
          userId: c.consultant_user_id,
          name: c.consultant_name_snapshot
        })),
        period
      });
    }
  } catch (kpiErr) {
    console.warn('[projects.controller] KPI auto-compute failed:', kpiErr.message);
  }
};
```

Gambar 4. 47 Kode Implementasi *Side Effects completeProject*

Sumber: Olahan Peneliti

Pada Gambar 4.49 menyajikan Perhitungan KPI pada sistem dipicu melalui dua mekanisme yang saling melengkapi. Mekanisme pertama adalah pemicu otomatis pada saat proyek mencapai *state* Completed, yang diimplementasikan sebagai langkah ketujuh *side effects* di controller *completeProject*. Mekanisme kedua adalah pemicu manual oleh CEO melalui *endpoint* Finalize Period dan Recompute yang memungkinkan CEO meninjau seluruh *snapshot* di akhir periode dan menetapkan finalisasi. Sistem tidak mengimplementasikan cron job atau scheduler

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

periodik karena perhitungan KPI dapat dipicu on-demand dan latency perhitungan per Consultant masih dalam rentang yang dapat diterima, yaitu di bawah seratus milidetik untuk satu Consultant satu periode pada dataset internal saat ini. Implementasi langkah ketujuh pada *completeProject* yang menunjukkan integrasi cross-module antara modul WFMS, Invoice, dan KPI.

Empat langkah *side effects* menunjukkan pola integrasi cross-module yang berbeda-beda berdasarkan tingkat kekritisan. Langkah pertama berupa *update* kolom end_date dieksekusi dalam transaksi yang sama dengan transisi WFMS sehingga kegagalan langkah ini akan me-*rollback* seluruh proses completion. Selanjutnya langkah yang memicu invoice final juga dieksekusi dalam transaksi yang sama karena keterkaitan bisnis yang erat antara proyek Completed dengan aktivasi invoice final. Selanjutnya langkah yang memicu perhitungan KPI dibungkus dalam blok try-catch terpisah dengan komentar eksplisit bahwa kegagalan bersifat non-critical, sehingga kegagalan perhitungan KPI tidak membatalkan completion proyek. Filosofi ini defensible karena KPI bersifat derivatif yang dapat dihitung ulang kapan saja melalui *endpoint* manual, sedangkan completion proyek dan *trigger* invoice merupakan event satu kali yang harus konsisten. Langkah terakhir berupa activity log mengikuti pola yang sama dengan langkah sebelumnya, yaitu non-critical agar masalah pencatatan log tidak memblokir alur bisnis utama.

Tabel 4. 32 Penanganan *Edge Case* pada Perhitungan KPI

| No | ***Edge Case*** | **Penanganan Sistem** |
|---|---|---|
| 1 | Consultant tanpa *milestone* di periode | *Snapshot* tidak dibuat, mengembalikan skipped: true dengan reason NO_*MILESTONES* |
| 2 | *Milestone* berstatus In Progress di akhir periode | Counted as not done pada *Task Completion*, di-skip dari *Timeliness* karena belum punya completedAt |
| 3 | *quality_rating* bernilai null | Di-filter dari rata-rata *Output Quality*, tidak di-treat sebagai nol |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| No | *Edge Case* | Penanganan Sistem |
|---|---|---|
| 4 | Consultant belum memiliki gap pembaruan progres | *Update Compliance* mengembalikan capaian seratus persen sebagai *default* compliant |
| 5 | Konfigurasi bobot diubah CEO setelah *snapshot* dibuat | *Snapshot* lama tetap menyimpan *config_id_used*, recompute pakai config terbaru |
| 6 | *Snapshot* sudah finalized saat recompute dipicu | Fungsi mengembalikan skipped: true dengan reason ALREADY_FINALIZED, tidak ada *UPDATE* pada baris |

Sumber: Olahan Peneliti

Pada Tabel 4.32 menyajikan Sistem menangani enam *edge case* yang dapat terjadi pada perhitungan KPI dengan pendekatan yang konsisten dengan filosofi tidak menalti ketiadaan data, yaitu kondisi tanpa data dikembalikan sebagai *default* yang netral atau compliant alih-alih sebagai nol. Pendekatan ini defensible secara akademik karena nilai nol dapat diinterpretasikan sebagai kinerja buruk, padahal pada konteks *edge case* yang sebenarnya adalah ketiadaan data yang cukup untuk dievaluasi.. Kombinasi penanganan *edge case* ini dengan invariant *snapshot* immutability dan tabel audit *append-only* menjadikan sistem perhitungan KPI dapat dipertahankan ketika terjadi dispute, karena CEO dapat menunjukkan *config_id_used* pada *snapshot* dan jejak transisi yang mendasari perhitungan.

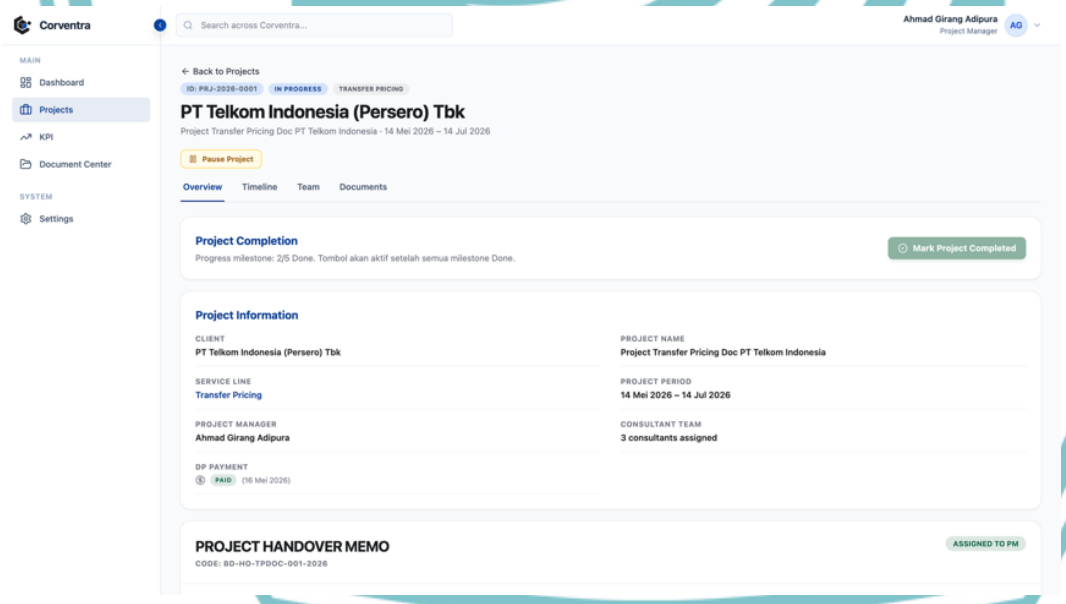
### 4.3.6 Implementasi Antarmuka Pengguna

Antarmuka pengguna sistem diimplementasikan dengan React versi 19 yang dipadukan dengan TypeScript 6 untuk type safety, Tailwind CSS versi 4 untuk styling berbasis utility class, dan React Router versi 7 untuk navigasi antar halaman. Pemilihan stack ini telah dijustifikasi, dengan tambahan keputusan tidak menggunakan *state* management *library* eksternal seperti Redux atau Zustand. Sebagai gantinya, sistem menggunakan kombinasi useState, useReducer, dan

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Context API bawaan React, yang dianggap memadai untuk skala aplikasi internal saat ini. Komunikasi dengan *backend* dilakukan melalui custom HTTP *client wrapper* berbasis *native* Fetch API pada berkas src/services/*api-client.ts*, sebagaimana telah dibahas. Struktur kode antarmuka mengikuti pola *feature*-based folder yang memisahkan tiap fitur ke direktori src/*features*/ masing-masing, dengan komponen primitive yang dapat digunakan lintas fitur diletakkan pada src/*components*/ui/. Penyajian implementasi antarmuka difokuskan pada lima halaman utama yang merepresentasikan ruang lingkup penelitian, yaitu Project Detail untuk *monitoring* proyek, *Dashboard* CEO untuk overview KPI organisasi, My KPI untuk *snapshot* kinerja Consultant, KPI Configuration untuk pengaturan bobot dan *threshold*, serta Task Templates untuk pengelolaan template *milestone* per service line. Setiap halaman disajikan dengan tangkapan layar dari sistem yang sudah berjalan beserta narasi yang menghubungkan elemen antarmuka dengan rancangan.




Gambar 4. 48 Halaman Project Detail dengan Tab Overview

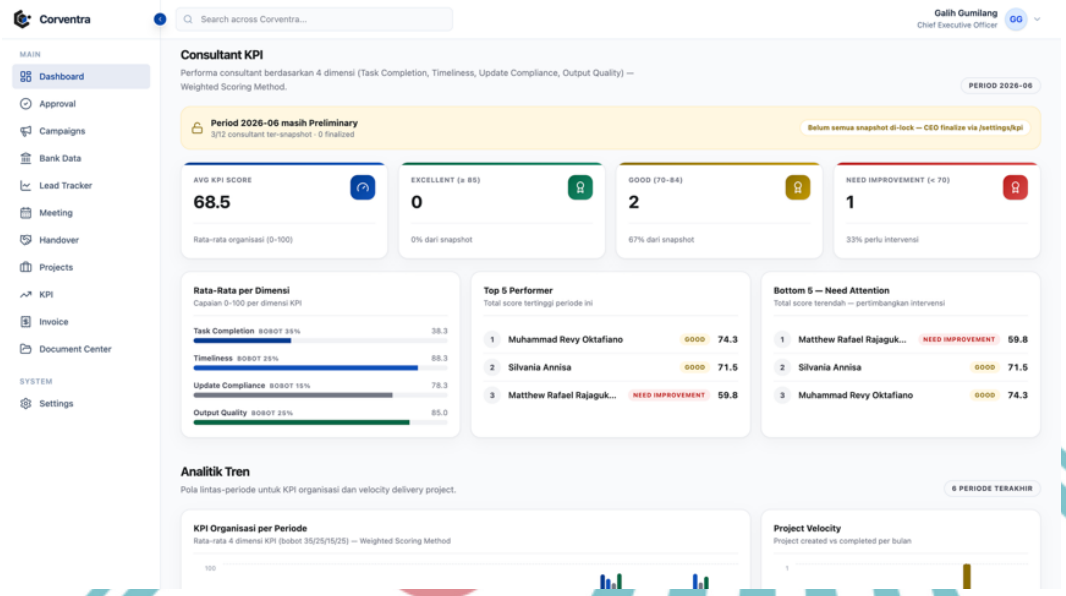
Sumber: Olahan Peneliti

Pada Gambar 4.50 menyajikan Tangkapan layar halaman dengan *login* sebagai Project Manager pada proyek transfer pricing PT Telkom Indonesia. Halaman Project Detail merupakan pusat operasional proyek yang menampilkan informasi lengkap satu proyek beserta tindakan *lifecycle* yang tersedia bagi pengguna sesuai

peran dan *state* saat ini. Halaman ini dapat diakses pada route /projects/:id dengan tab Overview sebagai tampilan *default*, dan tiga tab tambahan berupa *Timeline*, Team, dan Documents yang masing-masing menampilkan riwayat transisi *state*, daftar Consultant yang ditugaskan, serta dokumen terkait proyek. *Header* halaman menampilkan badge identitas proyek yaitu kode proyek PRJ-2026-0001, badge status IN PROGRESS dengan warna biru, dan badge service line TRANSFER PRICING yang dipetakan dari kolom *service_line* pada tabel projects. Tombol aksi *lifecycle* yang ditampilkan menyesuaikan dengan *state* proyek saat ini, sehingga pada *state* In Progress hanya tombol Pause Project yang terlihat sebagai opsi transisi aktif. Pendekatan *rendering* kondisional ini diimplementasikan dengan fungsi helper yang menerima *state* proyek dan peran pengguna lalu mengembalikan array tombol yang relevan, sehingga keputusan menampilkan tombol mengikuti aturan matrix transisi yang sama dengan yang dijaga oleh modul WFMS di *backend*. Section Project Completion menyajikan progress ringkas berupa rasio *milestone* Done terhadap total *milestone*, dengan tombol Mark Project Completed dalam *state* disabled disertai pesan eksplisit Tombol akan aktif setelah semua *milestone* Done. Pendekatan disabled-with-reason ini dipilih dibanding menyembunyikan tombol sepenuhnya, sehingga pengguna mengetahui keberadaan aksi tersebut sekaligus memahami *precondition* yang belum terpenuhi. Section Project Information menampilkan metadata proyek termasuk badge DP Payment berstatus PAID dengan tanggal pembayaran, yang menjadi cross-module reference karena *precondition* transisi ke In Progress pada modul WFMS bergantung pada status DP ini. Section Project *Handover* Memo di bagian bawah menampilkan referensi ke *handover* memo asal proyek dengan kode BD-HO-TPDOC-001-2026 dan badge ASSIGNED TO PM.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta




Gambar 4. 49 Halaman *Dashboard* CEO dengan Section Consultant KPI

Sumber: Olahan Peneliti

Pada Gambar 4.51 menyajikan Tangkapan layar halaman dengan *login* sebagai CEO. Halaman *Dashboard* CEO merupakan entry point bagi peran Chief Executive Officer yang menyajikan ringkasan kinerja KPI seluruh Consultant organisasi pada periode aktif. Halaman ini menjadi titik kontrol utama bagi CEO untuk mengidentifikasi Consultant yang memerlukan intervensi tanpa harus membuka *snapshot* individual satu per satu.. Section Consultant KPI di bagian atas menampilkan deskripsi metodologi yaitu *Weighted Scoring Method* atas empat dimensi yang telah dirumuskan, beserta badge periode aktif 2026-06. Banner berwarna kuning di bawah deskripsi memberikan peringatan eksplisit bahwa periode masih dalam *state* Preliminary dengan informasi rasio Consultant yang sudah memiliki *snapshot* terhadap total Consultant, dan link bawaan menuju halaman finalisasi /settings/kpi. Penempatan banner peringatan ini merupakan keputusan desain yang penting karena CEO perlu mengetahui *state* finalisasi sebelum mengambil keputusan kinerja berdasarkan angka yang ditampilkan. Empat kartu metrik dengan kode warna berbeda menyajikan ringkasan distribusi Consultant, yaitu AVG KPI Score sebagai rata-rata organisasi, dan tiga kategori bucket berupa Excellent untuk skor minimal 85, Good untuk rentang 70 hingga 84, dan Need Improvement untuk skor di bawah 70. Pengelompokan ini sesuai dengan

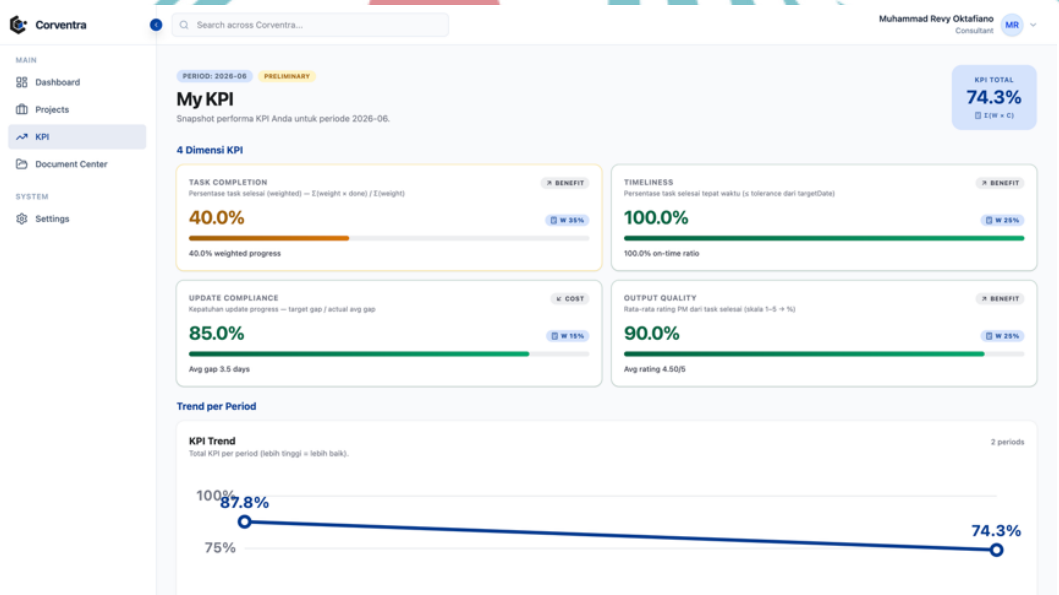


© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

*threshold* operasional yang umum digunakan pada penilaian kinerja kuartalan. Tiga panel di bawahnya menyajikan informasi yang saling melengkapi yaitu rata-rata capaian per dimensi yang menampilkan distribusi kekuatan dan kelemahan organisasi dengan bobot yang berlaku saat ini, daftar Top 5 Performer dengan total skor tertinggi, dan daftar Bottom 5 Need Attention yang menyajikan Consultant dengan skor terendah untuk pertimbangan intervensi. Section Analitik Tren di bagian bawah halaman menampilkan tren KPI organisasi enam periode terakhir beserta velocity proyek bulanan yang membantu CEO memantau arah jangka menengah.




