



© Hak Cipta milik 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



RANCANG BANGUN SISTEM PENDUKUNG *CODE REVIEW* BERBASIS *ARTIFICIAL INTELLIGENCE* MENGGUNAKAN *RULEBOOK* INTERNAL PADA APLIKASI WISE/ESCM

SKRIPSI

RAFIF NURAYDIN ADIPRATAMA
2207412051

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA DAN
KOMPUTER**

POLITEKNIK NEGERI JAKARTA

2026



© Hak Cipta milik 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



RANCANG BANGUN SISTEM PENDUKUNG *CODE REVIEW* BERBASIS *ARTIFICIAL INTELLIGENCE* MENGGUNAKAN *RULEBOOK* INTERNAL PADA APLIKASI WISE/ESCM

SKRIPSI

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

RAFIE NURAYDIN ADIPRATAMA

2207412051

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA DAN
KOMPUTER**

POLITEKNIK NEGERI JAKARTA

2026



SURAT PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan di bawah ini:

Nama : Rafif Nuraydin Adipratama

NIM : 2207412051

Jurusan/Program Studi : Teknik Informatika dan Komputer/Teknik
Informatika

Judul Skripsi : Rancang Bangun Sistem Pendukung *Code Review*
Berbasis *Artificial Intelligence* Menggunakan
Rulebook Internal pada Aplikasi WISE/ESCM

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, 7 Juli 2026

Yang membuat pernyataan



Rafif Nuraydin Adipratama

NIM. 2207412051

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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



© Hak Cipta milik 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

LEMBAR PENGESAHAN

Skripsi diajukan oleh:

Nama : Rafif Nuraydin Adipratama
NIM : 2207412051
Program Studi : Teknik Informatika
Judul Skripsi : Rancang Bangun Sistem Pendukung *Code Review*
Berbasis *Artificial Intelligence* Menggunakan
Rulebook Internal pada Aplikasi WISE/ESCM

Telah diuji oleh tim penguji dalam Sidang Skripsi pada hari Selasa, Tanggal 14 Juli Tahun 2026 dan dinyatakan **LULUS**.

Disahkan oleh:

Pembimbing I	Susana Dwi Yulianti, M.Kom.	()
Penguji I	Iklima Ermis Ismail, S.Kom., M.Kom.	()
Penguji II	Risna Sari, S.Kom., M.T.I.	()
Penguji III	Fajar Septian, M.Kom.	()

Mengetahui:

Jurusan Teknik Informatika dan Komputer

Ketua



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

NIP. 197908032003122003



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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

KATA PENGANTAR

Bismillahirrahmanirrahim.

Alhamdulillah rabbil'alamin, segala puji dan syukur penulis panjatkan ke hadirat Allah Subhanahu wa Ta'ala atas segala limpahan rahmat, karunia, dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul **"Rancang Bangun Sistem Pendukung *Code Review* Berbasis Artificial Intelligence Menggunakan *Rulebook* Internal pada Aplikasi WISE/ESCM"**. Shalawat Serta Salam Senantiasa Tercurah Kepada Nabi Muhammad Shallallahu 'Alaihi Wasallam, beserta keluarga, sahabat, dan seluruh umat yang mengikuti sunnahnya hingga akhir zaman.

Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Terapan (S.Tr.) pada Program Studi Teknik Informatika, Jurusan Teknik Informatika dan Komputer, Politeknik Negeri Jakarta. Penulis menyadari sepenuhnya bahwa keberhasilan penelitian ini tidak dapat dilepaskan dari kontribusi berbagai pihak yang telah memberikan dukungan, bimbingan, dan kepercayaan. Untuk itu, dengan segala ketulusan hati, penulis menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Dr. Syamsurizal, S.E., M.M., selaku Direktur Politeknik Negeri Jakarta.
2. Anita Hidayati, S.Kom., M.Kom., selaku Ketua Jurusan Teknik Informatika dan Komputer Politeknik Negeri Jakarta.
3. Euis Oktavianti, S.Si., M.T.I., selaku Ketua Program Studi Teknik Informatika.
4. Susana Dwi Yulianti, M.Kom., selaku Dosen Pembimbing, yang telah meluangkan waktu, tenaga, dan pikiran untuk memberikan arahan, bimbingan substantif, dan koreksi yang sangat berarti selama keseluruhan proses penulisan skripsi ini.
5. Seluruh jajaran manajemen dan tim Divisi Teknologi Informasi PT Wijaya Karya (Persero) Tbk, khususnya kepada *Lead Developer* aplikasi WISE/ESCM atas kesediaan meluangkan waktu dalam proses elisitasi



© Hak Cipta milik 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

pengetahuan, memberikan akses ke lingkungan repositori, serta kepercayaan yang telah diberikan kepada penulis selama pelaksanaan penelitian.