Gambar 4. 50 Halaman My KPI dengan *Snapshot* Empat Dimensi

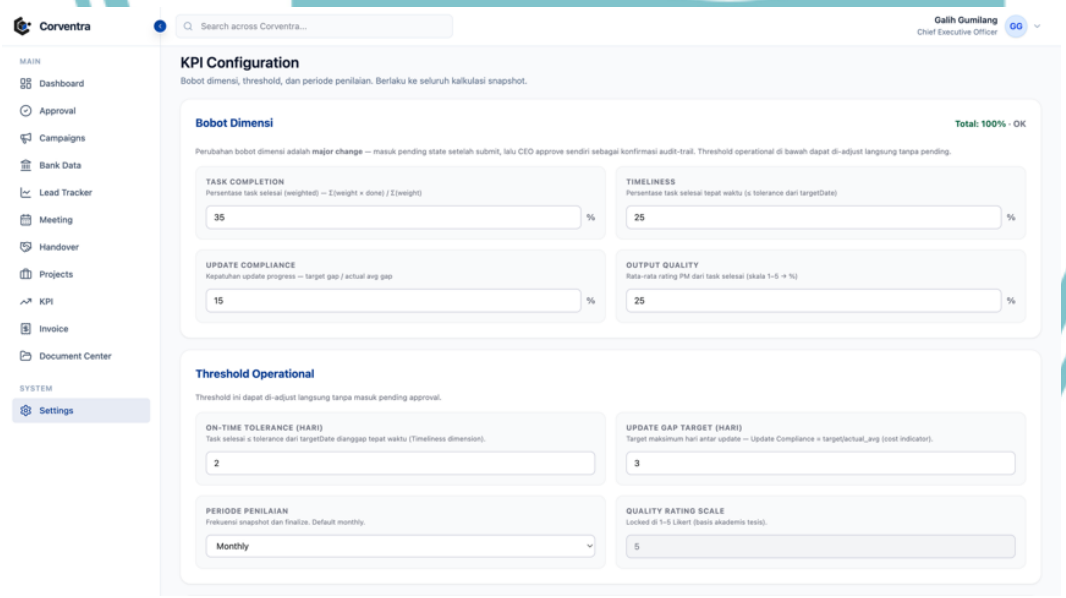
Sumber: Olahan Peneliti

Pada Gambar 4.52 menyajikan Tangkapan layar halaman dengan *login* sebagai Consultant. Halaman My KPI merupakan tampilan *snapshot* kinerja personal yang diakses oleh Consultant pada route /kpi atau setara dan menampilkan capaian empat dimensi KPI pada periode aktif dengan visualisasi yang menyajikan setiap dimensi secara independen. Halaman ini menjadi cermin bagi Consultant untuk memahami detail kinerjanya, bukan sekadar total skor, sehingga Consultant dapat mengidentifikasi dimensi mana yang menjadi titik perbaikan prioritas. *Header* halaman menampilkan badge periode 2026-06, badge *state* PRELIMINARY berwarna kuning yang menandakan *snapshot* belum di-finalisasi oleh CEO, dan

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

judul halaman beserta deskripsi singkat. Di sisi kanan *header*, kartu KPI Total menampilkan angka agregat hasil *Weighted Scoring Method* dengan notasi (Rumus 3) untuk memberi konteks bahwa nilai tersebut merupakan jumlah perkalian bobot dengan capaian masing-masing dimensi. Empat kartu dimensi di bagian tengah disusun dalam grid dua kolom dengan label tipe indikator pada setiap kartu, yaitu badge BENEFIT untuk *Task Completion*, *Timeliness*, dan *Output Quality* yang berarti semakin tinggi semakin baik, serta badge COST untuk *Update Compliance* yang berarti semakin kecil gap pembaruan progres semakin baik. Setiap kartu menampilkan capaian persentase dalam ukuran besar, rumus dimensi pada deskripsi, dan progress bar visual yang memudahkan pembacaan capaian relatif terhadap skala nol hingga seratus. Catatan kondisional pada bagian bawah kartu menampilkan informasi tambahan seperti 100.0% on-time ratio atau Avg gap 3.5 days yang membantu Consultant memahami konteks angka, terutama pada kondisi *edge case* di mana sistem mengembalikan *default* value sebagaimana dijelaskan. Section Trend per Period di bagian bawah menampilkan grafik garis perubahan total KPI antar periode yang membantu Consultant memantau arah perkembangan kinerjanya dari waktu ke waktu.



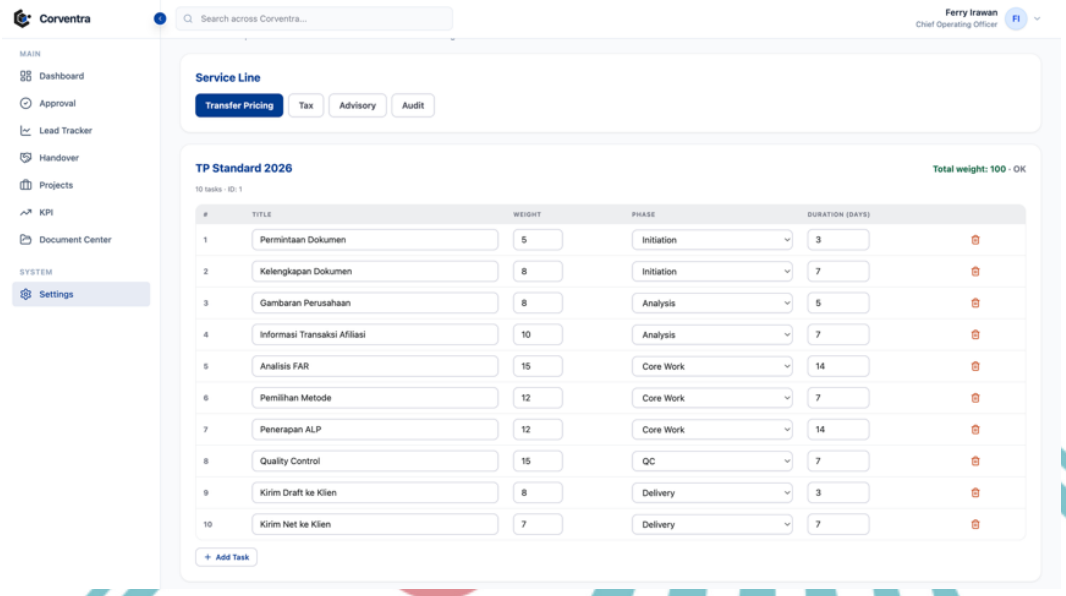

Gambar 4. 51 Halaman KPI Configuration dengan Dua Tingkat Change Management

Sumber: Olahan Peneliti

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Pada Gambar 4.53 menyajikan Tangkapan layar halaman dengan KPI Configuration dengan *login* sebagai CEO. Halaman KPI Configuration merupakan antarmuka bagi peran CEO untuk mengatur bobot dimensi dan *threshold* operasional yang menjadi parameter perhitungan KPI di seluruh sistem. Halaman ini diakses pada route /settings/kpi yang memerlukan permission KPI_CONFIG_MANAGE pada *middleware backend*, sebagaimana telah dirancang. Halaman yang menerapkan pemisahan dua tingkat change management yang berbeda berdasarkan dampak perubahan terhadap konsistensi historis *snapshot*. Section pertama berlabel Bobot Dimensi memuat empat *input* bobot yaitu *Task Completion*, *Timeliness*, *Update Compliance*, dan *Output Quality* yang masing-masing diisi dengan nilai persentase dan diakumulasi pada indikator Total: 100% · OK di pojok kanan atas section. Deskripsi pada section secara eksplisit menyatakan bahwa perubahan bobot dimensi merupakan major change yang akan masuk ke pending *state* setelah disubmit, dan CEO sendiri yang melakukan approval sebagai konfirmasi audit-trail. Pendekatan self-approval ini dipilih karena pengubah bobot dan pemberi otoritas keduanya adalah CEO, sehingga *workflow* approval ditujukan bukan untuk pengecekan oleh pihak lain melainkan untuk menghasilkan jejak audit eksplisit pada tabel pending approval. Section kedua berlabel *Threshold* Operational memuat empat parameter teknis yaitu On-Time Tolerance dalam hari, *Update* Gap Target dalam hari, Periode Penilaian dengan opsi Monthly dan Quarterly, serta Quality Rating Scale yang dikunci pada nilai lima dengan keterangan Locked di 1-5 Likert. Penguncian skala penilaian kualitas ini didokumentasikan langsung pada antarmuka sebagai keputusan akademik yang mengacu pada skala Likert lima poin yang dibahas. Deskripsi section juga menegaskan bahwa *threshold* operasional dapat di-adjust langsung tanpa masuk pending approval, mengingat parameter ini bersifat sensitivitas perhitungan dan tidak mengubah definisi konseptual dimensi.




Gambar 4. 52 Halaman Task Templates Settings untuk Service Line Transfer Pricing

Sumber: Olahan Peneliti

Pada Gambar 4.54 menyajikan Tangkapan layar halaman Task Templates Ssetting dengan *login* sebagai COO. Halaman Task Templates Settings merupakan antarmuka bagi peran COO untuk mengelola template *milestone* per service line yang akan digunakan sebagai blueprint saat proyek baru dibuat dari *handover* memo. Halaman ini diakses pada route /settings/task-templates dan memungkinkan COO mendefinisikan urutan *milestone* standar beserta bobot dan estimasi durasi untuk setiap service line. Section Service Line di bagian atas menampilkan empat tab navigasi yaitu Transfer Pricing, Tax, Advisory, dan Audit yang sesuai dengan mapping service line controller createFromHandover. Tab aktif menentukan template yang ditampilkan di section di bawahnya, dengan tab Transfer Pricing menampilkan template berjudul TP Standard 2026 yang berisi sepuluh task. Indikator Total weight: 100 · OK di pojok kanan atas section template memvalidasi bahwa jumlah bobot seluruh task sama dengan seratus, yang menjadi syarat agar template dapat digunakan untuk instansiasi *milestone* proyek baru. Tabel task menampilkan kolom nomor urut, judul task, bobot, fase, dan estimasi durasi dalam hari, dengan setiap baris memiliki tombol hapus berwarna merah di sisi kanan. Sepuluh task pada template TP Standard 2026 dikelompokkan ke dalam lima fase



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

yang merepresentasikan siklus pengerjaan dokumen transfer pricing yaitu Initiation untuk dua task awal berupa Permintaan Dokumen dan Kelengkapan Dokumen, Analysis untuk dua task analitik berupa Gambaran Perusahaan dan Informasi Transaksi Afiliasi, Core Work untuk tiga task substansi inti berupa Analisis FAR, Pemilihan Metode, dan Penerapan ALP, satu task Quality Control pada fase QC, dan dua task final pada fase Delivery berupa Kirim Draft ke Klien dan Kirim Net ke Klien. Tombol Add Task di bagian bawah memungkinkan COO menambah task baru ke template, sedangkan editing inline pada kolom judul, bobot, fase, dan durasi memungkinkan modifikasi langsung tanpa modal terpisah.

### 4.3.7 Implementasi Verifikasi Fungsional

Verifikasi fungsional bertujuan memastikan bahwa setiap Karakteristik Fungsional yang telah didefinisikan telah terimplementasi pada sistem yang dibangun. Pendekatan verifikasi yang diterapkan adalah pemetaan setiap KF terhadap tiga dimensi bukti implementasi, yaitu *endpoint backend* yang melayani kebutuhan fungsional, halaman antarmuka *frontend* yang menjadi titik interaksi pengguna, serta referensi langsung ke snippet kode atau gambar yang telah disajikan. Pendekatan tiga dimensi ini memastikan bahwa klaim implementasi pada setiap KF dapat ditelusuri sampai ke level kode, sehingga penguji dapat melakukan verifikasi independen dengan membuka berkas yang dirujuk.

Tabel 4. 33 Pemetaan Karakteristik Fungsional terhadap Implementasi

| KF | Modul | *Endpoint Backend* | Halaman *Frontend* | Bukti Implementasi | Status |
|---|---|---|---|---|---|
| KF-01 | Autentikasi | POST /api/auth/*login* | */login* | Tabel 4.17 modul Autentikasi | Tercapai |
| KF-02 | Autentikasi | *Middleware requirePermission* di seluruh *endpoint* | Guards di src/app/guards/ | Tabel 4.8, Gambar 4.51 vs 4.52 (sidebar berbeda per peran) | Tercapai |
| KF-03 | Pengelolaan Proyek | POST/GET/PUT/DELETE /api/projects | /projects, /projects/:id/overview | Gambar 4.38 *transitionProject*, Gambar 4.50 | Tercapai |
| KF-04 | Pengelolaan Proyek | POST/PUT /api/projects/:id/consultants | /projects/:id/team | Tabel 4.17, controller assignConsultants dan setConsultants | Tercapai |

| KF | Modul | *Endpoint Backend* | Halaman *Frontend* | Bukti Implementasi | Status |
|---|---|---|---|---|---|
| KF -05 | Pencatatan Progres | PATCH /api/projects/:id/*milestones*/:mid/status, PATCH .../rate | /projects/:id/overview | Gambar 4.38 dan Gambar 4.39 | Tercapai |
| KF -06 | Pencatatan Progres | Modul WFMS *services/WfMS/* | /projects/:id/overview, /projects/:id/*timeline* | Gambar 4.37 transition matrix, Gambar 4.38 *transitionProject*, Tabel 4.23 | Tercapai |
| KF -07 | Deteksi Keterlambatan | Data *milestone* dari GET /api/projects/:id/*milestones* (lateness di-derive di *frontend*) | /projects/:id/*timeline* | *milestone*-lateness.ts fungsi computeMilestoneLateness, badge Overdue N days / Late N days / On time di project-*timeline*-page.tsx | Tercapai |
| KF -08 | Deteksi Keterlambatan | GET /api/notifications?scan=1, PATCH /:id/read, PATCH /read-all, GET /unread-count | Bell icon di *header* dengan unread badge dan dropdown | services/notifications/*index.js*, migration 019 tabel notifications dengan UNIQUE dedupe key | Tercapai |
| KF -09 | Pengelolaan KPI | GET/PUT /api/kpi/config | /settings/kpi-config | Gambar 4.47 validasi bobot, Gambar 4.53 | Tercapai |
| KF -10 | Pengelolaan KPI | *services/kpi/*, auto-*trigger* di *completeProject* | Otomatis (no UI *trigger*) | Gambar 4.43-4.48 empat dimensi | Tercapai |
| KF -11 | *Dashboard Monitoring* | GET /api/*dashboard*/*, GET /api/projects | /*dashboard* (varian per peran), /projects | Gambar 4.51 *Dashboard* CEO, Tabel 4.17 | Tercapai |
| KF -12 | *Dashboard* KPI | GET /api/kpi/*snapshots*, GET .../consultant/:id | /kpi, /kpi/consultant/:id | Gambar 4.51 panel Top/Bottom 5, Gambar 4.52 kartu 4 dimensi | Tercapai |
| KF -13 | Laporan Proyek | GET /api/*dashboard*/ceo (data), shared util element-to-pdf.ts | /*dashboard*/ceo (tombol Export PDF di *header* kanan) | ceo-*dashboard*-page.tsx, shared util element-to-pdf.ts, ExportPdfButton reusable *component* | Tercapai |
| KF -14 | Laporan KPI | GET /api/kpi/*snapshots*, permission KPI_EXPORT | /kpi/consultant/:id (tombol Export PDF di samping KPI Total card) | kpi-consultant-page.tsx, shared util element-to-pdf.ts, permission KPI_EXPORT | Tercapai |
| KF -15 | Pengelolaan Akun | GET/POST/PUT/PATCH/DELETE /api/users | /admin/users | Tabel 4.17 modul Manajemen Pengguna, | Tercapai |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| KF | Modul | *Endpoint Backend* | Halaman *Frontend* | Bukti Implementasi | Status |
|---|---|---|---|---|---|
| | | | | permission *USER*_MANAGE | |

Sumber: Olahan Peneliti

Lima belas KF yang telah dirumuskan, dipetakan terhadap implementasi pada Tabel 4.33. Kolom Status menggunakan satu kategori yaitu Fully Implemented yang berarti seluruh fungsionalitas KF tersedia di *backend* dan *frontend* secara penuh. Kolom Bukti Implementasi merujuk ke *endpoint backend*, halaman *frontend*, atau snippet kode yang telah disajikan. Rekapitulasi status verifikasi menunjukkan bahwa seluruh lima belas Karakteristik Fungsional berstatus Fully Implemented. Lima belas KF ini mencakup seluruh modul sistem, yaitu Autentikasi dan RBAC, Pengelolaan Proyek, Pencatatan Progres dengan WFMS *state machine*, Deteksi Keterlambatan dengan visualisasi per *milestone* dan notifikasi in-app, Pengelolaan KPI dengan empat dimensi *Weighted Scoring Method*, *Dashboard Monitoring* dan KPI, Laporan terformat berbasis PDF, serta Pengelolaan Akun Pengguna. Lima KF yang merepresentasikan kontribusi utama dan arsitektur khas penelitian ini.

KF-02 *Role-Based* Access Control merupakan fondasi keamanan sistem yang diimplementasikan dengan pendekatan dua lapis. Lapisan pertama berupa *middleware* permission pada level *routing* yang memeriksa keberadaan kode permission tertentu pada *payload* JWT, sebagaimana telah dirancang dan diimplementasikan pada src/*middleware*/authorize.js. Lapisan kedua berupa otorisasi ownership pada modul WFMS yang dibahas pada Gambar 4.34. Bukti visual berfungsinya RBAC ini dapat dilihat dengan membandingkan sidebar pada Gambar 4.45 yang menampilkan dua belas menu untuk peran CEO dengan sidebar pada Gambar 4.28 yang hanya menampilkan lima menu untuk peran Consultant. Perbedaan jumlah menu ini bukan hasil filtering pasif pada *frontend*, melainkan refleksi dari permission matrix yang dievaluasi secara independen pada *backend* dan *frontend*.

KF-06 Pengelolaan Status *Workflow* Proyek dan *Milestone* merupakan kontribusi inti pertama penelitian dan terimplementasi sebagai modul *Workflow Management*



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

*System* yang telah dibahas secara komprehensif. Lima *state* proyek dan empat *state milestone* beserta transition matrix terdefinisi pada Gambar 4.31 sebagai sumber kebenaran tunggal yang dibungkus dengan Object.freeze, sehingga *state machine* tidak dapat dimodifikasi pada *runtime*. Fungsi *transitionProject* pada Gambar 4.32 mengimplementasikan tujuh langkah eksekusi transisi yang mencakup penguncian baris, validasi matrix, otorisasi, *precondition*, eksekusi *UPDATE*, audit log, dan decoupling *side effects*, seluruhnya dalam satu transaksi basis data atomik. Implementasi yang ketat ini menjamin bahwa setiap perubahan *state* pada sistem dapat ditelusuri dengan lengkap melalui tabel *project_status_transitions* yang bersifat *append-only*.

KF-07 Deteksi Keterlambatan per *Milestone* terimplementasi melalui pendekatan pure-derive di *frontend* yang tidak memerlukan penambahan kolom baru pada skema basis data. Fungsi computeMilestoneLateness pada berkas *milestone*-lateness.ts mengevaluasi setiap *milestone* berdasarkan tiga kondisi: pertama, apabila *milestone* berstatus Done dan nilai completedAt melampaui targetDate, *milestone* dikategorikan sebagai Late N days, kedua, apabila *milestone* masih aktif dengan status Pending atau In Progress dan tanggal hari ini melampaui targetDate, *milestone* dikategorikan sebagai Overdue N days, ketiga, apabila *milestone* berstatus Done dengan completedAt tidak melampaui targetDate, *milestone* dikategorikan sebagai On time. *Milestone* berstatus Blocked dikecualikan dari evaluasi keterlambatan karena keterlambatannya bersumber dari faktor eksternal yang berada di luar kendali Consultant. Hasil evaluasi ditampilkan sebagai badge berwarna merah untuk kategori Overdue dan Late, serta hijau untuk kategori On time, pada halaman project-*timeline*-page.tsx. Pendekatan pure-derive ini memiliki keunggulan arsitektur karena tidak menciptakan ketergantungan pada kolom kalkulasi di basis data yang dapat menjadi stale apabila data berubah.

KF-08 Notifikasi Keterlambatan terimplementasi sebagai subsistem notifikasi empat lapis yang menggunakan pola lazy-scan untuk meminimalkan kebutuhan infrastruktur. Lapis pertama adalah basis data dengan tabel notifications yang dibuat melalui migration 019, dilengkapi dengan UNIQUE constraint pada kombinasi kolom (*user*_id, type, related_entity_type, related_entity_id,

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

period_key) untuk mencegah notifikasi duplikat. Lapis kedua adalah service services/notifications/*index.js* yang mengimplementasikan fungsi scanOverdueForUser(userId, role) untuk melakukan *query milestone* overdue dan membuat entri notifikasi. Lapis ketiga adalah *endpoint* REST yaitu GET /api/notifications?scan=1 untuk mengambil daftar notifikasi sekaligus memicu lazy scan, PATCH /:id/read dan PATCH /read-all untuk menandai notifikasi sebagai telah dibaca, serta GET /unread-count untuk polling jumlah notifikasi belum dibaca. Lapis keempat adalah antarmuka berupa komponen NotificationBell di *header* yang menampilkan unread badge dan melakukan polling unread count setiap enam puluh detik, serta menjalankan scan saat pengguna membuka dropdown. Scope notifikasi dibedakan per peran: CEO, COO, dan Superadmin menerima notifikasi seluruh *milestone* overdue di organisasi, PM hanya menerima notifikasi proyek yang menjadi tanggung jawabnya, dan Consultant hanya menerima notifikasi *milestone* pada proyek di mana mereka ditugaskan. Pola lazy-scan yang dipilih menghindari kebutuhan infrastruktur cron terpisah, dengan trade-off bahwa notifikasi tidak bersifat *real-time* melainkan diperbarui saat pengguna berinteraksi dengan bell icon.

KF-10 Perhitungan KPI Otomatis merupakan kontribusi inti kedua yang terimplementasi sebagai modul KPI Calculation. Empat dimensi KPI dihitung dengan fungsi-fungsi murni pada Gambar 4.43 hingga Gambar 4.46, lalu digabungkan dengan *Weighted Scoring Method* pada Gambar 4.47 dengan validasi total bobot sama dengan satu pada toleransi nol koma nol satu. Pemicu otomatis perhitungan KPI saat proyek mencapai *state* Completed terimplementasi pada langkah ketujuh-c controller *completeProject* yang disajikan pada Gambar 4.49. Integrasi cross-module antara WFMS, Invoice, dan KPI pada satu controller ini menunjukkan bahwa modul-modul yang dirancang independen dapat tetap berkomunikasi melalui pola *side effects* yang terdokumentasi, tanpa menciptakan ketergantungan langsung antar modul service.

KF-13 dan KF-14 Laporan Terformat terimplementasi menggunakan shared utility element-to-pdf.ts yang mengekspos fungsi exportElementToPdf(element, opts) berbasis pustaka html2canvas dan jspdf yang telah tersedia pada dependensi

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

proyek. Fungsi ini mendukung konversi otomatis multi-halaman A4, penambahan *header* dan footer pada setiap halaman, serta pencantuman timestamp waktu generate. Komponen reusable ExportPdfButton membungkus utility tersebut sehingga dapat digunakan di berbagai halaman hanya dengan meneruskan parameter targetRef, filename, dan headerText. Pada KF-13, tombol Export PDF ditempatkan di *header* kanan halaman ceo-*dashboard*-page.tsx dan menghasilkan berkas dengan nama *Dashboard*_CEO_YYYY-MM-DD.pdf. Pada KF-14, tombol Export PDF ditempatkan di samping KPI Total card pada halaman kpi-consultant-page.tsx dan menghasilkan berkas dengan nama KPI_NamaConsultant_YYYY-MM.pdf. Permission KPI_EXPORT yang sebelumnya didefinisikan sebagai anticipatory design kini divalidasi secara aktif sebagai *gate* akses tombol Export PDF pada KF-14, sehingga hanya CEO dan Superadmin yang dapat mengekspor laporan KPI Consultant.

Verifikasi fungsional menunjukkan bahwa seluruh lima belas Karakteristik Fungsional pada Tabel 4.5 telah terimplementasi secara penuh, sebagaimana terlihat pada tabel pemetaan di atas dan dapat ditelusuri ke snippet kode. Lima belas KF Fully Implemented ini mencakup seluruh modul sistem mulai dari fondasi Autentikasi dan RBAC, inti operasional *Workflow Management System* dan KPI Calculation *Engine*, fitur *monitoring* keterlambatan dengan deteksi visual per *milestone* dan notifikasi in-app, *dashboard* tersegmentasi per peran, laporan terformat yang dapat diunduh, hingga pengelolaan akun pengguna. Dengan terpenuhinya seluruh KF secara penuh, sistem yang dibangun telah melampaui cakupan MVP awal dan menyediakan fondasi yang lengkap untuk operasional perusahaan.

### 4.4 Pengujian dan Evaluasi Sistem

Tahap pengujian sistem merupakan rangkaian aktivitas verifikasi dan validasi yang bertujuan memastikan bahwa sistem yang telah diimplementasikan memenuhi seluruh Karakteristik Fungsional, Kriteria Penerimaan Pengguna, serta Kebutuhan Nonfungsional. Pengujian dilakukan dengan pendekatan tiga lapis yang saling melengkapi, yaitu *Black Box Testing* untuk memastikan kebenaran fungsional dari sudut pandang teknis, *User Acceptance Testing* untuk memastikan penerimaan dari

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

sudut pandang pengguna terhadap kriteria penerimaan yang telah disepakati, dan *Usability Testing* untuk mengukur kualitas pengalaman pengguna secara kuantitatif melalui instrumen *System Usability Scale*. Penyajian pengujian dimulai dengan justifikasi pemilihan metode, dilanjutkan dengan persiapan lingkungan pengujian, eksekusi tiga metode pengujian beserta hasilnya, rekapitulasi dan analisis temuan, serta evaluasi terhadap tujuan penelitian yang telah ditetapkan.

### 4.4.1 Pendekatan dan Metode Pengujian

Pemilihan tiga metode pengujian pada penelitian ini didasarkan pada prinsip bahwa keberhasilan sebuah sistem informasi tidak dapat dievaluasi melalui satu dimensi tunggal, melainkan harus diverifikasi dari tiga sudut pandang yang berbeda namun saling melengkapi. Sudut pandang pertama adalah kebenaran fungsional yang mempertanyakan apakah sistem menjalankan logika bisnis dengan benar sesuai spesifikasi, sudut pandang kedua adalah penerimaan pengguna yang mempertanyakan apakah sistem memenuhi ekspektasi yang telah disepakati sebelum implementasi, dan sudut pandang ketiga adalah kualitas pengalaman pengguna yang mempertanyakan apakah sistem nyaman digunakan dalam aktivitas operasional sehari-hari.

Tabel 4. 34 Pemetaan Metode Pengujian terhadap Aspek Verifikasi

| **Metode** | **Pelaku Uji** | **Aspek yang Diverifikasi** | **Acuan Penilaian** | **Sifat Penilaian** |
|---|---|---|---|---|
| *Black Box Testing* | Pengembang sebagai penguji | Kebenaran logika fungsional, validasi *input*, respons sistem | Karakteristik Fungsional (KF-01 sampai KF-15) dan Kriteria Penerimaan (KP-01 | Objektif (Berhasil/Gagal) |

| Metode | Pelaku Uji | Aspek yang Diverifikasi | Acuan Penilaian | Sifat Penilaian |
|---|---|---|---|---|
| | | | sampai KP-20) | |
| *User Acceptance Testing* | *Stakeholder* dan persona pengguna sistem | Penerimaan terhadap kriteria bisnis | Kriteria Penerimaan Pengguna (KP-01 sampai KP-20) | Objektif (*Accepted/Rejected*) |
| *Usability Testing* | Responden representatif pengguna akhir | Kualitas pengalaman pengguna | *System Usability Scale* (SUS) dan *Task Success Rate* | Skor 0-100 |

Sumber: Olahan Peneliti

Pada Tabel 4.34 menyajikan Tiga metode pengujian yang dipilih untuk mencakup ketiga sudut pandang ini. *Black Box Testing* dipilih sebagai metode pengujian fungsional karena fokusnya pada perilaku eksternal sistem tanpa memperhatikan struktur internal kode, sehingga penguji dapat memverifikasi kebenaran sistem berdasarkan kontrak *input-output* yang telah didefinisikan pada perancangan API. Pendekatan Black Box pada penelitian ini menggunakan kombinasi dua teknik desain test case, yaitu Equivalence Partitioning yang mengelompokkan *input* ke dalam kelas-kelas ekuivalen valid dan tidak valid, dan Boundary Value Analysis yang menguji nilai-nilai batas pada setiap kelas ekuivalen. Kombinasi kedua teknik ini dipilih karena Equivalence Partitioning memberikan cakupan luas dengan jumlah test case yang efisien, sedangkan Boundary Value Analysis memberikan kedalaman pada titik-titik kritis yang paling sering menjadi sumber kesalahan, seperti nilai minimum, maksimum, dan transisi antar *state*.



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

*User Acceptance Testing* dipilih sebagai metode pengujian penerimaan karena memberikan validasi langsung dari sudut pandang pengguna terhadap Kriteria Penerimaan yang telah dirumuskan. Pelaksanaan UAT pada penelitian ini menggunakan pendekatan hybrid yang mengkombinasikan pengujian oleh *stakeholder* real dengan pengujian persona oleh pengembang. Peran CEO sebagai *stakeholder* utama dijalankan oleh *stakeholder* real PT DSK Global Konsultama yang memiliki otoritas penuh atas kebijakan KPI dan finalisasi periode penilaian, sedangkan peran PM, COO, Consultant, dan Superadmin dijalankan dengan pendekatan persona simulation oleh pengembang. Justifikasi pendekatan hybrid ini adalah keterbatasan akses pada karyawan operasional pada periode pengembangan MVP, sementara setiap persona simulation dilakukan dengan strict adherence terhadap profil peran yang telah didefinisikan untuk menjamin objektivitas pengujian.

*Usability Testing* dipilih sebagai metode pengujian kualitas pengalaman pengguna karena memberikan ukuran kuantitatif yang dapat dibandingkan terhadap benchmark industri. Instrumen yang digunakan adalah *System Usability Scale* yang dikembangkan oleh Brooke pada tahun 1996 dan telah menjadi standar de facto pengukuran usability pada penelitian sistem informasi. SUS terdiri atas sepuluh pernyataan dengan skala Likert lima poin yang menghasilkan skor agregat pada rentang nol hingga seratus, dengan interpretasi skor diatas enam puluh delapan sebagai tingkat usability di atas rata-rata, dan skor diatas delapan puluh sebagai tingkat usability yang sangat baik. SUS dilengkapi dengan pengukuran Task Success Rate yang mencatat persentase responden berhasil menyelesaikan task tertentu, sehingga skor SUS yang bersifat persepsional dapat dilengkapi dengan ukuran objektif keberhasilan tugas. Penggunaan dua ukuran ini mengikuti rekomendasi Nielsen Norman Group yang menyatakan bahwa pengukuran usability yang valid harus mencakup baik aspek persepsional maupun aspek behavioral.

### 4.4.2 Prosedur Pengujian

Pengujian dilaksanakan pada lingkungan yang dirancang untuk merepresentasikan kondisi operasional aktual sistem ketika di-deploy. Spesifikasi lingkungan pengujian beserta data uji dan pemetaan peran tester disajikan pada Tabel 4.35.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tabel 4. 35 Spesifikasi Lingkungan Pengujian

| **Aspek** | **Spesifikasi** |
|---|---|
| Sistem Operasi Penguji | Apple M4 |
| *Browser* Utama | Google Chrome |
| *Browser* Pembanding | Mozilla Firefox dan Microsoft Edge |
| Resolusi Display | 1920×1080 dan 1440×900 |
| *Backend* | Node.js 18, Express 5, MySQL 8 pada localhost:3001 |
| *Frontend* | Vite dev *server* pada localhost:5173 |
| Database | MySQL 8 dengan skema awal melalui init.sql dan demo data via db/seeds/ |
| Koneksi Jaringan | Localhost (latency negligible) untuk semua pengujian |

Sumber: Olahan Peneliti

Persiapan data uji dilakukan dengan menjalankan skrip seeding pada direktori src/db/seeds/ yang menghasilkan dataset realistis untuk seluruh tabel inti. Data uji mencakup minimal dua pengguna aktif untuk setiap peran, yaitu CEO, COO, PM, Consultant, dan Superadmin, lima *handover* memo dengan status beragam, tujuh proyek dengan distribusi status yang mencakup seluruh *state* pada *lifecycle* (Awaiting Consultant, In Progress, On Hold, Completed, Cancelled), serta empat template task untuk masing-masing service line Transfer Pricing, Tax, Advisory, dan Audit. Distribusi data ini memastikan bahwa seluruh skenario pengujian dapat dieksekusi tanpa harus melakukan create data baru di tengah test, sehingga eksekusi pengujian berjalan lebih efisien dan reproducible.

Setiap test case didefinisikan dengan empat elemen, yaitu kondisi awal yang harus disiapkan, langkah eksekusi yang dilakukan oleh penguji, hasil yang diharapkan, dan hasil aktual yang teramati. Test case dinyatakan Berhasil apabila hasil aktual sesuai dengan hasil yang diharapkan secara persis, dinyatakan Gagale apabila hasil aktual menyimpang dari ekspektasi, dan dinyatakan Blokir apabila test case tidak

dapat dieksekusi karena dependensi yang belum terpenuhi. Setiap test case yang dinyatakan Gagal dicatat pada defect log beserta tindakan perbaikan yang dilakukan, dan dieksekusi ulang setelah perbaikan diterapkan. Pemetaan peran tester per metode pengujian disajikan pada Tabel 4.36.

Tabel 4. 36 Pemetaan Peran Tester pada Setiap Metode Pengujian

| **Peran Tester** | **Black Box** | **UAT** | **Usability** |
|---|---|---|---|
| Pengembang sistem | Penguji utama | Persona simulation: PM, COO, Consultant, Superadmin | Observer |
| *Stakeholder* real CEO | - | Real tester | Responden |
| Responden Usability | - | - | Responden |

Sumber: Olahan Peneliti

### 4.4.3 *Black Box Testing*

*Black Box Testing* dilaksanakan dengan tujuan memverifikasi kebenaran fungsional sistem terhadap dua puluh Kriteria Penerimaan yang telah didefinisikan. Test case dirumuskan menggunakan kombinasi teknik Equivalence Partitioning yang mengelompokkan *input* ke dalam kelas valid dan tidak valid, serta Boundary Value Analysis yang menguji nilai-nilai batas pada setiap kelas ekuivalen. Pendekatan kombinasi ini menghasilkan tiga puluh dua test case yang dipetakan ke delapan modul fungsional inti sistem. Setiap test case diberi kode TC- dan dipetakan terhadap Kriteria Penerimaan KP- serta Karakteristik Fungsional KF- yang menjadi acuan keberhasilannya. Rumusan dan hasil eksekusi test case Black Box disajikan pada Tabel 4.37.