6. Kedua orang tua dan seluruh keluarga penulis, yang tidak pernah berhenti memberikan doa, kasih sayang, dukungan moral, dan motivasi yang menjadi fondasi terkuat bagi penulis dalam menempuh dan menyelesaikan studi ini.
7. Rekan-rekan mahasiswa Program Studi Teknik Informatika Politeknik Negeri Jakarta angkatan 2022, atas kebersamaan, diskusi, dan semangat saling mendukung yang tidak ternilai sepanjang masa perkuliahan.

Penulis menyadari sepenuhnya bahwa skripsi ini masih mengandung keterbatasan dan jauh dari kesempurnaan. Kritik dan saran yang bersifat membangun sangat penulis harapkan demi perbaikan karya ini maupun penelitian-penelitian yang akan datang. Semoga skripsi ini dapat memberikan manfaat yang nyata bagi pengembangan ilmu pengetahuan, khususnya di bidang rekayasa perangkat lunak, otomatisasi *quality assurance*, dan penerapan kecerdasan buatan dalam lingkungan pengembangan perangkat lunak industri.

**POLITEKNIK
NEGERI
JAKARTA**

Depok, 7 Juli 2026

Rafif Nuraydin Adipratama

NIM 2207412051



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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS

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

Nama : Rafif Nuraydin Adipratama
NIM : 2207412051
Jurusan/Program Studi : Teknik 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 Pendukung *Code Review* Berbasis Artificial Intelligence Menggunakan *Rulebook* Internal pada Aplikasi WISE/ESCM

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, 7 Juli 2026

Yang menyatakan



Rafif Nuraydin Adipratama

NIM. 2207412051



ABSTRAK

Pemeriksaan kode (*code review*) secara manual pada pengembangan aplikasi perusahaan rentan menimbulkan antrean dan keterlambatan umpan balik ketika jumlah perubahan kode melebihi ketersediaan peninjau. Penelitian ini bertujuan merancang dan membangun sistem pendukung *code review* berbasis Artificial Intelligence (AI) yang memeriksa kode berdasarkan *rulebook* internal pada aplikasi WISE/ESCM milik PT Wijaya Karya (Persero) Tbk. Sistem dikembangkan dengan metode *Waterfall* dan terdiri atas mesin pemeriksa (*review engine*) berbasis Node.js serta antarmuka web. Aturan pemeriksaan disusun menjadi *rulebook* melalui wawancara dengan Lead Developer. Antarmuka web memungkinkan pengguna mengunggah berkas kode PHP secara bebas dan menampilkan hasil pemeriksaan, rincian temuan, riwayat, dan kuesioner *System Usability Scale* (SUS). Pengujian dilakukan melalui *black-box testing*, penilaian pengguna dengan SUS, dan evaluasi kemampuan deteksi berdasarkan validasi pakar menggunakan data uji terkontrol sebagai uji validasi. Penilaian SUS terhadap 15 responden pengembang WISE/ESCM memperoleh skor rata-rata 69,67 berkategori Baik, dengan hambatan teknis dan kebutuhan bantuan tambahan sebagai aspek yang masih perlu disempurnakan. Sistem ini berperan sebagai pemberi pendapat kedua yang tepercaya dan mendukung, bukan menggantikan, peninjau manusia.

Kata Kunci: *Artificial Intelligence, black-box testing, code review*

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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR ISI

SURAT PERNYATAAN BEBAS PLAGIARISME	ii
LEMBAR PENGESAHAN	iii
KATA PENGANTAR.....	iv
SURAT PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS.....	vi
ABSTRAK	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR	xiii
BAB I.....	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah	3
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.....	6
TINJAUAN PUSTAKA.....	6
2.1 Landasan Teori	6
2.1.1 Penjaminan Mutu Perangkat Lunak (<i>Software Quality Assurance</i>).....	6
2.1.2 <i>Code Review</i>	6
2.1.3 <i>Artificial Intelligence</i> dan <i>Large Language Model</i>	6



© Hak Cipta milik 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

2.1.4 <i>Prompt Engineering</i> dalam <i>Code Review</i>	7
2.1.5 Metode Penelitian Kombinasi (<i>Mixed Methods</i>).....	7
2.1.6 Model <i>Waterfall</i>	7
2.1.7 Aplikasi Web dan Arsitektur <i>Client-Server</i>	7
2.1.8 <i>Rulebook</i> dan Pengetahuan Internal.....	8
2.1.9 <i>Technical Debt</i>	8
2.1.10 <i>Quality Gate</i>	8
2.1.11 <i>Black-box Testing</i>	9
2.1.12 <i>User Acceptance Testing</i> dan <i>System Usability Scale</i>	9
2.1.13 OpenAI GPT 4o-mini.....	10
2.2 Penelitian Terdahulu.....	10
BAB III	13
METODE PENELITIAN.....	13
3.1 Rancangan Penelitian.....	13
3.1.1 Tahapan Penelitian	14
3.1.2 Objek Penelitian.....	16
3.2 Sumber Data.....	16
3.2.1 Data Primer	17
3.2.2 Data Sekunder	17
3.3 Teknik Pengumpulan Data	17
BAB IV	22
HASIL DAN PEMBAHASAN	22
4.1 Analisis Kebutuhan	22
4.1.1 Kebutuhan Fungsional	22
4.1.2 Kebutuhan Non-Fungsional	23
4.1.3 Perangkat Pengembangan	23