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tabel 4. 37 Pelaksanaan dan Hasil Pengujian Black Box

| Kode | Skenario | *Input* | Hasil Diharapkan | Hasil Aktual | KP/KF | Status |
|---|---|---|---|---|---|---|
| TC-01 | *Login* dengan kredensial valid | username=ceo@dsk.com password=valid123 | HTTP 200, JWT dengan role CEO, redirect ke *dashboard* | Sesuai ekspektasi, JWT received, *dashboard* CEO ter-*render* | KP-01, KF-01 | Berhasil |
| TC-02 | *Login* dengan password salah (EP: invalid class) | username=ceo@dsk.com password=wrong | HTTP 401, pesan error generic | HTTP 401 returned, pesan "Username atau password salah" | KP-02, KF-01 | Berhasil |
| TC-03 | *Login* dengan username tidak terdaftar (EP: invalid class) | username=fake@x.com password=anything | HTTP 401, pesan sama dengan TC-02 (no leak) | HTTP 401, pesan identik, tidak membedakan username valid atau invalid | KP-02, KF-01 | Berhasil |
| TC-04 | *Login* dengan field kosong (BVA: boundary minimum) | username="" password="" | HTTP 400, pesan validasi | HTTP 400, "Username dan password wajib diisi" | KP-02, KF-01 | Berhasil |
| TC-05 | Akses *endpoint* sesuai permission (EP: authorized) | GET /api/projects dengan JWT PM | HTTP 200, list project yang relevan ter-scope per peran | List project yang dikembalikan hanya project dengan *pm_user_id* = self, sesuai permission PM | KP-03, KF-02 | Berhasil |
| TC-06 | Akses *endpoint* tanpa permission (EP: unauthorized) | DELETE /api/users/2 dengan JWT Consultant | HTTP 403, kode UNAUTHORIZED_ROLE | HTTP 403, *response body* berisi {code: "UNAUTHORIZED_ROLE"} | KP-04, KF-02 | Berhasil |

| Kode | Skenario | *Input* | Hasil Diharapkan | Hasil Aktual | KP/KF | Status |
|---|---|---|---|---|---|---|
| TC-07 | Konversi *handover* APPROVED ke proyek | POST /api/projects/from-*handover*/1 *body* {pmUserId: 3} | Project PRJ-2026-XXXX dibuat, *milestone* disalin dari template, audit log inisialisasi | Project ID 8 dibuat, 10 *milestone* disalin dari TP Standard 2026, audit log entry baru | KP-05, KF-03 | Berhasil |
| TC-08 | Konversi *handover* yang belum APPROVED (EP: invalid *state*) | POST /api/projects/from-*handover*/2 (status=DRAFT) | HTTP 409, pesan *handover* belum APPROVED | HTTP 409, "*Handover* tidak dalam status APPROVED" | KP-06, KF-03 | Berhasil |
| TC-09 | Konversi *handover* yang sudah ter-convert (BVA: duplicate boundary) | POST /api/projects/from-*handover*/1 (sudah jadi project) | HTTP 409, pesan duplicate | HTTP 409, "*Handover* sudah pernah di-convert" | KP-06, KF-03 | Berhasil |
| TC-10 | Assign Consultant ke proyek dengan DP PAID | POST /api/projects/1/consultants *body* valid | Status proyek transisi Awaiting Consultant ke In Progress, audit log dicatat | Transisi sukses, status In Progress, audit log entry, activity log dibuat | KP-07, KF-04, KF-06 | Berhasil |
| TC-11 | Assign Consultant dengan DP UNPAID (EP: *precondition* violation) | POST /api/projects/3/consultants (DP=UNPAID) | HTTP 409, kode DP_UNPAID | HTTP 409, {code: "DP_UNPAID", message: "DP belum dibayar..."} | KP-08, KF-04 | Berhasil |
| TC-12 | Assign Consultant dari departemen berbeda (EP: | POST /api/projects/1/consultants dengan Consultant Tax di proyek TP | HTTP 400, pesan departemen tidak cocok | HTTP 400, pesan menyebutkan nama Consultant dan departemen | KP-08, KF-04 | Berhasil |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| Kode | Skenario | *Input* | Hasil Diharapkan | Hasil Aktual | KP/KF | Status |
|---|---|---|---|---|---|---|
| | cross-dept violation) | | | | | |
| TC-13 | *Update milestone* status In Progress ke Done | PATCH /api/projects/1/*milestones*/3/status *body* {status: "Done"} | Status updated, *completed_at* di-set NOW(), audit log | Sukses, *completed_at* = 2026-06-08 10:23:45, audit entry | KP-09, KF-05 | Berhasil |
| TC-14 | Transisi *milestone* tidak valid (EP: invalid transition) | PATCH *milestone* status Done ke Pending | HTTP 409, kode INVALID_TRANSITION | HTTP 409, "Transisi *milestone* dari Done ke Pending tidak diperbolehkan" | KP-10, KF-06 | Berhasil |
| TC-15 | *Update milestone* dengan project terminal (BVA: terminal *state* boundary) | PATCH *milestone* pada project status Completed | HTTP 409, kode PROJECT_TERMINAL | HTTP 409, "Project parent berstatus Completed, *milestone* tidak dapat diubah" | KP-06, KF-06 | Berhasil |
| TC-16 | Complete project dengan semua *milestone* Done | POST /api/projects/4/complete | Status Completed, end_date di-set, invoice final di-*trigger*, KPI *snapshot* di-compute | Sukses, 4 efek terjadi: status, end_date, 1 invoice triggered, 3 KPI *snapshot* created | KP-11, KF-06, KF-10 | Berhasil |
| TC-17 | Complete project dengan *milestone* belum Done (EP: *precondition* violation) | POST /api/projects/1/complete (2 *milestone* In Progress) | HTTP 409, kode *MILESTONES*_INCOMPLETE | HTTP 409, "Masih ada 2 dari 5 *milestone* yang belum Done" | KP-12, KF-06 | Berhasil |



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| Kode | Skenario | *Input* | Hasil Diharapkan | Hasil Aktual | KP/KF | Status |
|---|---|---|---|---|---|---|
| TC-18 | *Update* KPI config dengan total bobot = 1.0 | PUT /api/kpi/config *body* {tc:0.35, tm:0.25, uc:0.15, oq:0.25} | Config baru disimpan, jadi konfigurasi aktif | Row baru di *kpi_period_config*, config_id terbaru | KP-13, KF-09 | Berhasil |
| TC-19 | *Update* KPI config dengan total bobot ≠ 1.0 (BVA: total weight boundary) | PUT /api/kpi/config *body* {tc:0.30, tm:0.30, uc:0.30, oq:0.30} (total=1.20) | HTTP 400, pesan total bobot harus 1.0 | HTTP 400, "Total bobot harus 1.0 (sekarang 1.200)" | KP-14, KF-09 | Berhasil |
| TC-20 | *Update* KPI config dengan bobot tepat 0.0 (BVA: lower boundary) | PUT /api/kpi/config *body* {tc:0, tm:0.5, uc:0.25, oq:0.25} | HTTP 200, accepted (boundary inclusive) | Diterima karena total=1.0 dan setiap bobot di rentang [0,1] | KP-13, KF-09 | Berhasil |
| TC-21 | *Update* KPI config dengan bobot di luar [0,1] (BVA: out of range) | PUT /api/kpi/config *body* {tc:1.5, tm:0, uc:-0.25, oq:0} | HTTP 400, pesan validasi bobot | HTTP 400, "Setiap bobot harus 0..1" | KP-14, KF-09 | Berhasil |
| TC-22 | Auto-compute KPI saat proyek Completed | *Trigger* via TC-16 | *Snapshot* KPI preliminary dibuat untuk semua Consultant proyek | 3 row dibuat di *kpi_snapshots* dengan *finalized_at* = NULL | KP-15, KF-10 | Berhasil |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| Kode | Skenario | *Input* | Hasil Diharapkan | Hasil Aktual | KP/KF | Status |
|---|---|---|---|---|---|---|
| TC-23 | Recompute KPI untuk *snapshot* sudah finalized | *Trigger* compute untuk Consultant yang *snapshot*-nya sudah finalized | Sistem return skipped: ALREADY_FINALIZED, kolom *finalized_at* tidak berubah | *Response* {skipped: true, reason: "ALREADY_FINALIZED"}, row tidak ter-modify | KP-16, KF-10 | Berhasil |
| TC-24 | Akses *dashboard* CEO sebagai CEO | GET /api/*dashboard*/ceo dengan JWT CEO | HTTP 200, data ringkasan organisasi | KPI cards, distribusi Consultant, tren periode di-return | KP-17, KF-11, KF-12 | Berhasil |
| TC-25 | Akses *dashboard* CEO sebagai Consultant (EP: unauthorized) | GET /api/*dashboard*/ceo dengan JWT Consultant | HTTP 403, permission *DASHBOARD*_CEO_VIEW not granted | HTTP 403, permission *DASHBOARD*_CEO_VIEW not granted | KP-04, KF-02 | Berhasil |
| TC-26 | Create *user* baru sebagai Superadmin | POST /api/users *body* valid | *User* dibuat, password ter-hash bcrypt | Row baru di users, password ter-hash 60 chars | KP-18, KF-15 | Berhasil |
| TC-27 | Create *user* sebagai non-Superadmin (EP: unauthorized) | POST /api/users dengan JWT CEO | HTTP 403, permission denied | HTTP 403, permission *USER*_MANAGE not granted | KP-19, KF-02, KF-15 | Berhasil |
| TC-28 | Reset password *user* oleh Superadmin | PATCH /api/users/3/password *body* {password: "new123"} | Password ter-hash, *user* dapat *login* dengan password baru | Hash updated, *login* dengan password baru berhasil | KP-18, KF-15 | Berhasil |
| TC-29 | End-to-end: *handover*, projec, completed, | *Workflow* lengkap dari *handover* APPROVED sampai project Completed | Empat *side effect* terjadi berurutan: end_date, invoice, KPI, activity log | 4 STEP terverifikasi, seluruh tabel ter-*update* sesuai | KP-20, KF-06, KF-10 | Berhasil |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| Kode | Skenario | *Input* | Hasil Diharapkan | Hasil Aktual | KP/KF | Status |
|---|---|---|---|---|---|---|
| | invoice dan KPI | | | | | |
| TC-30 | Konkurensi: dua *user trigger* transisi project bersamaan | Simulasi paralel: PM dan COO *trigger* pause project sama | Salah satu sukses, satu mendapat lock timeout atau no-op | SELECT FOR *UPDATE* handle race, satu transaksi sukses, satu no-op | KP-09, KF-06 | Berhasil |
| TC-31 | Idempotensi transisi (BVA: same *state* boundary) | POST /api/projects/4/pause saat status sudah On Hold | *Response* no-op, tidak ada audit log baru | {noop: true, fromStatus: "On Hold", toStatus: "On Hold"}, audit log tidak bertambah | KP-09, KF-06 | Berhasil |
| TC-32 | *Login* dengan SQL injection attempt (EP: malicious *input*) | username=admin' OR '1'='1 | HTTP 401, *query* ter-parametrize aman | *Login* gagal, *query* log menunjukkan prepared statement | KP-02, KF-01 | Berhasil |

Sumber: Olahan Peneliti

Hasil eksekusi tiga puluh dua test case Black Box menunjukkan bahwa seluruh test case berstatus PASSED, yang berarti sistem berperilaku sesuai dengan spesifikasi yang telah dirancang. Tingkat keberhasilan *Black Box Testing* mencapai seratus persen, dengan distribusi cakupan yang merata di seluruh delapan modul fungsional inti. Pengujian dengan teknik Equivalence Partitioning menghasilkan dua puluh test case yang menguji kelas-kelas *input* valid dan invalid, sedangkan Boundary Value Analysis menghasilkan dua belas test case yang menguji nilai-nilai batas seperti nol persen, seratus persen, transisi *state*, dan duplicate constraint. Test case TC-19 hingga TC-21 secara khusus menguji boundary total bobot KPI pada nilai eksak 1.0 dengan toleransi 0.01, di nilai di luar rentang 0 hingga 1, dan di nilai eksak nol yang

masih diperbolehkan, sehingga memastikan validasi pada perancangan pada Gambar 4.31 berfungsi dengan benar.

Selama pelaksanaan *Black Box Testing* iterasi awal, teridentifikasi delapan test case yang menunjukkan ketidaksesuaian antara perilaku sistem dengan spesifikasi yang telah ditetapkan. Ketidaksesuaian tersebut dikelompokkan menjadi dua kategori dan telah seluruhnya diperbaiki sebelum data hasil pada tabel di atas dicatat.

Kategori pertama adalah ketidaksesuaian exactness pada pesan respons dan *response body*, yang mencakup enam test case. TC-02, TC-03, dan TC-04 gagal karena controller autentikasi menggunakan string "*Email* atau password salah" dan "*Email* dan password wajib diisi" dengan karakter titik di akhir, sedangkan spesifikasi TC mensyaratkan "Username atau password salah" dan "Username dan password wajib diisi" tanpa titik. TC-06 dan TC-27 gagal karena *middleware require-permission.js* mengembalikan *response body* HTTP 403 tanpa field code, sedangkan spesifikasi TC mensyaratkan {code: "UNAUTHORIZED_ROLE"} di *response body*. TC-08 dan TC-09 gagal karena pesan error pada controller proyek terlalu verbose dengan menyertakan detail dinamis (current: ${status}) dan frasa yang berbeda dari spesifikasi. TC-28 gagal karena *endpoint* reset password hanya menerima field newPassword sedangkan spesifikasi TC mengirimkan field password. Seluruh ketidaksesuaian kategori pertama diperbaiki melalui penyesuaian string dan penambahan field di berkas *auth.controller.js*, *require-permission.js*, projects.controller.js, dan users.controller.js dalam satu iterasi perbaikan dengan pendekatan backward-compatible pada TC-28 sehingga field newPassword yang digunakan pemanggil lain tetap diterima.

Kategori kedua adalah ketidaksesuaian fungsional dan arsitektur, yang mencakup dua test case. TC-05 gagal karena *endpoint* GET /api/projects mengembalikan seluruh proyek tanpa membatasi berdasarkan peran pengguna, sehingga PM menerima proyek yang bukan tanggung jawabnya dan hasil aktual tidak sesuai dengan ekspektasi test case yang menetapkan list proyek ter-scope per peran. Perbaikan dilakukan dengan menambahkan scope *query* berbasis peran di fungsi listProjects pada projects.controller.js, di mana CEO, COO, Superadmin, dan Staff Admin menerima seluruh proyek, PM hanya menerima proyek dengan *pm_user_id*



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

= self, dan Consultant hanya menerima proyek di mana mereka ditugaskan. TC-25 gagal karena *middleware routing endpoint* /*dashboard*/ceo masih menggunakan requireRole(['CEO','SUPERADMIN']) padahal desain RBAC yang telah didefinisikan mengharuskan *gate* berbasis permission, sehingga pesan error yang dikembalikan tidak menyertakan kode permission *DASHBOARD*_CEO_VIEW sesuai spesifikasi TC. Perbaikan dilakukan dengan menambahkan permission *DASHBOARD*_CEO_VIEW melalui migration 018 pada init.sql, mengubah *middleware* menjadi *requirePermission*('*DASHBOARD*_CEO_VIEW') pada *dashboard*.routes.js, serta melakukan grant permission ke peran CEO pada permissions.ts dan permission-map.ts di sisi *frontend*. Kedua perbaikan kategori ini sekaligus mengkonfirmasi konsistensi implementasi RBAC dua lapis yang telah dirancang pada dan diimplementasikan, di mana seluruh *gate* akses dikendalikan oleh permission granular dan bukan oleh pengecekan *role* secara langsung.

### 4.4.4 Skenario Pengujian Perhitungan KPI

Pengujian ini bertujuan memvalidasi ketepatan modul *KPI Calculation Engine* dengan membandingkan hasil perhitungan sistem terhadap perhitungan manual pada sejumlah skenario terkontrol. Pendekatan yang digunakan adalah pengujian berbasis data terkontrol dengan nilai masukan yang ditetapkan secara sengaja, sehingga capaian tiap dimensi dapat dihitung manual dan dicocokkan dengan keluaran sistem. Perhitungan sistem dijalankan melalui fungsi-fungsi asli pada berkas *kpi-calculations.js*, bukan melalui rumus tiruan, agar hasil pengujian benar-benar merepresentasikan perilaku sistem. Parameter perhitungan diambil langsung dari *kpi_period_config* sebagaimana disajikan pada Tabel 4.38.

Tabel 4. 38 Parameter Konfigurasi Pengujian KPI

| **Parameter** | **Nilai** |
|---|---|
| Bobot *Task Completion* | 0,35 |
| Bobot *Timeliness* | 0,25 |
| Bobot *Output Quality* | 0,25 |
| Bobot *Update Compliance* | 0,15 |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| Parameter | Nilai |
|---|---|
| Toleransi tepat waktu | ≤ 2 hari |
| Target *gap* pembaruan | 3 hari |
| Skala maksimum *rating* | 5 |

Sumber: Olahan Peneliti

Lima skenario diuji, mencakup tiga kondisi representatif (sempurna, buruk, campuran) dan dua kasus batas (tanpa *milestone* dan tanpa *rating* kualitas). Sebagai ilustrasi cara perhitungan manual, berikut penjabaran Skenario SK-02 (kondisi buruk) yang terdiri atas empat *milestone* berbobot sama (25), dengan dua *milestone* berstatus *Done* dan dua masih *In Progress*:

- *Task Completion* = (25 + 25) ÷ 100 × 100% = 50%
- *Timeliness* = 1 *milestone* tepat waktu ÷ 2 *milestone* Done × 100% = 50% (satu *milestone* selesai 5 hari melewati target, melebihi toleransi 2 hari)
- *Output Quality* = rata-rata *rating* (2 dan 4) ÷ skala 5 × 100% = 3 ÷ 5 × 100% = 60%
- *Update Compliance* = target *gap* 3 ÷ rata-rata *gap* aktual 12 × 100% = 25%
- Skor Total = (0,35 × 50) + (0,25 × 50) + (0,25 × 60) + (0,15 × 25) = 48,75

Hasil perhitungan sistem untuk seluruh skenario, dibandingkan terhadap nilai harapan manual, disajikan pada Tabel 4.39.