© Hak Cipta milik 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

4.2 Perancangan Sistem	24
4.2.1 Perancangan Arsitektur dan Logika Sistem	24
4.2.2 Penyusunan Buku Pedoman (<i>Rulebook</i>) melalui Elisitasi Pengetahuan	35
4.2.3 Perancangan Instruksi (<i>Prompt Engineering</i>)	49
4.2.4 Analisis Kebutuhan Antarmuka Web	52
4.3 Implementasi Sistem	53
4.3.1 Implementasi Pemodelan Deteksi AI (<i>Review Engine</i>)	53
4.3.2 Implementasi Antarmuka Web	65
4.4 Pengujian.....	75
4.4.1 Pengujian Fungsional (<i>Black-box Testing</i>).....	75
4.4.2 Penilaian Pengguna dengan System Usability Scale (SUS)	79
4.4.3 Pembahasan Hasil Evaluasi	84
4.4.4 Pengujian Non-Fungsional.....	85
PENUTUP.....	84
5.1 Kesimpulan	84
5.2 Saran.....	85
DAFTAR PUSTAKA.....	87
DAFTAR RIWAYAT HIDUP PENULIS	89
LAMPIRAN.....	90
Lampiran 1 – Dokumentasi Wawancara	90
Lampiran 2 – Hasil Analisis Temuan Analisis Kode Statis pada Repositori WISE/ESCM.....	91
Lampiran 3 – Rincian Gap Analysis 48 Klaim Wawancara vs Implementasi Kode	98
Lampiran 4 – Intruksi Pemodelan Lengkap.....	110



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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR TABEL

Tabel 1. Penelitian Terdahulu.....	11
Tabel 2. Topik Wawancara (Penggalian Aturan Internal yang tak tertulis).....	17
Tabel 3. Perangkat Pengembangan Sistem.....	23
Tabel 4. Ringkasan Temuan Analisis Kode Statis pada Repositori WISE/ESCM	38
Tabel 5. Ringkasan Analisa Hasil Wawancara dengan Basis Kode WISE/ESCM	40
Tabel 6. menjadi isi penuh 30 baris.....	44
Tabel 7. Ringkasan inti instruksi sistem ((Prompt Engineering)	51
Tabel 8. Pemetaan Tingkat Risiko Aturan terhadap Tingkat Keparahatan Temuan AI	58
Tabel 9. Hasil Black-box Testing (10 Skenario)	76
Tabel 10. Skala Penilaian SUS.....	79
Tabel 11. Daftar Pernyataan Instrumen SUS	80
Tabel 12. Kategori Interpretasi Skor SUS.....	81
Tabel 13. Rekapitulasi Skor SUS Responden	83

**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 mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR GAMBAR

Gambar 3. 1 Tahapan Penelitian	14
Gambar 4. 1 Rancangan Pemodelan Deteksi AI Berbasis Ekstraksi Pengetahuan	26
Gambar 4. 2 <i>Use Case Diagram</i>	28
Gambar 4. 3 <i>Flowchart</i> Sistem	30
Gambar 4. 4 Rancangan Halaman Utama dan Area Unggah	31
Gambar 4. 5 Rancangan Halaman Tampilan Proses	32
Gambar 4. 6 Rancangan Halaman Ringkasan dan Rincian Laporan	33
Gambar 4. 7 Tampilan Riwayat Run	33
Gambar 4. 8 Alur Data <i>Review Engine</i>	35
Gambar 4.9 Cuplikan kode proses perubahan kode menjadi <i>synthetic diff</i>	55
Gambar 4.10 Kode proses pemecahan kode menjadi beberapa bagian	56
Gambar 4.11 Kode struktur keluaran AI pada <i>reviewSchema</i>	57
Gambar 4.12 Kode proses penyaringan hasil dan penyesuaian <i>severity</i>	62
Gambar 4.13 Kode fungsi <i>reviewSource</i> sebagai pengendali utama <i>review engine</i>	65
Gambar 4.14 Kode pengaturan alur utama antarmuka web	67
Gambar 4.15 Tampilan Area Unggah Berkas Kode PHP	67
Gambar 4.16 Kode pengiriman file dan pembacaan <i>streaming progress</i>	69
Gambar 4.17 Kode komponen <i>progress scan</i>	70
Gambar 4.18 Tampilan Progres Pemindaian	71
Gambar 4.19 Kode tampilan laporan hasil <i>review</i>	72
Gambar 4.20 Tampilan Antarmuka Laporan Interaktif Hasil Tinjauan Kode	73
Gambar 4.21 Kode tampilan <i>rulebook</i> dan riwayat pemeriksaan	74
Gambar 4.22 Tampilan Antarmuka Daftar <i>Rulebook</i> dan Riwayat Pemeriksaan.	75
Gambar 4. 23 Tampilan Hasil <i>performance test</i> pada laman <i>pagespeed.web.dev</i>	85