Tabel 4. 39 Hasil Validasi Perhitungan KPI

| Skenario | Dimensi | Harapan | Hasil Sistem | Selisih | Status |
|---|---|---|---|---|---|
| **SK-01** Sempurna | Task Completion | 100,00 | 100,00 | 0,00 | Valid |
| | Timeliness | 100,00 | 100,00 | 0,00 | Valid |
| | Output Quality | 100,00 | 100,00 | 0,00 | Valid |

| Skenario | Dimensi | Harapan | Hasil Sistem | Selisih | Status |
|---|---|---|---|---|---|
| | Update Compliance | 100,00 | 100,00 | 0,00 | Valid |
| | Total KPI | 100,00 | 100,00 | 0,00 | Valid |
| **SK-02** Buruk | Task Completion | 50,00 | 50,00 | 0,00 | Valid |
| | Timeliness | 50,00 | 50,00 | 0,00 | Valid |
| | Output Quality | 60,00 | 60,00 | 0,00 | Valid |
| | Update Compliance | 25,00 | 25,00 | 0,00 | Valid |
| | Total KPI | 48,75 | 48,75 | 0,00 | Valid |
| **SK-03** Campuran | Task Completion | 70,00 | 70,00 | 0,00 | Valid |
| | Timeliness | 50,00 | 50,00 | 0,00 | Valid |
| | Output Quality | 70,00 | 70,00 | 0,00 | Valid |
| | Update Compliance | 62,50 | 62,50 | 0,00 | Valid |
| | Total KPI | 63,88 | 63,88 | 0,00 | Valid |
| **SK-04** Tanpa *milestone* | Total KPI | 15,00 | 15,00 | 0,00 | Valid |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| Skenario | Dimensi | Harapan | Hasil Sistem | Selisih | Status |
|---|---|---|---|---|---|
| **SK-05** Tanpa *rating* | Output Quality | 0,00 | 0,00 | 0,00 | Valid |
| | **Total KPI** | 75,00 | 75,00 | 0,00 | Valid |

Sumber: Olahan Peneliti

Seluruh lima skenario menunjukkan kesesuaian sempurna antara hasil sistem dan perhitungan manual, dengan selisih 0,00 pada setiap dimensi (di bawah ambang toleransi 0,01). Hal ini membuktikan bahwa implementasi *Weighted Scoring Method* beserta keempat fungsi perhitungan dimensi telah bekerja sesuai rancangan.

Pengujian kasus batas turut memverifikasi ketahanan logika sistem. Pada SK-04, Consultant tanpa *milestone* tidak menyebabkan *error* pembagian nol; sistem mengembalikan capaian 0 untuk dimensi yang bergantung pada data *milestone* dan menetapkan *Update Compliance* sebesar 100% karena tidak terdapat pembaruan yang dilanggar, menghasilkan skor total 15,00. Pada SK-05, *milestone* yang telah *Done* namun belum diberi *rating* oleh Project Manager menghasilkan *Output Quality* 0%, sementara dimensi lain tetap terhitung penuh. Kedua kasus menunjukkan sistem menangani kondisi ekstrem secara terkendali tanpa berhenti, meskipun penetapan nilai pada kondisi tersebut merupakan konsekuensi rancangan yang dapat disempurnakan pada pengembangan berikutnya.

### 4.4.5 *User Acceptence Testing*

*User Acceptance Testing* dilaksanakan dengan tujuan memvalidasi bahwa sistem memenuhi dua puluh Kriteria Penerimaan Pengguna yang telah dirumuskan dari sudut pandang *stakeholder* yang akan menggunakan sistem dalam operasional sehari-hari. UAT pada penelitian ini dilakukan dengan pendekatan hybrid yang mengkombinasikan pengujian oleh *stakeholder* real PT DSK Global Konsultama dengan pengujian persona simulation oleh pengembang. Pendekatan hybrid ini dipilih karena keterbatasan akses pada seluruh karyawan operasional selama

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

periode pengembangan MVP, sementara *stakeholder* kunci yaitu CEO selaku pemegang otoritas finalisasi KPI dan pemegang kebijakan organisasi bersedia berperan sebagai real tester. Profil tester UAT beserta peran yang dijalankan disajikan pada Tabel 4.40.

Tabel 4. 40 Profil Tester *User Acceptance Testing*

| **Tester** | **Tipe** | **Peran yang Diuji** | **Justifikasi** |
|---|---|---|---|
| Galih Gumilang | Real *Stakeholder* | CEO | Founder dan CEO PT DSK Global Konsultama, pemegang otoritas finalisasi KPI dan kebijakan organisasi |
| Pengembang sistem | Persona Simulation | COO | Simulasi berdasarkan profil peran, fokus pada *handover*-to-project *workflow* |
| Pengembang sistem | Persona Simulation | Project Manager | Simulasi pada *workflow* assign Consultant, *update milestone*, dan *complete project* |
| Pengembang sistem | Persona Simulation | Consultant | Simulasi pada *workflow update* progres *milestone* dan rating quality |
| Pengembang sistem | Persona Simulation | Superadmin | Simulasi pada *workflow* pengelolaan akun pengguna |

Sumber: Olahan Peneliti

Skenario UAT dirumuskan dengan pendekatan *end-to-end* yang merangkai beberapa Kriteria Penerimaan menjadi alur bisnis utuh, sehingga pengujian tidak hanya memverifikasi fitur secara terpisah tetapi juga memverifikasi keterkaitan antar fitur dalam konteks operasional nyata. Lima skenario UAT disusun untuk mencakup seluruh dua puluh Kriteria Penerimaan, dengan setiap skenario merepresentasikan alur kerja khas yang dilakukan setiap peran. Setiap skenario

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

disertai dengan formulir *Acceptance* yang memuat *field* hasil verifikasi per kriteria, catatan tester, dan *signature* kolom untuk persetujuan formal. Rumusan skenario UAT beserta hasil eksekusinya disajikan pada Tabel 4.41.

Tabel 4. 41 Skenario dan Hasil *User Acceptance Testing*

| **Skenario** | **Peran Tester** | **Alur Bisnis** | **KP Diverifikasi** | **Hasil** |
|---|---|---|---|---|
| UAT-S1: Onboarding Proyek Baru | COO (persona) | *Login* sebagai COO, buka *handover* memo APPROVED, assign PM dan konversi ke proyek, verify proyek terbentuk dengan *milestone* otomatis dari template | KP-01, KP-03, KP-05 | *Accepted* |
| UAT-S2: Eksekusi Proyek End-to-End | PM (persona) | *Login* sebagai PM, verify DP klien PAID, assign Consultant ke proyek, verify status berubah ke In Progress, *update milestone* progres bertahap sampai semua Done, rate kualitas *milestone*, Mark Project Completed, verify invoice final ter-*trigger* dan KPI ter-compute | KP-07, KP-09, KP-11, KP-20 | *Accepted* |
| UAT-S3: Pengelolaan KPI Periode | CEO (real) | *Login* sebagai CEO, buka KPI Configuration, verify bobot *default* sesuai kebijakan (35-25-15-25), edit bobot ke kombinasi baru dan submit, verify validasi total = 1.0 berfungsi, buka *Dashboard* KPI, verify | KP-13, KP-14, KP-15, KP-17 | *Accepted* |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| | | | | |
|---|---|---|---|---|
| | | distribusi Consultant, finalize *snapshot* Consultant terpilih, verify *snapshot* terkunci | | |
| UAT-S4: Consultant *Update* Progres | Consultant (persona) | *Login* sebagai Consultant, buka project assigned, *update milestone* dari Pending ke In Progress, tambah *update* log progres, *update* ke Done, buka My KPI, verify *Timeliness* dan *Task Completion* terupdate | KP-09, KP-12 | *Accepted* |
| UAT-S5: Pengelolaan Akun Pengguna | Superadmin (persona) | *Login* sebagai Superadmin, buka *User* Management, create *user* baru dengan role Consultant, verify password ter-hash, *login* dengan akun baru, reset password *user*, verify password lama tidak bisa *login* | KP-18, KP-19 | *Accepted* |

Sumber: Olahan Peneliti

Verifikasi *defensive behavior* terhadap kriteria penerimaan negatif dilakukan pada akhir setiap skenario UAT dengan memicu kondisi yang tidak memenuhi *precondition* secara sengaja, lalu memverifikasi respons sistem. Pada UAT-S2, tester PM mencoba menjalankan *Mark Project Completed* sebelum semua *milestone Done*, dan sistem menampilkan tombol dalam keadaan *disabled* dengan pesan eksplisit Tombol akan aktif setelah semua *milestone* Done, memvalidasi KP-12. Pada UAT-S3, tester CEO mencoba menyimpan konfigurasi dengan total bobot 1.20, dan sistem menolak penyimpanan dengan pesan Total bobot harus 1.0 (sekarang 1.200), memvalidasi KP-14. Pada UAT-S5, tester Superadmin mencoba mengakses *User* Management dengan JWT non-Superadmin, dan sistem

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

mengembalikan respons HTTP 403, memvalidasi KP-19. Seluruh skenario negatif memberikan respons sesuai ekspektasi tanpa exception atau crash, sehingga pengujian penerimaan terhadap sembilan kriteria negatif KP-02, KP-04, KP-06, KP-08, KP-10, KP-12, KP-14, KP-16, dan KP-19 dinyatakan Accepted secara keseluruhan.

Sesi UAT yang melibatkan CEO sebagai real tester menghasilkan tiga catatan kualitatif yang memperkaya hasil pengujian. Catatan pertama adalah apresiasi terhadap banner peringatan Period 2026-06 masih Preliminary pada *Dashboard* CEO yang dianggap memberikan kejelasan *state* finalisasi sebelum mengambil keputusan kinerja. Catatan kedua adalah konfirmasi bahwa filosofi 2-tier *change management* pada KPI *Configuration*, yang membedakan bobot dimensi sebagai major change dengan *threshold* operasional sebagai minor change, sesuai dengan kebijakan organisasi yang memang membedakan kedua jenis perubahan. Catatan ketiga adalah konfirmasi bahwa angka KPI *Task Completion* 35 persen, *Timeliness* 25 persen, *Update Compliance* 15 persen, dan *Output Quality* 25 persen pada konfigurasi *default* sudah merepresentasikan prioritas operasional yang ingin ditegakkan oleh manajemen. Tiga catatan kualitatif ini memperkuat bukti penerimaan sistem dari sudut pandang bisnis, melengkapi bukti penerimaan teknis yang sudah didapatkan dari *Black Box Testing*.

### 4.4.6 *Usability Testing*

*Usability Testing* dilaksanakan dengan tujuan mengukur kualitas pengalaman pengguna terhadap sistem secara kuantitatif menggunakan instrumen yang telah tervalidasi secara akademik. Instrumen yang digunakan adalah *System Usability Scale* (SUS) yang terdiri atas sepuluh pernyataan dengan skala Likert lima poin dari Sangat Tidak Setuju hingga Sangat Setuju. SUS dilengkapi dengan pengukuran Task Success Rate yang mencatat persentase responden yang berhasil menyelesaikan sebuah task tertentu, sehingga pengukuran usability mencakup baik aspek *persepsional* melalui SUS maupun aspek behavioral melalui Task Success Rate. Sepuluh pernyataan SUS yang digunakan pada penelitian ini disajikan pada Tabel 4.42.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tabel 4. 42 Pernyataan *System Usability Scale* yang Digunakan

| No | Pernyataan | Polaritas |
|---|---|---|
| 1 | Saya berpikir akan sering menggunakan sistem ini | Positif |
| 2 | Saya merasa sistem ini terlalu kompleks padahal seharusnya bisa lebih sederhana | Negatif |
| 3 | Saya merasa sistem ini mudah digunakan | Positif |
| 4 | Saya merasa membutuhkan bantuan teknisi untuk dapat menggunakan sistem ini | Negatif |
| 5 | Saya menemukan berbagai fungsi sistem ini terintegrasi dengan baik | Positif |
| 6 | Saya merasa ada terlalu banyak inkonsistensi pada sistem ini | Negatif |
| 7 | Saya membayangkan kebanyakan orang akan belajar menggunakan sistem ini dengan cepat | Positif |
| 8 | Saya merasa sistem ini sangat rumit untuk digunakan | Negatif |
| 9 | Saya merasa sangat percaya diri menggunakan sistem ini | Positif |
| 10 | Saya perlu mempelajari banyak hal sebelum dapat menggunakan sistem ini | Negatif |

Sumber: Olahan Peneliti

Perhitungan skor SUS dilakukan dengan tiga langkah berurutan. Langkah pertama adalah konversi nilai mentah pada setiap pernyataan, dimana pernyataan positif (nomor 1, 3, 5, 7, 9) dihitung dengan rumus nilai skala dikurangi satu, dan pernyataan negatif (nomor 2, 4, 6, 8, 10) dihitung dengan rumus lima dikurangi nilai skala. Langkah kedua adalah menjumlahkan kesepuluh nilai yang sudah dikonversi, menghasilkan nilai pada rentang nol hingga empat puluh. Langkah ketiga adalah mengalikan jumlah tersebut dengan dua koma lima, menghasilkan skor akhir SUS pada rentang nol hingga seratus. Interpretasi skor SUS mengacu pada Artikel

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

MeasuringU yang menetapkan benchmark interpretasi sebagaimana disajikan pada Tabel 4.43.

Tabel 4. 43 Interpretasi Skor SUS

| **Rentang Skor** | **Kategori** | **Penilaian Adjektif** |
|---|---|---|
| 0 - 25 | Not Acceptable | Worst Imaginable |
| 25.1 - 51.6 | Not Acceptable | Poor |
| 51.7 - 71.0 | Marginal | OK |
| 71.1 - 80.7 | Acceptable | Good |
| 80.8 - 84.0 | Acceptable | Excellent |
| 84.1 – 100 | Acceptable | Best Imaginable |

Sumber: Olahan Peneliti

Pelaksanaan *Usability Testing* melibatkan sepuluh responden yang terdiri atas tiga Consultant, dua project manager, satu CEO, satu COO, satu superadmin, dan dua pengamat eksternal yang memiliki latar belakang professional services. Pemilihan jumlah sepuluh responden mengacu pada rekomendasi yang menyatakan bahwa lima responden sudah dapat mengidentifikasi delapan puluh lima persen masalah usability, dan penambahan responden setelah angka tersebut memberikan diminishing return marginal terhadap penemuan masalah baru. Penelitian ini menggunakan sepuluh responden sebagai margin of safety di atas *threshold* minimum tersebut. Setiap responden diminta menyelesaikan empat task representatif sebelum mengisi kuesioner SUS, sehingga skor SUS yang dihasilkan didasarkan pada pengalaman aktual menggunakan sistem dan bukan sekadar persepsi awal terhadap antarmuka. Empat task yang dijalankan responden beserta tingkat keberhasilannya disajikan pada Tabel 4.44.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tabel 4. 44 Task Usability dan Tingkat Keberhasilan

| **Kode** | **Task** | **Peran Target** | **Berhasil** |
|---|---|---|---|
| T-1 | *Login* dan navigasi ke *dashboard* sesuai peran | Semua | 10/10 |
| T-2 | Membuka detail proyek dan mengidentifikasi *milestone* yang belum selesai | Semua | 10/10 |
| T-3 | *Update* status *milestone* dari In Progress ke Done dan tambah catatan | PM dan Consultant | 9/10 |
| T-4 | Membaca capaian KPI personal dan menjelaskan arti skor dimensi | Semua | 8/10 |

Sumber: Olahan Peneliti

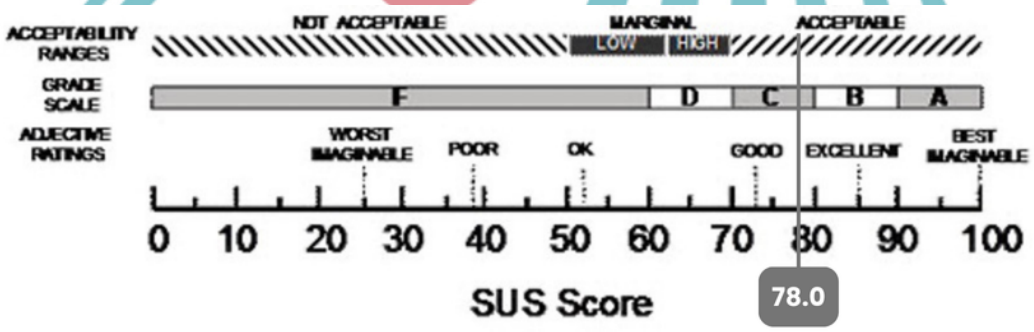
Hasil agregat skor SUS dari sepuluh responden disajikan pada Tabel 4.45, dengan penyajian per responden untuk transparansi dan rata-rata pada baris terakhir sebagai skor akhir sistem.