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pengembangan perangkat lunak di perusahaan menuntut proses kerja yang cepat dan rapi, tetapi mutu kode tetap harus dijaga. Salah satu kegiatan penting untuk menjaga mutu kode adalah *code review*, yaitu pemeriksaan setiap perubahan kode sebelum digabungkan ke kode utama. Davila dan Nunes (2021) menjelaskan bahwa *code review* sudah menjadi praktik umum yang membantu menemukan kesalahan, membuat kode lebih mudah dibaca, dan menjaga kode tetap sesuai standar pengembangan.

Pada praktiknya, melakukan *review* pada ribuan kode secara manual bisa menjadi suatu penghambat. Ini terjadi ketika jumlah perubahan kode yang masuk lebih banyak daripada jumlah orang yang bisa memeriksanya. Davila dan Nunes (2021) mencatat bahwa beban kerja dan waktu peninjauan menjadi salah satu tantangan utama dalam *review* kode. Hal serupa juga terjadi pada pengembangan aplikasi WISE/ESCM di PT Wijaya Karya (Persero) Tbk. Aplikasi ini bukan hanya mengikuti aturan umum *framework* Laravel, tetapi juga memiliki banyak aturan internal sendiri, seperti penggunaan *framework* internal bernama Citadel, cara khusus memeriksa data masukan, aturan hak akses berdasarkan peran pengguna, dan alur persetujuan (*approval*). Sebagian aturan ini belum terdokumentasi dengan lengkap, sehingga pemeriksaan kode masih sangat bergantung pada pengalaman pribadi *Lead Developer*.

Di sisi lain, perkembangan *Large Language Model* (LLM) membuka peluang untuk membantu proses *code review*. Hou dkk. (2024), dalam tinjauan pustaka sistematisnya, menunjukkan bahwa LLM sudah banyak dimanfaatkan untuk berbagai tugas rekayasa perangkat lunak, termasuk pemeriksaan kode. Pornprasit dan Tantithamthavorn (2024) secara khusus menyatakan bahwa penyusunan instruksi, atau *prompt engineering*, bisa menjadi cara yang tepat untuk mengotomatisasi *code review*, terutama bagi organisasi yang tidak punya sumber daya untuk melatih ulang model AI sendiri.



© Hak Cipta milik 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Meski begitu, penggunaan AI dalam *code review* bagusnya dibatasi dengan hati-hati. Huang dkk. (2024) menjelaskan bahwa LLM bisa menghasilkan jawaban yang terlihat meyakinkan, padahal sebenarnya salah. Kondisi ini disebut *hallucination*. Dalam konteks pemeriksaan kode, kesalahan seperti ini bisa berupa temuan yang menunjuk ke bagian kode yang sebenarnya tidak ada, atau saran yang tidak sesuai dengan aturan yang berlaku. Karena itu, sistem yang dibangun perlu diberi aturan yang jelas dan penyaring, supaya hasil temuannya tetap bisa dipercaya.

Berdasarkan hal-hal tersebut, penelitian ini merancang dan membangun sebuah sistem pendukung *code review* berbasis Artificial Intelligence (AI) untuk aplikasi WISE/ESCM. Sistem ini menggunakan sebuah *rulebook* internal sebagai dasar pemeriksaan. Dengan begitu, saran yang diberikan tidak hanya mengikuti aturan umum pemrograman, tetapi juga aturan internal yang berlaku khusus di aplikasi ini. Sistem ini tidak dirancang untuk menggantikan peninjau manusia, melainkan untuk membantu peninjau dengan memberi saran awal, supaya pemeriksaan kode menjadi lebih terstruktur dan konsisten. Selain memeriksa kode, sistem ini juga dilengkapi antarmuka web yang memungkinkan pengguna mengunggah file kode PHP secara bebas, melihat hasil pemeriksaan, rincian temuan, riwayat pemeriksaan, dan formulir penilaian pengguna. Antarmuka web ini penting supaya cara kerja sistem bisa diperagakan dan dinilai dengan lebih mudah, baik oleh dosen penguji maupun pengguna terbatas lainnya.

1.2 Perumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana mengidentifikasi kebutuhan sistem pendukung *code review* berbasis *Artificial Intelligence* pada aplikasi WISE/ESCM?
2. Bagaimana merancang dan membangun sistem pendukung *code review* berbasis *Artificial Intelligence* yang mampu memeriksa kode berdasarkan *rulebook* internal?
3. Bagaimana hasil pengujian fungsional sistem pendukung *code review* menggunakan metode *black-box testing*?



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

4. Bagaimana tingkat penerimaan dan kemudahan penggunaan sistem berdasarkan penilaian pengguna menggunakan instrumen *System Usability Scale*?

1.3 Batasan Masalah

Agar penelitian ini lebih terarah dan sesuai ruang lingkupnya, batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Penelitian dilakukan pada studi kasus aplikasi WISE/ESCM di lingkungan PT Wijaya Karya (Persero) Tbk, dengan fokus pada pemeriksaan kode berdasarkan kebutuhan dan aturan internal pengembangan aplikasi.
2. Sistem yang dibangun merupakan purwarupa sistem pendukung *code review* berbasis *Artificial Intelligence*. Sistem hanya memberi saran atau hasil pemeriksaan kode dan tidak mengubah kode secara otomatis (*advisory-only*).
3. Aturan pemeriksaan kode disusun menjadi *rulebook* internal berdasarkan hasil wawancara dengan *Lead Developer* dan penelaahan kebutuhan pengembangan aplikasi WISE/ESCM.
4. Sistem menerima masukan berupa file kode berformat .php yang diunggah pengguna secara bebas melalui antarmuka web. Evaluasi kemampuan deteksi sistem terhadap *rulebook* dilakukan secara terpisah, menggunakan data uji terkontrol sebagai uji validasi (Saia dkk., 2022).
5. Pengujian sistem dibatasi pada pengujian fungsional menggunakan *black-box testing*, serta penilaian penerimaan dan kemudahan penggunaan menggunakan instrumen *System Usability Scale*. Penelitian ini tidak mencakup penerapan penuh sistem pada lingkungan produksi harian perusahaan.

1.4 Tujuan dan Manfaat

1.4.1 Tujuan Penelitian

Tujuan dari pelaksanaan penelitian skripsi ini adalah:



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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

1. Mengidentifikasi kebutuhan sistem pendukung *code review* berbasis *Artificial Intelligence* pada aplikasi WISE/ESCM.
2. Merancang dan membangun sistem pendukung *code review* berbasis *Artificial Intelligence* yang mampu memeriksa kode berdasarkan *rulebook* internal.
3. Menguji fungsi sistem pendukung *code review* menggunakan metode *black-box testing*.
4. Menilai tingkat penerimaan dan kemudahan penggunaan sistem menggunakan instrumen *System Usability Scale*.

1.4.2 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Bagi perusahaan, penelitian ini menyediakan purwarupa sistem pendukung *code review* yang dapat membantu memberi saran awal terhadap perubahan kode berdasarkan aturan internal.
2. Bagi pengembang, sistem ini dapat membantu memahami aturan pengembangan aplikasi WISE/ESCM melalui hasil pemeriksaan yang ditampilkan secara lebih terstruktur.
3. Bagi *Lead Developer*, sistem ini membantu mendokumentasikan sebagian aturan internal ke dalam bentuk *rulebook*, sehingga pemeriksaan kode tidak hanya bergantung pada ingatan atau pengalaman seseorang.
4. Bagi penguji dan pengguna terbatas, antarmuka web yang dibangun membantu memahami cara kerja sistem karena hasil pemeriksaan ditampilkan dalam bentuk yang lebih mudah diamati.
5. Bagi akademik, penelitian ini memberi contoh penerapan AI pada sistem pendukung *code review* yang dikembangkan dengan metode *Waterfall* dan diuji menggunakan gabungan *black-box testing* dan *System Usability Scale*.

1.5 Sistematika Penulisan

Laporan skripsi ini disusun dengan sistematika sebagai berikut:

1. BAB I PENDAHULUAN



© Hak Cipta milik 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Bab ini menguraikan latar belakang, perumusan masalah, batasan masalah, tujuan dan manfaat penelitian, serta sistematika penulisan.

2. BAB II TINJAUAN PUSTAKA

Bab ini berisi landasan teori yang mendukung penelitian serta tinjauan terhadap penelitian terdahulu yang relevan.

3. BAB III METODE PENELITIAN

Bab ini menjelaskan rancangan penelitian dengan metode *Waterfall*, objek penelitian, sumber data, teknik pengumpulan data, kebutuhan sistem, perancangan sistem, dan rancangan pengujian.

4. BAB IV HASIL DAN PEMBAHASAN

Bab ini memaparkan hasil analisis kebutuhan, perancangan sistem, implementasi sistem dan antarmuka web, hasil *black-box testing*, hasil penilaian pengguna dengan SUS, serta evaluasi kemampuan deteksi berdasarkan validasi pakar.

5. BAB V PENUTUP

Bab ini berisi kesimpulan, saran, dan kontribusi penelitian.

**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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 Penjaminan Mutu Perangkat Lunak (*Software Quality Assurance*)

Penjaminan mutu perangkat lunak atau *Software Quality Assurance* (SQA) adalah kegiatan untuk memastikan bahwa proses pengembangan menghasilkan produk yang sesuai dengan kebutuhan dan standar mutu. Salah satu praktik penting dalam menjaga mutu kode adalah *code review*, yaitu pemeriksaan perubahan kode sebelum digabungkan ke kode utama. Davila dan Nunes (2021) menunjukkan bahwa *code review* merupakan praktik mutu yang banyak diterapkan untuk menemukan kesalahan lebih awal dan menjaga kesesuaian kode dengan standar pengembangan.