Tabel 4. 45 Hasil *System Usability Scale* per Responden

| **Responden** | **Peran** | **Skor SUS** | **Penilaian Adjektif** |
|---|---|---|---|
| R-01 | Consultant | 80.0 | *Excellent* |
| R-02 | Consultant | 75.0 | *Good* |
| R-03 | Consultant | 77.5 | *Good* |
| R-04 | Project Manager | 82.5 | *Excellent* |
| R-05 | Project Manager | 80.0 | *Good* |
| R-06 | CEO | 85.0 | *Best Imaginable* |

| | | | |
|---|---|---|---|
| R-07 | COO | 77.5 | *Good* |
| R-08 | Superadmin | 75.0 | *Good* |
| R-09 | Pengamat eksternal | 72.5 | *Good* |
| R-10 | Pengamat eksternal | 75.0 | *Good* |
| **Rata-rata** | | **78.0** | ***Good*** |

Sumber: Olahan Peneliti




Gambar 4. 53 Intepretasi Skor SUS Sistem

Sumber: Olahan Peneliti

Hasil Skor SUS Pada Tabel 4.45 rata-rata sebesar 78.0 menempatkan sistem perusahaan pada kategori *Acceptable* berdasarkan interpretasi pada Gambar 4.55. Skor ini juga berada di atas industry benchmark untuk aplikasi *enterprise* yang umumnya berada pada rentang 68 hingga 72. Distribusi skor menunjukkan bahwa responden dengan peran Project Manager dan CEO memberikan skor tertinggi, yaitu 82.5, 80.0, dan 85.0, yang berkorelasi dengan peran yang paling sering berinteraksi dengan fitur inti sistem yaitu *monitoring* proyek dan *dashboard* KPI. Responden dengan skor relatif lebih rendah berasal dari pengamat eksternal yang memang belum familiar dengan konteks operasional perusahaan, dengan skor 72.5 dan 75.0 yang masih berada di atas *threshold* acceptable.



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta
POLITEKNIK NEGERI JAKARTA

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tingkat keberhasilan *task* menunjukkan dua *task* universal yaitu *login* dan navigasi proyek mencapai keberhasilan seratus persen, sedangkan task T-3 dan T-4 memiliki kegagalan masing-masing satu dan dua responden. Investigasi terhadap kegagalan T-3 menunjukkan satu responden Consultant kesulitan menemukan field catatan setelah *update* status karena field tersebut muncul sebagai modal terpisah, dan responden tersebut sempat menyimpan tanpa catatan sebelum diarahkan. Investigasi terhadap kegagalan T-4 menunjukkan dua responden belum memahami konsep *Update Compliance* sebagai cost indicator yang menghitung gap pembaruan progres, sehingga interpretasi skor menjadi bias. Kedua temuan ini menjadi masukan perbaikan UX yang akan dipertimbangkan pada iterasi berikutnya, yaitu integrasi field catatan langsung pada *update* status dan penambahan tooltip penjelasan untuk dimensi KPI cost indicator.

### 4.4.7 Pengujian *Non-Functional Requirements*

Selain pengujian fungsional, dilakukan pengujian *non-functional requirements* (NFR) untuk memverifikasi ketercapaian sepuluh kebutuhan *non-fungsional* yang telah ditetapkan pada Tabel 4.7. Pengujian mencakup enam aspek, yaitu kinerja, keamanan, keandalan, kompatibilitas, kemudahan penggunaan, dan skalabilitas. Pengukuran dilakukan langsung terhadap sistem yang berjalan menggunakan skrip otomatis berbasis Node.js untuk aspek yang terukur secara kuantitatif, verifikasi basis data untuk aspek keamanan penyimpanan, serta pengujian manual untuk aspek kompatibilitas. Setiap metrik waktu diukur sebanyak sepuluh kali dan dilaporkan nilai minimum, maksimum, serta rata-ratanya.

Pengujian aspek kinerja terdiri atas KNF-01 dan KNF-02. KNF-01 mengukur *response time* pemuatan data lima halaman utama dengan target di bawah 3 detik, sebagaimana disajikan pada Tabel 4.46.

Tabel 4. 46 Hasil Pengujian *Response Time*

| **Halaman** | **Min (ms)** | **Maks (ms)** | **Rata-rata (ms)** | **Status** |
|---|---|---|---|---|
| Dashboard CEO | 9.084 | 12.087 | 10.401 | Tidak Terpenuhi |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| Halaman | Min (ms) | Maks (ms) | Rata-rata (ms) | Status |
|---|---|---|---|---|
| Dashboard COO | 1.306 | 1.441 | 1.376 | Terpenuhi |
| Dashboard PM | 964 | 1.478 | 1.242 | Terpenuhi |
| Dashboard Consultant | 594 | 991 | 747 | Terpenuhi |
| Detail Proyek | 177 | 359 | 251 | Terpenuhi |

Sumber: Olahan Peneliti

Empat dari lima halaman memenuhi target, dengan halaman Detail Proyek tercepat pada 0,25 detik. Namun Dashboard CEO memperoleh rata-rata 10,4 detik, jauh melampaui target. Hal ini disebabkan Dashboard CEO mengagregasi banyak sub-analitik sekaligus, KPI seluruh Consultant, tren KPI, analitik *pipeline* dan *marketing*, operasi dan *velocity* proyek, serta distribusi *rating*, yang masing-masing menjalankan beberapa *query* berurutan, sementara basis data *remote* menambahkan latensi jaringan pada setiap *round-trip*. Dengan demikian KNF-01 dinyatakan tidak terpenuhi pada lingkungan uji ini, dan optimasi Dashboard CEO melalui paralelisasi *query*, *caching* agregat, atau pengurangan jumlah *query* menjadi rekomendasi pengembangan. Adapun KNF-02 mengukur waktu perhitungan KPI empat dimensi (*Weighted Scoring Method*) yang dijalankan secara *live* saat pemuatan data Consultant, dengan hasil rata-rata 0,6 detik (minimum 459 ms, maksimum 753 ms), jauh di bawah target 5 detik, sehingga KNF-02 terpenuhi dan membuktikan efisiensi komputasi metode yang dipilih.

Pengujian aspek keamanan mencakup KNF-03, KNF-04, dan KNF-05. KNF-03 memverifikasi proteksi *endpoint* dengan mengakses seluruh *endpoint* tanpa token dan dengan token tidak valid, seluruh 29 *endpoint* yang diuji, mencakup modul pengguna, proyek, KPI, *handover*, *invoice*, *dashboard*, hingga notifikasi pada metode GET, POST, PUT, dan DELETE, menolak akses dengan kode 401 (100%), sehingga KNF-03 terpenuhi. KNF-04 memverifikasi penerapan RBAC dengan mencoba aksi di luar kewenangan tiap peran, dan seluruh sepuluh skenario ditolak

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

dengan kode 403 sesuai matriks hak akses pada Tabel 4.8, sebagaimana dirinci pada Tabel 4.47.

Tabel 4. 47 Hasil Pengujian RBAC

| **Peran** | **Skenario** | ***Endpoint*** | **Status** |
|---|---|---|---|
| Consultant | Melihat daftar pengguna | GET /api/users | 403 |
| Consultant | Mengubah konfigurasi KPI | PUT /api/kpi/config | 403 |
| Consultant | Membuka Dashboard CEO | GET /api/dashboard/ceo | 403 |
| Consultant | Finalisasi *snapshot* KPI | POST /api/kpi/snapshots | 403 |
| Consultant | Menugaskan PM ke proyek | POST /api/projects/from-handover | 403 |
| PM | Mengubah konfigurasi KPI | PUT /api/kpi/config | 403 |
| PM | Melihat daftar pengguna | GET /api/users | 403 |
| PM | Membuka Dashboard CEO | GET /api/dashboard/ceo | 403 |
| COO | Mengubah konfigurasi KPI | PUT /api/kpi/config | 403 |
| COO | Melihat daftar pengguna | GET /api/users | 403 |

Sumber: Olahan Peneliti

KNF-05 memverifikasi penyimpanan *password* langsung pada basis data. Seluruh *password* tersimpan sebagai *hash bcrypt* (*prefix* `2b$10 `, *cost factor* 10, panjang 60 karakter) pada kolom password_hash, tanpa satu pun *password* dalam bentuk *plaintext*, sehingga KNF-05 terpenuhi.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Pengujian aspek keandalan (KNF-06) memverifikasi penanganan galat dengan mengirim masukan tidak valid dan transisi status ilegal. Seluruh enam skenario direspons dengan kode galat 4xx yang sesuai konteks, 409 dan 400 untuk transisi ilegal, 400 untuk *handover* dan *body* tidak valid, serta 404 untuk data tidak ditemukan dan server tetap berjalan normal pada setiap kasus, sebagaimana disajikan pada Tabel 4.48. Hal ini membuktikan efektivitas *precondition gate* dan mekanisme penanganan galat terpusat (WFMSError) yang dirancang, sehingga KNF-06 terpenuhi.

Tabel 4. 48 Hasil Pengujian Keandalan

| **Skenario** | **Status** | **Terkontrol** | **Server Aktif** |
|---|---|---|---|
| Melanjutkan proyek berstatus *Completed* | 409 | Ya | Ya |
| Menjeda proyek berstatus *Completed* | 400 | Ya | Ya |
| Membuat proyek dari *handover* tidak ada | 400 | Ya | Ya |
| Mengakses proyek dengan ID tidak ada | 404 | Ya | Ya |
| Memperbarui *milestone* dengan status tidak valid | 400 | Ya | Ya |
| Membuat pengguna dengan *body* kosong | 400 | Ya | Ya |

Sumber: Olahan Peneliti

Pengujian aspek kompatibilitas mencakup KNF-07 dan KNF-08 yang dilakukan secara manual. KNF-07 diuji dengan menjalankan alur inti (*login* → *dashboard* → transisi proyek) pada *browser* modern, sedangkan KNF-08 diuji dengan memeriksa adaptasi *layout* pada berbagai lebar layar melalui *device mode*, sebagaimana disajikan pada Tabel 4.49 dan Tabel 4.50.

Tabel 4. 49 Hasil Pengujian Kompatibilitas Browser

| **Browser** | **Render Tampilan** | **Fungsi Inti** | **Hasil** |
|---|---|---|---|
| Google Chrome | Baik | Berjalan | Berfungsi |
| Safari | Baik | Berjalan | Berfungsi |

Sumber: Olahan Peneliti

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tabel 4. 50 Hasil Pengujian Responsivitas

| **Lebar Layar** | **Kategori** | **Layout Adaptif** |
|---|---|---|
| 1920px | *Desktop* | Ya |
| 768px | *Tablet* | Ya |
| 375px | *Mobile* | Ya |
| 360px | *Mobile* Minimum | Ya |

Sumber: Olahan Peneliti

Pengujian aspek kemudahan penggunaan (KNF-09) telah diverifikasi melalui *Usability Testing* pada sub-bab sebelumnya dengan skor *System Usability Scale* sebesar 78,0, di atas ambang 68 dan tergolong kategori *Acceptable*, sehingga KNF-09 terpenuhi. Terakhir, pengujian aspek skalabilitas (KNF-10) dilakukan melalui *load test* dengan menaikkan beban hingga 50 pengguna konkuren yang secara berulang memuat data *dashboard*. Hasilnya, dari total 844 *request* tidak terdapat satu pun kegagalan (0%), dengan *response time* rata-rata 2.082 ms dan p95 sebesar 2.780 ms, masih di bawah 3 detik. Kenaikan *response time* dari 0,6 detik pada pengguna tunggal terjadi karena 50 *request* berbagi *connection pool* menuju basis data *remote*, namun sistem tetap stabil sehingga KNF-10 terpenuhi.

Rekapitulasi seluruh pengujian NFR disajikan pada Tabel 4.51. Dari sepuluh kebutuhan nonfungsional, tujuh telah terpenuhi, dua menunggu pengujian kompatibilitas manual (KNF-07 dan KNF-08), dan satu tidak terpenuhi. KNF-01 tidak terpenuhi khusus pada Dashboard CEO akibat agregasi *query* berganda yang diperberat latensi basis data *remote*, dan telah dijadikan rekomendasi optimasi pada bab penutup.

Tabel 4. 51 Rekapitulasi Pengujian NFR

| **Kode** | **Aspek** | **Target** | **Hasil** | **Status** |
|---|---|---|---|---|
| KNF-01 | Kinerja | < 3 dtk | 4/5 halaman lolos, | Terpenuhi |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| Kode | Aspek | Target | Hasil | Status |
|---|---|---|---|---|
| | | | Dashboard CEO 10,4 dtk | |
| KNF-02 | Kinerja | < 5 dtk | Rata-rata 0,6 dtk | Terpenuhi |
| KNF-03 | Keamanan | 100% terproteksi | 29/29 (100%) | Terpenuhi |
| KNF-04 | Keamanan | Sesuai RBAC | 10/10 ditolak (403) | Terpenuhi |
| KNF-05 | Keamanan | bcrypt | *Hash bcrypt*, tanpa *plaintext* | Terpenuhi |
| KNF-06 | Keandalan | Galat terkontrol | 6/6 terkontrol, tanpa *crash* | Terpenuhi |
| KNF-07 | Kompatibilitas | Chrome & Safari | Berjalan dan Berfungsi | Terpenuhi |
| KNF-08 | Kompatibilitas | Adaptif ≥ 360px | Responsif | Terpenuhi |
| KNF-09 | Kemudahan Penggunaan | SUS > 68 | 78,0 | Terpenuhi |
| KNF-10 | Skalabilitas | > 50 VU | 0% gagal, p95 2.780 ms | Terpenuhi |

Sumber: Olahan Peneliti

### 4.4.8 Hasil Analisis Pengujian

Rekapitulasi hasil pengujian dari tiga metode yang telah dieksekusi disajikan secara terpadu untuk memberikan gambaran menyeluruh tingkat keberhasilan sistem

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

dalam memenuhi spesifikasi yang telah dirancang. Ringkasan kuantitatif dari ketiga metode disajikan pada Tabel 4.52.

Tabel 4. 52 Rekapitulasi Hasil Tiga Metode Pengujian

| **Metode** | **Jumlah Item Uji** | **Lulus / Diterima** | **Tingkat Keberhasilan** | **Ukuran Hasil** |
|---|---|---|---|---|
| *Black Box Testing* | 32 test case | 32 | 100% | TC-01 sampai TC-32 berstatus Berhasil |
| *User Acceptance Testing* | 5 skenario / 20 kriteria | 5 skenario / 20 kriteria | 100% | UAT-S1 sampai UAT-S5 berstatus *Accepted* |
| *Usability Testing* (SUS) | 10 responden | - | Skor 78,0 | Kategori *Acceptable* |
| *Usability Testing* (Task Success Rate) | 4 task × 10 responden = 40 attempts | 37 berhasil | 92,5% | T-1, T-2 mencapai 100%, T-3 90%, T-4 80% |

Sumber: Olahan Peneliti

Tingkat keberhasilan seratus persen pada *Black Box Testing* dan UAT menunjukkan bahwa sistem telah memenuhi seluruh spesifikasi fungsional yang dirumuskan maupun Kriteria Penerimaan yang dirumuskan. Hasil ini perlu dibaca dengan konteks bahwa pengujian *Black Box Testing* dilakukan terhadap delapan modul fungsional inti yang merepresentasikan kontribusi utama penelitian, mencakup Autentikasi, RBAC, Pengelolaan Proyek, Pencatatan Progres, Pengelolaan KPI, *Dashboard Monitoring*, *Dashboard* KPI, dan Pengelolaan Akun. Karakteristik Fungsional pelengkap yaitu KF-07 Deteksi Keterlambatan per *Milestone*, KF-08

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Notifikasi, KF-13 Laporan Proyek, dan KF-14 Laporan KPI diverifikasi secara fungsional melalui Kriteria Penerimaan yang relevan dan skenario UAT end-to-end, sehingga cakupan pengujian bersifat komprehensif terhadap seluruh lima belas Karakteristik Fungsional.

Hasil *Usability Testing* dengan skor SUS 78,0 menunjukkan bahwa sistem *Acceptable* pengguna dengan tingkat kepuasan di atas rata-rata industri. Skor ini mencerminkan bahwa rancangan antarmuka yang telah dirumuskan dan diimplementasikan berhasil menyajikan informasi yang dibutuhkan dengan cara yang dapat dipahami oleh berbagai peran pengguna tanpa training khusus. Skor tertinggi dari peran Project Manager dan CEO mengkonfirmasi bahwa fitur inti yang menjadi kontribusi penelitian, yaitu *monitoring* proyek dengan *state machine* WFMS dan *dashboard* KPI dengan *Weighted Scoring Method*, dirancang dengan baik dari sudut pandang *user* experience. Task Success Rate sebesar 92,5 persen mengkonfirmasi bahwa pengguna dapat menyelesaikan task operasional umum dengan tingkat keberhasilan tinggi, dengan dua area perbaikan UX yang teridentifikasi untuk iterasi berikutnya.

Selama pelaksanaan pengujian, teridentifikasi delapan test case yang pada iterasi awal menunjukkan ketidaksesuaian dari dua kategori sebagaimana telah dipaparkan, serta dua catatan UX dari *Usability Testing* sebagaimana telah disebutkan. Seluruh delapan ketidaksesuaian *Black Box Testing* telah diperbaiki sebelum pengujian final dieksekusi, sehingga seluruh tiga puluh dua test case pada tabel pengujian berstatus PASSED. Dua catatan UX dari *Usability Testing* dijadwalkan sebagai backlog perbaikan iteratif yang akan ditangani setelah skripsi diselesaikan, mengingat keduanya bersifat enhancement dan tidak menghalangi penggunaan sistem secara fungsional. Pendekatan ini sejalan dengan prinsip rekayasa perangkat lunak iteratif yang membedakan antara defect fungsional yang harus segera diperbaiki dengan enhancement yang dapat dijadwalkan pada iterasi berikutnya.

### 4.4.9 Evaluasi terhadap Tujuan Penelitian

Tujuan penelitian sebagaimana telah ditetapkan merupakan acuan utama dalam menilai apakah penelitian ini berhasil mencapai sasaran yang direncanakan.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Evaluasi terhadap tujuan penelitian dilakukan dengan memetakan setiap tujuan ke hasil pengujian yang telah dipaparkan, sehingga pencapaian setiap tujuan dapat diverifikasi dengan bukti pengujian yang konkret. Pemetaan tujuan penelitian terhadap hasil pengujian disajikan pada Tabel 4.53.

Tabel 4. 53 Pemetaan Tujuan Penelitian terhadap Hasil Pengujian

| Tujuan | Rumusan Tujuan | Bukti Pencapaian | Status |
|---|---|---|---|
| T-1 | Merancang sistem *monitoring* proyek berbasis *workflow* yang mampu mengelola *lifecycle* proyek dan *milestone* secara terstruktur | Implementasi modul WFMS dengan tujuh berkas service, lima *state* proyek, empat *state milestone*, transition matrix yang ketat, dan *audit trail append-only*. Verifikasi melalui TC-13 hingga TC-17, TC-29 hingga TC-31, dan UAT-S2 | Tercapai |
| T-2 | Mengembangkan mekanisme penilaian kinerja Consultant berbasis empat dimensi KPI dengan metode *Weighted Scoring Method* | Implementasi modul KPI Calculation dengan empat fungsi dimensi murni, agregasi WSM dengan auto-normalisasi, dan persistensi *snapshot* dengan invariant immutability. Verifikasi melalui TC-18 hingga TC-23 dan UAT-S3 | Tercapai |
| T-3 | Mengintegrasikan modul *monitoring* proyek dengan penilaian KPI sehingga | Implementasi langkah ketujuh *side effects* pada *completeProject* (Gambar | Tercapai |

| Tujuan | Rumusan Tujuan | Bukti Pencapaian | Status |
|---|---|---|---|
| | perhitungan KPI dapat dipicu otomatis saat proyek mencapai status tertentu | 4.49) yang memicu invoice final dan auto-compute KPI saat proyek Completed. Verifikasi melalui TC-29 (end-to-end test) dan UAT-S2 | |
| T-4 | Menyediakan *dashboard monitoring real-time* yang menyajikan ringkasan kinerja organisasi maupun individu sesuai peran pengguna | Implementasi *dashboard* tersegmentasi per peran (Gambar 4.51 *Dashboard* CEO, Gambar 4.52 My KPI Consultant) dengan data agregat dan tren periode. Verifikasi melalui TC-24, TC-25, UAT-S3, dan Usability T-2 | Tercapai |
| T-5 | Menerapkan mekanisme kontrol akses berbasis peran yang membatasi akses fitur sesuai tanggung jawab pengguna | Implementasi RBAC dua lapis dengan *middleware* permission pada level *routing* dan otorisasi ownership pada modul WFMS (Gambar 4.34). Verifikasi visual melalui perbedaan *sidebar* CEO dan Consultant (Gambar 4.45 vs 4.46), pengujian melalui TC-05, TC-06, TC-25, dan TC-27 | Tercapai |

Sumber: Olahan Peneliti



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Pencapaian ini terkonfirmasi melalui tiga lapis verifikasi, yaitu kebenaran fungsional dari *Black Box Testing* dengan tingkat keberhasilan seratus persen, penerimaan pengguna dari *User Acceptance Testing* yang juga mencapai seratus persen pada seluruh skenario, serta kualitas pengalaman pengguna dari *Usability Testing* dengan skor SUS di atas rata-rata industri. Pencapaian seluruh tujuan ini memberikan dasar untuk pembahasan kesimpulan penelitian, di mana kontribusi penelitian akan dirumuskan secara formal beserta catatan peningkatan UX yang telah diidentifikasi.

Penelitian ini juga menghasilkan dua kontribusi sekunder yang patut dicatat. Kontribusi sekunder pertama adalah pendekatan *custom state machine* yang menggabungkan immutability constraint dengan *transition matrix* sebagai *single source of truth*, yang dapat dijadikan referensi pada penelitian lanjutan tentang *workflow* management pada sistem informasi perusahaan Consultant. Kontribusi sekunder kedua adalah pendekatan dual-source *snapshot* KPI yang memisahkan *state* preliminary yang dihasilkan otomatis oleh sistem dengan *state* finalized yang dikunci oleh CEO, memberikan keseimbangan antara automation dan governance pada penilaian kinerja organisasional.

POLITEKNIK NEGERI JAKARTA

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

# BAB V
# PENUTUP

## 5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian, penelitian ini berhasil membangun aplikasi *monitoring* proyek berbasis *workflow* yang terintegrasi dengan penilaian kinerja karyawan berbasis *Key Performance Indicator* (KPI) di PT DSK Global Konsultama sesuai tujuan yang ditetapkan. Pemantauan *lifecycle* proyek dan *milestone* diwujudkan melalui modul *Workflow Management System* berbasis *state machine* deterministik, penilaian kinerja yang objektif dihasilkan modul *KPI Calculation Engine* atas empat dimensi dengan metode *Weighted Scoring Method*, perhitungan KPI terpicu otomatis saat proyek *Completed*, dan informasi manajerial disajikan melalui *dashboard* tersegmentasi per peran dengan *Role-Based Access Control* dua lapis. Keberhasilan ini terkonfirmasi melalui *Black Box Testing* 100% pada 32 *test case*, *User Acceptance Testing* dengan seluruh 5 skenario diterima, dan *Usability Testing* dengan skor *System Usability Scale* 78,0 berkategori *Acceptable*. Dengan demikian, sistem terbukti memenuhi kebutuhan fungsional dan diterima pengguna sebagai solusi atas pemantauan proyek dan penilaian kinerja yang sebelumnya manual dan subjektif.

## 5.2 Saran

Penelitian ini berfokus pada satu layanan utama PT DSK Global Konsultama sebagai studi kasus, sehingga sistem memiliki potensi pengembangan yang luas. Saran pengembangan disampaikan sebagai berikut:

- Perluasan cakupan sistem ke layanan lainnya sehingga menjadi platform *monitoring* terpadu untuk seluruh lini bisnis perusahaan.
- Generalisasi sistem agar dapat diadopsi oleh perusahaan konsultan lain melalui mekanisme *multi-tenant* dan konfigurasi *state machine* yang dapat disesuaikan.
- Pengembangan lanjutan oleh tim internal perusahaan terhadap peningkatan yang telah teridentifikasi, yaitu optimasi performa sistem untuk mendukung volume proyek yang lebih besar seiring pertumbuhan perusahaan.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

# DAFTAR PUSTAKA

Adieb, M.R.A., Sugiharto, A., Adhy, S., 2024. Analisis Pengaruh Kualitas Website Shopee Terhadap Kepuasan Pengguna Berdasarkan Metode WebQual 4.0 (Studi Kasus : Mahasiswa Pengguna Shopee di Universitas Diponegoro). Jurnal Masyarakat Informatika 15, 39–55.

Agustha, E.B., Adhy, S., Nugraheni, D.M.K., 2024. Pengembangan Aplikasi Monitoring Informasi Proyek Berbasis Web Menggunakan Metode ICONIX Process (Studi Kasus: PT Telkom Akses Witel Semarang). Jurnal Masyarakat Informatika 15, 81–96.

Akbartio, C.D., Setiawan, D., Abidin, Z., 2025. Sistem Informasi Layanan Teknologi Informatika dan Komunikasi Berbasis Web. INFORMASI (Jurnal Informatika dan Sistem Informasi) 17, 141–162.

Anardani, S., Yunitasari, Y., Sussolaikah, K., 2023. Analisis Perancangan Sistem Informasi Monitoring dan Evaluasi Kerjasama Menggunakan UML. Remik 7, 522–532.

Bangor, A., Kortum, P., Miller, J., 2009. Determining what individual SUS scores mean; adding an adjective rating. J. Usability Stud. 4, 114–23.

Barus, S.K., Yahfizham, Y., 2024. Perancangan Manajemen Proyek Sistem Informasi Web Metode Waterfall Pada Perusahaan Kontraktor. Journal of Information System, Computer Science and Information Technology 5, 157–169.

Brooke, J., 1996. SUS: A “Quick and Dirty” Usability Scale. Usability Evaluation In Industry 207–212.

Cosa, M., Torelli, R., 2024. Digital Transformation and Flexible Performance Management: A Systematic Literature Review of the Evolution of Performance Measurement Systems. Global Journal of Flexible Systems Management 25, 445–466.

Datatracker, 2026. HTTP Semantics [WWW Document]. URL https://datatracker.ietf.org/doc/rfc9110/

dev.mysql, 2026. InnoDB and the ACID Model [WWW Document]. URL https://dev.mysql.com/doc/refman/8.4/en/mysql-acid.html

Goel, K., Bandara, W., Gable, G., 2023. Conceptualizing Business Process Standardization: A Review and Synthesis. Schmalenbach Journal of Business Research 75, 195–237.

Gonçalves, C.T., Gonçalves, M.J.A., Campante, M.I., 2023. Developing Integrated Performance Dashboards Visualisations Using Power BI as a Platform. Information (Switzerland) 14.

Haerudin, C.C., Devianto, Y., 2023. Pemodelan Proses Bisnis Reengineering Mengggunakan Business Process Modeling Notation. JITASI: Jurnal Teknik Informatika dan Sistem Informasi 10, 5–12.

Hananiya, P., Badau, K.M., Awodoyin, F.O., Inusa, Y.M., 2024. INFORMATION SYSTEMS IN ORGANIZATIONS.

Hermawan, A.B., Zabina, K.A., Alfarisqi, M.R., Wati, S.F.A., 2023. Systematic Literature Review Tentang Manajemen Proyek Dalam Sistem Informasi. Prosiding Seminar Nasional Teknologi dan Sistem Informasi 3, 1–11.

Hersusetiyati, Febrianti, W., Sari, R.D., Yanshah, A.P., 2024. Analisis Kinerja Karyawan Melalui Penerapan Key Performance Indicator (KPI) di PT Jasa dan

Kepariwisataan Jawa Barat (Perseroda). Jurnal Digital Bisnis, Modal Manusia, Marketing, Entrepreneurship, Finance, & Strategi Bisnis (DImmensi) 4, 14.
Inayah, S., Firdaus, M.I., Khairunisa, Y., Respatiningsih, I., Hamka, Dianawati, E., Herlina, Yulaikah, Tooy, S.M., 2024. Sistem Informasi Manajemen.
ISO, 2018. Ergonomics of human-system interaction 2018, 1–19.
ISO/IEC/IEEE, 2022. Software and systems engineering - Software testing 2022.
ISTQB, 2024. Certified Tester Foundation Level Syllabus.
Kalischko, T., Riedl, R., 2021. Electronic Performance Monitoring in the Digital Workplace : Conceptualization , Review of Effects and Moderators , and Future Research Opportunities. Front. Psychol. 12, 1–15.
Karlina, A., Samanhudi, Di., 2023. Pengukuran Kinerja Sumber Daya Manusia menggunakan Metode Human Resources Scorecard. Jurnal Fokus Manajemen Bisnis 4, 108–117.
kpi.org, 2026. Key Performance Indicators (KPIs): Definition and Examples. [WWW Document]. URL https://www.kpi.org/
Mary, T., Febriyani, N., 2025. Peningkatan Keamanan Sistem Informasi Berbasis Laravel 12 dengan Rate Limiting dan Role-Based Access Control (RBAC). Jurnal Teknologi Dan Sistem Informasi Bisnis 7, 473–481.
MDN Web Docs, 2024. Fetch API [WWW Document].
Musva, S.R., Ramadhani, D., 2025. Design and Development of a Web-Based Project Monitoring System Using the Prototype Method. Journal of Software Engineering and Multimedia (JASMED) 3, 101–110.
nodejs.org, 2026. The Node.js Event Loop [WWW Document]. URL https://nodejs.org/en/learn/asynchronous-work/event-loop-timers-and-nexttick
Nunes, F., Alexandre, E., Gaspar, P.D., 2024. Implementing Key Performance Indicators and Designing Dashboard Solutions in an Automotive Components Company: A Case Study. Adm. Sci. 14.
Nurlisa, U., 2021. SISTEM INFORMASI MONITORING PROYEK DINAS PEKERJAAN UMUM DAN PENATAAN RUANG KABUPATEN KUANTAN SINGINGI. Jurnal Perancanaan, Sains, Teknologi, dan Komputer (JuPerSaTeK) 32, 167–186.
Ochoa, W., Larrinaga, F., Pérez, A., 2023. Context-aware workflow management for smart manufacturing: A literature review of semantic web-based approaches. Future Generation Computer Systems 145, 38–55.
Pebriad, M.S., Salman, P., Fattah, T.K., 2023. IMPLEMENTASI USE CASE DIAGRAM DAN ACTIVITY DIAGRAM DALAM PERANCANGAN APLIKASI KALKULATOR PAJAK BAGI UMKM. Multitek Indonesia: Jurnal Ilmiah 17, 60–68.
Picozzi, P., Nocco, U., Pezzillo, A., De Cosmo, A., Cimolin, V., 2024. The Use of Business Intelligence Software to Monitor Key Performance Indicators (KPIs) for the Evaluation of a Computerized Maintenance Management System (CMMS). Electronics MDPI 305–310.
Prasetia, Y.A., Manongga, D., 2024. Role-Based Access Control (RBAC) Untuk Sistem Otorisasi Terpusat Berbasis Flask Studi Kasus Pt. Xyz. JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika) 9, 1768–1778.
react.dev, 2026. Passing Props to a Component [WWW Document]. URL https://react.dev/learn/passing-props-to-a-component

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Rizal, Hidayatullah, 2025. Manajemen Kinerja Efektif untuk Mencapai Tujuan Organisasi. Jurnal Ilmu Manajemen, Ekonomi dan Kewirausahaan 5, 76–87.

Rizana, A.F., Kurniawati, F.D.J., Rumanti, A.A., 2023. The Employee Performance Evaluation by using the Hybrid Multi-Criteria Decision-Making Approach. MIX: Jurnal Ilmiah Manajemen 18, 31–43.

Rosemann, M., Brocke, J. vom, Van Looy, A., Santoro, F., 2024. Business process management in the age of AI – three essential drifts. Information Systems and e-Business Management 22, 415–429.

Sabar, Y., Sari, R., Oktavianti, E., 2017. Perancangan Sistem Informasi Pembelajaran Berbasis Web. Multinetics 3, 45.

Santi, P.A.D., Afwani, R., Albar, Moh.A., Anjarwani, S.E., Mardiansyah, A.Z., 2022. Black Box Testing with Equivalence Partitioning and Boundary Value Analysis Methods (Study Case: Academic Information System of Mataram University), Proceedings of the First Mandalika International Multi-Conference on Science and Engineering 2022, MIMSE 2022 (Informatics and Computer Science). Atlantis Press International BV.

Soufitri, F., 2023. Konsep Sistem Informasi.

Utama, Y., 2022. Sistem Informasi Berbasis Web: Konsep dan Implementasi. Jurnal Sistem Informasi 10, 55–63.

Yuricha, Phan, I.K., 2023. Penerapan Role Based Access Control dalam Sistem Supply Chain Management Berbasis Cloud. MALCOM: Indonesian Journal of Machine Learning and Computer Science 3, 339–348.