2.1.2 *Code Review*

Code review adalah pemeriksaan perubahan kode oleh peninjau sebelum kode digabungkan ke kode utama. Davila dan Nunes (2021) dalam tinjauan pustakanya menjelaskan bahwa tujuan utama *code review* meliputi penemuan kesalahan, peningkatan keterbacaan kode, dan menjaga kesesuaian kode dengan aturan pengembangan, sekaligus mencatat bahwa beban dan waktu peninjauan menjadi tantangan tersendiri.

2.1.3 *Artificial Intelligence* dan *Large Language Model*

Artificial Intelligence (AI) adalah bidang ilmu komputer yang berfokus pada pembuatan sistem yang mampu mengerjakan tugas yang biasanya membutuhkan kecerdasan manusia. Salah satu bentuk AI yang berkembang pesat adalah *Large Language Model* (LLM). Hou dkk. (2024) dalam tinjauan pustaka sistematisnya mencatat bahwa LLM telah dimanfaatkan pada beragam tugas rekayasa perangkat lunak, termasuk pembuatan, perbaikan, dan pemeriksaan kode.



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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

2.1.4 *Prompt Engineering* dalam *Code Review*

Prompt engineering adalah teknik menyusun instruksi agar model AI menghasilkan keluaran yang sesuai dengan kebutuhan. Sahoo dkk. (2024) menjelaskan bahwa *prompt engineering* mencakup berbagai teknik penyusunan instruksi, termasuk pemberian peran, konteks, dan contoh, untuk mengarahkan keluaran model tanpa mengubah model itu sendiri.

2.1.5 Metode Penelitian Kombinasi (*Mixed Methods*)

Metode penelitian pada dasarnya dikelompokkan menjadi metode kualitatif dan metode kuantitatif. Sugiyono (2019) menjelaskan bahwa ketika suatu penelitian membutuhkan pemahaman mendalam atas suatu fenomena (data kualitatif) sekaligus pengukuran yang objektif dan teruji (data kuantitatif), kedua pendekatan tersebut dapat dipadukan menjadi metode kombinasi (*mixed methods*). Creswell (2014) menambahkan bahwa pendekatan kombinasi memberikan pemahaman yang lebih utuh dibandingkan hanya menggunakan salah satu pendekatan saja, terutama pada penelitian yang menghasilkan suatu artefak yang pengetahuannya perlu digali dari pakar sekaligus diuji kinerjanya secara terukur.

2.1.6 Model *Waterfall*

Metode *Waterfall* adalah model pengembangan perangkat lunak yang menyusun proses pengembangan secara berurutan, meliputi analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan. Saravanos dan Curinga (2023) menjelaskan bahwa model *Waterfall* dapat menggambarkan siklus hidup pengembangan perangkat lunak melalui fase-fase yang saling berurutan, sehingga membantu memperkirakan alur pengerjaan, kebutuhan sumber daya, serta kemungkinan hambatan pada tiap fase.

2.1.7 Aplikasi Web dan Arsitektur *Client-Server*

Menurut Saia SM (2022) Aplikasi web adalah aplikasi yang dapat diakses melalui peramban atau *browser* dan umumnya terdiri atas sisi tampilan (*frontend*) dan sisi *server* (*backend*). Arsitektur *client-server* memisahkan tanggung jawab antara sisi pengguna dan sisi *server*, sehingga antarmuka dapat dikembangkan tanpa mencampur seluruh logika di sisi peramban.



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Pemisahan ini juga menjaga keamanan, yaitu kunci API dan proses pemanggilan model AI tetap berada di sisi *server*. Dokumentasi resmi OpenAI (2024) menyarankan agar kunci API tidak diletakkan pada sisi peramban atau aplikasi sisi klien untuk mencegah penyalahgunaan. Oleh karena itu, dalam penelitian ini peramban hanya menerima hasil yang sudah diolah, sedangkan logika pemeriksaan dan pemanggilan model dijalankan oleh *backend* dan *review engine*.

2.1.8 Rulebook dan Pengetahuan Internal

Dalam pengembangan perangkat lunak perusahaan, tidak semua aturan tertulis lengkap pada dokumentasi resmi. Sebagian aturan berupa kebiasaan tim atau pola penataan kode yang dipahami pengembang senior. Pengetahuan seperti ini dikenal sebagai aturan tak tertulis atau *tacit knowledge*. Fenoglio dkk. (2022) menjelaskan bahwa aturan tak tertulis dari pakar dapat dikumpulkan melalui penggalian pengetahuan atau *knowledge elicitation*, kemudian diubah ke dalam bentuk yang dapat disimpan dan digunakan kembali.