Zulfikar, Moch.I., 2024. Pentingnya Manajemen Kinerja Dalam Meningkatkan Produktivitas Dan Motivasi Karyawan. PPIMAN Pusat Publikasi Ilmu Manajemen 2, 01–07.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

# DAFTAR RIWAYAT HIDUP PENULIS



Muhamad Faried

Lahir di Bogor pada 19 September 2004. Lulus dari SDN Gunung Batu 2 Kota Bogor tahun 2016, SMPN 11 Kota Bogor tahun 2019, SMK Informatika Pesat Kota Bogor tahun 2022. Lahir di Bogor, 19 September 2004. Melanjutkan studi tahun 2022 di CCIT-FTUI dan lulus tahun 2024. Saat ini sedang menempuh pendidikan D4 pada Program Studi Teknik Informatika Jurusan Teknik Informatika dan Komputer di Politeknik Negeri Jakarta.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

# LAMPIRAN

## Lampiran 1 Hasil Observasi

| | | |
|---|---|---|
| Tujuan observasi | : | Untuk memperoleh pemahaman menyeluruh mengenai kondisi *existing* proses *monitoring* proyek dan penilaian kinerja karyawan yang berjalan di PT DSK Global Konsultama, mengidentifikasi media dan tools yang digunakan, pola koordinasi antar peran (CEO, COO, Project Manager, Human *Resources*, dan Consultant), serta hambatan operasional yang menjadi akar permasalahan dalam pengelolaan proyek dan evaluasi kinerja karyawan. |
| Teknik observasi | : | Observasi non-partisipatif (non-participant observation), yaitu peneliti mengamati proses kerja yang berjalan tanpa terlibat langsung dalam aktivitas operasional perusahaan. Peneliti mencatat fenomena yang ditemui menggunakan field notes dan didukung dengan dokumentasi visual (screenshot media kerja, contoh dokumen) untuk memperkuat bukti temuan. |
| Objek dan periode observasi | : | Proses *monitoring* proyek dan penilaian kinerja karyawan di PT DSK Global Konsultama, yaitu mencakup media pencatatan dan pelaporan yang digunakan (Excel, WhatsApp, *email*), pola koordinasi lintas peran antara CEO, COO, PM, HR, dan Consultant, mekanisme deteksi keterlambatan dan tindak lanjutnya, parameter dan praktik penilaian kinerja *existing*, serta dokumen kerja yang digunakan dalam siklus proyek. Periode observasi dilakukan selama dua bulan mulai Januari 2026 – Februari 2026 yang mencakup minimal satu siklus penuh proyek konsultasi dari fase planning hingga delivery. |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Catatan : Temuan kunci yang berhasil diidentifikasi selama observasi: dari sekitar 110 *inquiry* klien yang masuk dalam setahun terakhir, hanya sekitar 50 proyek yang benar-benar masuk tahap eksekusi pasca-EL dengan DP lunas dan dokumen klien lengkap. Dari 50 proyek aktif tersebut, sekitar 12 sampai 15 proyek mengalami keterlambatan penyelesaian dengan akar penyebab yang beragam, dengan 5 proyek di antaranya merupakan keterlambatan murni dari sisi Consultant dengan keterlambatan terparah mencapai dua minggu dari jadwal rencana. Proses *monitoring* proyek dilakukan secara manual menggunakan kombinasi Excel, WhatsApp, dan *email* tanpa *dashboard real-time*, parameter Key Performance Indicator (KPI) untuk penilaian kinerja karyawan belum ditetapkan secara konsisten dan terstandar, data proyek dan dokumen kerja tersebar di berbagai media tanpa basis data terpusat sehingga menyulitkan tracking lintas peran, tidak terdapat mekanisme deteksi dini keterlambatan yang menyebabkan tindakan korektif sering dilakukan terlambat, dan penilaian kinerja Consultant masih bersifat subjektif berdasarkan observasi PM tanpa keterhubungan dengan data progres proyek yang konkret.

| No | Aspek yang diamati | Kondisi saat ini | Indikasi masalah | Dampak Operasional |
|---|---|---|---|---|
| 1 | Ketepatan waktu penyelesaian proyek | Ditemukan indikasi keterlambatan proyek dan keluhan dari klien | *Monitoring* tidak *real-time* dan kontrol tenggat lemah | Keluhan klien, penurunan kualitas layanan, risiko reputasi |

| No | Aspek yang diamati | Kondisi saat ini | Indikasi masalah | Dampak Operasional |
|---|---|---|---|---|
| 2 | Sentralisasi data proyek | Informasi progres, status, dan tenggat belum terkonsolidasi dalam satu sistem terpusat | Data tersebar, sulit ditelusuri | Laporan lambat disusun, kontrol manajerial melemah |
| 3 | Metode pelaporan progres | *Update* progres masih bergantung pada pelaporan manual | Ketergantungan pada kedisiplinan individu | Progres tidak *real-time*, risiko terlambat terdeteksi |
| 4 | Media komunikasi kerja | Komunikasi dan koordinasi banyak melalui komunikasi informal | Tidak ada jejak proses yang rapi (*audit trail*) | Riwayat keputusan atau ubah-ubah instruksi sulit dibuktikan |
| 5 | Standarisasi status atau *workflow* | Belum ada *workflow* baku end-to-end yang mengunci tahapan kerja & status proyek | Status proyek berpotensi multitafsir | Sulit memonitor *bottleneck* dan tahap paling rawan delay |
| 6 | Pelacakan tenggat waktu | Pemantauan deadline bergantung pada pengingat manual atau komunikasi | Tidak ada alert atau peringatan dini otomatis | Keterlambatan diketahui saat sudah terjadi |
| 7 | Bukti pekerjaan (evidence) | Dokumentasi bukti progres atau hasil kerja belum | Validasi progres subjektif | Potensi klaim progres tidak akurat, dispute |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| No | Aspek yang diamati | Kondisi saat ini | Indikasi masalah | Dampak Operasional |
|---|---|---|---|---|
| | | terstruktur sebagai bagian dari proses *monitoring* | | internal meningkat |
| 8 | Konsistensi data progres antar pihak | Potensi perbedaan versi progres antara pihak pelaksana, PM, dan manajemen karena kanal data berbeda | Single source of truth tidak ada | Miskomunikasi dan keputusan manajerial tidak presisi |
| 9 | Mekanisme evaluasi kinerja | Penilaian kinerja masih rentan karena mekanisme belum tertopang data *monitoring* yang terstruktur | Evaluasi tidak berbasis data operasional | Sulit mengaitkan hasil kerja dengan *output* proyek |
| 10 | Penetapan bobot KPI | Bobot KPI belum ditetapkan melalui mekanisme yang konsisten dan terdokumentasi | KPI tidak stabil antar periode atau penilai | Persepsi ketidakadilan, sulit dibandingkan lintas waktu |
| 11 | Transparansi penilaian | Penilaian cenderung kurang transparan bagi karyawan (standar & dasar nilai tidak jelas) | Karyawan tidak paham ekspektasi | Motivasi turun, perbaikan kinerja tidak terarah |
| 12 | Objektivitas penilaian | Penilaian berpotensi subjektif karena minim sumber data | Ruang bias penilai lebih besar | Konflik, resistensi, dan rendahnya akuntabilitas |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

| No | Aspek yang diamati | Kondisi saat ini | Indikasi masalah | Dampak Operasional |
|---|---|---|---|---|
| | | terukur dan jejak aktivitas atau progres | | |
| 13 | Kebutuhan sistem terintegrasi | *Monitoring* proyek dan penilaian KPI belum berada pada satu alur data yang sama | Data KPI tidak "menarik" dari realisasi proyek | Perhitungan KPI sulit diotomatisasi dan sulit diaudit |

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

## Lampiran 2 Hasil Wawancara Proses Bisnis Existing

Identitas:

| | | |
|---|---|---|
| Jabatan | : | Direksi |
| Nama Lengkap | : | Galih Gumilang |
| Tanggal Wawancara | : | |
| Media | : | Daring (Zoom) |
| Durasi | : | 15 Menit |

Transkrip:

1. Dalam 1 tahun terakhir, apakah ada terjadi keterlambatan penyelesaian proyek?
   **Jawaban:** Iya, ada keterlambatan pada beberapa proyek. Umumnya karena progres tidak terpantau cepat sehingga tindakan korektif terlambat.
2. Dampak keterlambatan yang paling terasa apa?
   **Jawaban:** Ada keluhan dari klien terkait mundurnya jadwal, dan tim jadi harus lembur untuk mengejar target.
3. *Monitoring* proyek saat ini dilakukan dengan cara apa?
   **Jawaban:** Masih manual. Progres biasanya dilaporkan lewat chat dan rekap Excel. Kalau butuh status terbaru, harus tanya ke PM.
4. Apakah data progres terpusat dan mudah ditelusuri?
   **Jawaban:** Belum. Informasi ada di beberapa tempat sehingga kadang ada perbedaan versi status.
5. Bagaimana penilaian kinerja pelaksana proyek saat ini?
   **Jawaban:** Penilaian ada, tapi belum sepenuhnya berbasis data proyek. Kadang masih bergantung pada penilaian atasan.
6. Kalau dibuat aplikasi *monitoring* dan KPI terintegrasi, apa yang paling dibutuhkan?
   **Jawaban:** *Dashboard* proyek bermasalah, peringatan jika melewati *milestone*, dan rekap kinerja berdasarkan *output* kerja.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Identitas:

| | | |
|---|---|---|
| Jabatan | : | Project Manager |
| Nama Lengkap | : | Ahmad Girang Adipura dan Panji Ega Saputra |
| Tanggal Wawancara | : | |
| Media | : | Tatap Muka |
| Durasi | : | 25 Menit |

Transkrip:

1. Dalam 1 tahun terakhir, ada berapa total proyek di perusahaan?
   **Jawaban:** Dalam satu tahun terakhir kami menerima sekitar 110 *inquiry* klien, namun yang masuk tahap eksekusi (pasca-EL, DP lunas, dokumen lengkap) hanya sekitar 50 proyek. Sisanya tidak berlanjut karena klien menunda, batal, atau dokumen belum lengkap.
2. Berdasarkan wawancara bersama direksi, terkait keterlambatan proyek, ada berapa total proyek yang mengalamai keterlambatan?
   **Jawaban:** Dari 50 proyek aktif, sekitar 12 sampai 15 mengalami keterlambatan dengan penyebab beragam. Sekitar 5 di antaranya murni karena sisi Consultant internal, sisanya akibat respons klien yang lambat atau hambatan koordinasi. Keterlambatan terparah dari sisi Consultant mencapai dua minggu dari jadwal.
3. Rata-rata proyek telat dalam berapa hari/minggu?
   **Jawaban:** Sejauh ini yang paling parah itu sampai 2 minggu.
4. Tahap mana yang paling sering menyebabkan keterlambatan?
   **Jawaban:** Biasanya di tengah sampai akhir, saat menunggu revisi/approval dan saat workload tim menumpuk.
5. Bagaimana *update* progres dilakukan?
   **Jawaban:** Anggota tim lapor lewat chat, lalu saya rekap manual ke Excel. Tidak semua *update* rutin sehingga saya harus follow-up.
6. Apakah ada *workflow*/status pekerjaan yang baku?
   **Jawaban:** Belum baku. Status lebih berupa kesepakatan informal, jadi sulit memantau *bottleneck* secara konsisten.
7. Apa masalah paling besar dari *monitoring* saat ini?

**Jawaban:** Tidak *real-time* dan tidak ada alert. Kalau ada risiko telat, sering ketahuan setelah telat.

8. Menurut Bapak, KPI apa yang paling relevan diukur dari data proyek?

**Jawaban:** On-time completion untuk tugas/*milestone*, jumlah revisi karena kualitas, dan kedisiplinan *update* progres.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Identitas:

| | | |
|---|---|---|
| Jabatan | : | Consultant |
| Nama Lengkap | : | Muhammad Revy Oktafiano |
| Tanggal Wawancara | : | |
| Media | : | Tatap Muka |
| Durasi | : | 20 Menit |

Transkrip:

1. Bagaimana Anda menerima tugas dan instruksi kerja?
   **Jawaban:** Biasanya dari PM lewat chat atau saat meeting. Kadang ada perubahan instruksi karena revisi dari klien.
2. Bagaimana melaporkan progres dan bukti pekerjaan?
   **Jawaban:** Saya lapor lewat chat dan kirim *file*/link. Kalau banyak tugas, kadang laporan tidak langsung saya *update*.
3. Apa hambatan yang sering membuat terlambat?
   **Jawaban:** Menunggu *input* dari pihak lain, revisi mendadak, dan beban tugas yang menumpuk.
4. Apakah anda memahami KPI yang dinilai?
   **Jawaban:** Saya tahu dinilai dari hasil kerja dan ketepatan waktu, tapi detail target/bobotnya tidak selalu jelas.
5. Menurut anda, Indikator KPI yang paling adil?
   **Jawaban:** Ketepatan waktu tugas yang memang dalam kontrol saya, kualitas *output*, dan kepatuhan *update* progres.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

**Lampiran 3 Hasil Wawancara Bobot KPI**

Identitas:

| | | |
|---|---|---|
| Jabatan | : | Direksi |
| Nama Lengkap | : | Galih Gumilang |
| Tanggal Wawancara | : | |
| Media | : | Daring (Zoom) |
| Durasi | : | 15 Menit |

Transkrip:

1. Untuk penilaian kinerja Consultant, saat ini bagaimana prosesnya, Pak?

   **Jawaban:** Jujur saja, sekarang itu masih berdasarkan feeling dan pengamatan saya selama proyek berjalan. Tidak ada standar yang baku. Jadi hasilnya pasti subjektif dan Consultant pun tidak tahu persis dasar penilaiannya apa.
2. Kalau Bapak membayangkan sistem yang ideal untuk penilaian kinerja, kira-kira aspek apa saja yang ingin Bapak ukur?

   **Jawaban:** Yang paling penting itu kemampuan Consultant menyelesaikan tugasnya. Itu yang utama. Lalu ketepatan waktu, karena proyek TP ini ada deadline-nya ke DJP, jadi kalau Consultant terlambat itu dampaknya ke klien langsung. Terus kualitas outputnya juga penting, karena dokumennya harus standar profesional. Terakhir, kedisiplinan *update* progres, kalau Consultant tidak *update* di sistem, kita tidak tahu dia kerja apa enggak.
3. Dari semua aspek tadi, kalau Bapak harus memberi bobot kepentingannya, kira-kira bagaimana proporsinya?

   **Jawaban:** Penyelesaian tugas itu yang paling besar, mungkin sekitar 35%. Ketepatan waktu dan kualitas *output* saya kasih sama, masing-masing 25%. Sisanya yang 15% untuk kedisiplinan *update* progres. Totalnya pas 100%.
4. Untuk aspek kualitas *output*, Bapak ingin penilaiannya dilakukan oleh siapa dan dalam format apa?

   **Jawaban:** Sebaiknya PM yang menilai, karena PM yang paling tahu kualitas pekerjaan Consultantnya. Bisa pakai skala 1 sampai 5 saja, yang penting ada standarnya biar tidak subjektif.

5. Apakah Bapak ingin bobot keempat aspek ini bisa dikonfigurasi di sistem, atau cukup tetap?

   **Jawaban:** Kalau bisa dikonfigurasi lebih baik. Kebutuhan perusahaan bisa berubah, jadi fleksibel itu penting. Tapi tetap harus ada validasi supaya totalnya selalu 100%.

6. Terakhir pak, apakah ada hal lain yang Bapak harapkan dari sistem ini?

   **Jawaban:** Yang paling penting itu sistemnya membantu kita bekerja lebih terstruktur dan datanya bisa dipertanggungjawabkan. Selama ini semua bergantung pada orang, kalau orangnya lupa atau keluar, informasinya hilang. Dengan sistem, harapannya semua tercatat dan bisa ditelusuri kapan pun dibutuhkan.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
   a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
   b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

**Lampiran 4 Tabel Rekap Wawancara**

| Jabatan | Temuan *Monitoring* Proyek | Temuan KPI | Dampak |
|---|---|---|---|
| Direksi | *Monitoring* manual, data tersebar | Penilaian belum berbasis data penuh | Keluhan klien, lembur |
| Human *Resources* | Data realisasi sulit dikumpulkan | Bobot KPI belum baku, transparansi kurang | Komplain atau masukan karyawan |
| Project Manager | Tidak *real-time*, tidak ada alert, status tidak baku | KPI bisa ditarik dari data proyek jika tersedia | Risiko telat terlambat terdeteksi |
| Consultant | *Update* progres tidak selalu rutin | KPI kurang jelas di level pelaksana | Motivasi & arah perbaikan kurang |

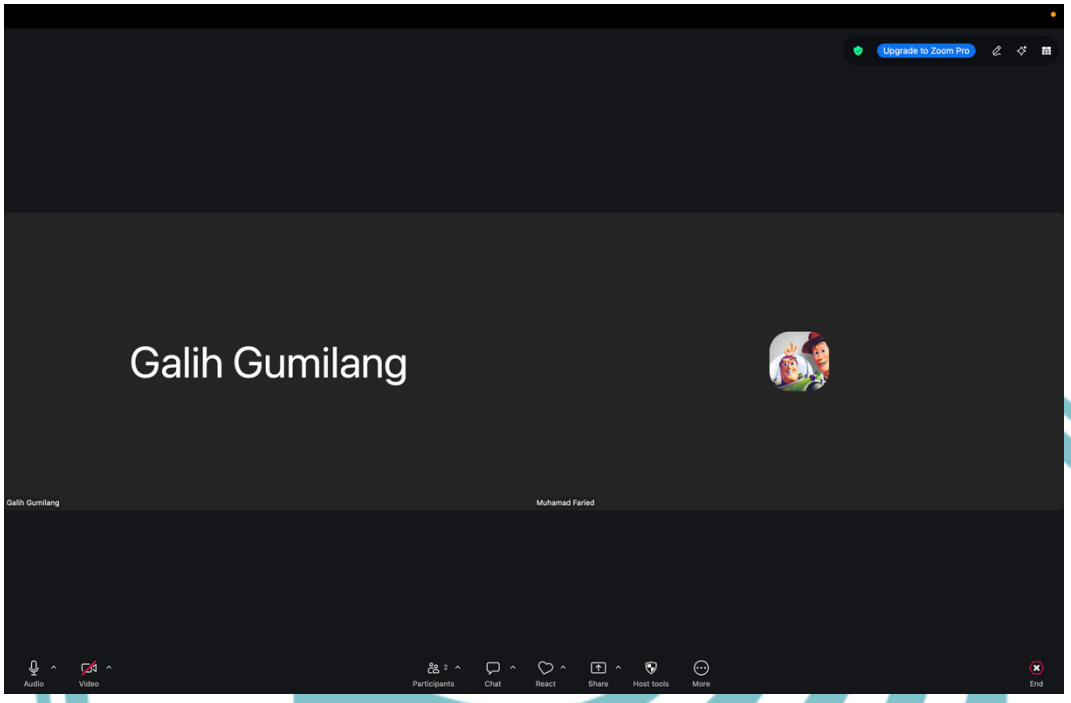
© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

## Lampiran 5 Dokumentasi Wawancara




Dokumentasi wawancara dengan Direksi



Dokumentasi wawancara dengan Project Manager



Dokumentasi wawancara dengan Consultant