2.1.9 Technical Debt

Technical debt adalah kondisi ketika keputusan teknis yang diambil untuk mempercepat pengembangan menimbulkan beban perawatan di kemudian hari. Albuquerque dkk. (2022) menunjukkan bahwa *technical debt* dipahami oleh banyak praktisi sebagai masalah yang berdampak pada berbagai bagian perangkat lunak, termasuk kode, desain, dan arsitektur, serta dapat muncul dari kode yang sulit dipahami, penggandaan logika, atau pengujian yang kurang memadai.

2.1.10 Quality Gate

Quality gate adalah mekanisme pemeriksaan yang ditempatkan pada alur pengembangan untuk memastikan bahwa perubahan kode telah memenuhi standar tertentu sebelum dilanjutkan ke tahap berikutnya. Gluhak dan Pavlič (2022) menjelaskan bahwa *quality gate* dalam alur pengiriman perangkat lunak dapat diwujudkan melalui analisis kode otomatis sebagai bentuk kontrol mutu.



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

2.1.11 *Black-box Testing*

Black-box testing adalah metode pengujian perangkat lunak yang berfokus pada masukan dan keluaran sistem tanpa memeriksa struktur kode program di dalamnya. Kapoh dkk. (2021) menggunakan *black-box testing* pada aplikasi web untuk memastikan bahwa fungsi sistem menghasilkan keluaran sesuai yang diharapkan.

Dalam mengevaluasi efektivitas skenario uji fungsional yang dijalankan, tingkat keberhasilannya perlu diukur secara kuantitatif. Mengacu pada pedoman pengujian oleh Maulida dkk. (2025), persentase keberhasilan pengujian *black-box* dihitung dengan membandingkan jumlah skenario yang memberikan hasil valid (sesuai yang diharapkan) terhadap keseluruhan total skenario pengujian yang dilakukan. Rumus perhitungannya adalah sebagai berikut:

$$\text{Tingkat Keberhasilan} = \frac{\text{Jumlah Skenario}}{\text{Total Skenario Uji}} \times 100 \%$$

Semakin mendekati persentase 100%, maka perangkat lunak tersebut dapat disimpulkan telah berfungsi dengan baik dan siap digunakan oleh pengguna akhir.

2.1.12 *User Acceptance Testing dan System Usability Scale*

Pengujian penerimaan pengguna atau *User Acceptance Testing* (UAT) bertujuan mengetahui apakah sistem dapat diterima pengguna berdasarkan kebutuhan yang ditetapkan. Pada penelitian ini, penilaian pengguna dilakukan menggunakan instrumen *System Usability Scale* (SUS), yaitu kuesioner kebergunaan yang terdiri atas 10 pernyataan dengan skala 1 sampai 5.

Skor akhir SUS tidak didapatkan dengan menjumlahkan nilai mentah secara langsung. Berdasarkan standar perhitungan evaluasi *usability* yang dikaji oleh Firmansyah dkk. (2024), terdapat prosedur perhitungan khusus untuk menentukan skor kontribusi pada tiap item kuesioner. Aturan perhitungannya adalah:

1. Untuk pernyataan bernomor ganjil (pernyataan positif), skor kontribusi dihitung dengan mengurangi nilai 1 dari skor jawaban responden.
2. Untuk pernyataan bernomor genap (pernyataan negatif), skor kontribusi dihitung dengan mengurangi nilai jawaban responden dari angka 5.



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

3. Total skor kontribusi dari 10 pernyataan kemudian dijumlahkan dan dikalikan mb

Secara matematis, rumus perhitungan skor SUS tersebut dituliskan sebagai berikut:

$$\text{Skor SUS} = ((\text{Skor Ganjil} - 1) + (5 - \text{Skor Genap})) \times 2,5$$

Skor akhir rata-rata kemudian dicocokkan dengan kategori tingkat penerimaan (*acceptability ranges*) untuk menentukan apakah sistem tersebut masuk ke dalam kategori tidak dapat diterima (*not acceptable*), batas marginal (*marginal*), atau dapat diterima (*acceptable*).

2.1.13 OpenAI GPT 4o-mini

GPT-4o-mini adalah model AI (*Large Language Model*) buatan OpenAI yang dirilis pada Juli 2024, sebagai versi yang lebih kecil, lebih cepat, dan lebih murah dibandingkan model utamanya, GPT-4o. Meski ukurannya lebih kecil, GPT-4o-mini tetap mampu memahami dan menghasilkan teks, termasuk kode program, dengan kualitas yang cukup baik untuk berbagai jenis tugas. Model ini mendukung *context window* (batas jumlah teks yang bisa dibaca sekaligus dalam satu kali permintaan) sebesar 128.000 token, sehingga cukup untuk memproses potongan kode program yang panjang dalam satu kali pemanggilan. Dangi dkk. (2025) menunjukkan bahwa GPT-4o-mini tetap bisa diandalkan untuk tugas analisis berbasis teks dan gambar, meskipun performanya berada di bawah model utama GPT-4o. Karena biaya penggunaannya jauh lebih murah dibandingkan model yang lebih besar, GPT-4o-mini menjadi pilihan yang sesuai untuk sistem yang perlu memeriksa banyak kode secara berulang-ulang, seperti pada *review engine* yang dibangun dalam penelitian ini. (Dangi et al., 2025)

2.2 Penelitian Terdahulu

Penelitian mengenai pemanfaatan AI pada *code review*, otomatisasi pengembangan perangkat lunak, dan evaluasi aplikasi web telah banyak dilakukan sebelumnya. Tinjauan terhadap penelitian terdahulu dilakukan untuk memetakan posisi



© Hak Cipta milik 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

penelitian ini dan membedakannya dengan penelitian sebelumnya. Ringkasan penelitian terdahulu ditampilkan pada Tabel 1.

Tabel 1. Penelitian Terdahulu

No	Peneliti (Tahun)	Fokus Penelitian	Hasil Penelitian
1	Kapoh dkk. (2021)	<i>Black-box testing</i> pada aplikasi web	Pengujian memastikan fungsi aplikasi web berjalan sesuai keluaran yang diharapkan
2	Davila & Nunes (2021)	Tinjauan pustaka sistematis tentang <i>code review</i> modern	Memetakan tujuan, praktik, dan tantangan <i>code review</i> , termasuk beban dan waktu peninjauan
3	Fenoglio dkk. (2022)	Proses penggalan <i>tacit knowledge</i>	Pengetahuan pakar dapat dikumpulkan dan diubah menjadi bentuk yang sudapat digunakan kembali
4	Saravanos & Curinga (2023)	Simulasi siklus hidup pengembangan dengan model <i>Waterfall</i>	Model <i>Waterfall</i> dapat menggambarkan fase pengembangan secara berurutan
5	Pornprasit & Tantithamthavorn (2024)	<i>Prompt engineering</i> dan <i>fine-tuning</i> untuk otomatisasi <i>code review</i> berbasis LLM	<i>Prompt engineering</i> dapat memandu model dalam tugas <i>code review</i> tanpa selalu memerlukan <i>fine-tuning</i>



© Hak Cipta milik 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

No	Peneliti (Tahun)	Fokus Penelitian	Hasil Penelitian
6	Perrig dkk. (2024)	Praktik pengukuran dalam penelitian pengalaman pengguna	Penelitian berbasis skala perlu melaporkan instrumen dan cara analisis secara jelas

Sumber: Mendeley, diolah oleh penulis

Berdasarkan Tabel 1., penelitian terdahulu umumnya membahas satu aspek secara terpisah, seperti pengujian aplikasi web, penggalian pengetahuan, metode pengembangan, atau pemanfaatan LLM pada *code review*. Posisi penelitian ini adalah memadukan aspek-aspek tersebut, yaitu menyusun *rulebook* internal dari pengetahuan *Lead Developer*, menerapkan AI dengan pendekatan *prompt engineering*, membangun antarmuka web, serta melakukan evaluasi yang menggabungkan *black-box testing*, *System Usability Scale*, dan evaluasi kemampuan deteksi berbasis validasi pakar. Penggabungan inilah yang membedakan penelitian ini dari penelitian terdahulu.

**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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

BAB III

METODE PENELITIAN

3.1 Rancangan Penelitian

Penelitian ini menggunakan metode penelitian kombinasi (*mixed methods*), yang menggabungkan pendekatan kualitatif dan kuantitatif (Sugiyono, 2019). Pendekatan kualitatif digunakan pada tahap *knowledge elicitation* melalui wawancara mendalam dengan *Lead Developer*, analisis tema dari hasil wawancara, serta *gap analysis* yang mengelompokkan kesesuaian temuan ke dalam tiga status, sesuai, tidak sesuai, atau parsial. Pendekatan kuantitatif digunakan pada tahap pengujian, yaitu perhitungan skor *System Usability Scale* (SUS) dari penilaian pengguna, serta hasil *black-box testing*. Data kualitatif digunakan untuk menyusun *rulebook* sebagai dasar pengetahuan sistem, sedangkan data kuantitatif digunakan untuk mengevaluasi kinerja sistem yang dibangun dari *rulebook* tersebut.

Untuk mewujudkan sistem ini, penelitian menggunakan metode pengembangan sistem (*System Development Life Cycle/SDLC*) *Waterfall*, dengan tahapan analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan (Saravanos & Curinga, 2023). Tahap pemeliharaan tidak dibahas sebagai tahap operasional jangka panjang, karena sistem ini masih berada pada tahap purwarupa dan belum diterapkan sebagai sistem produksi harian di perusahaan.

Secara keseluruhan, penelitian ini tidak langsung dimulai dari tahap pengembangan sistem. Penelitian diawali dengan identifikasi masalah dan pengumpulan data, sebelum masuk ke tahap pengembangan sistem menggunakan metode *Waterfall*, dan diakhiri dengan penulisan laporan. Rincian setiap tahapan dijelaskan pada Subbab 3.1.1.

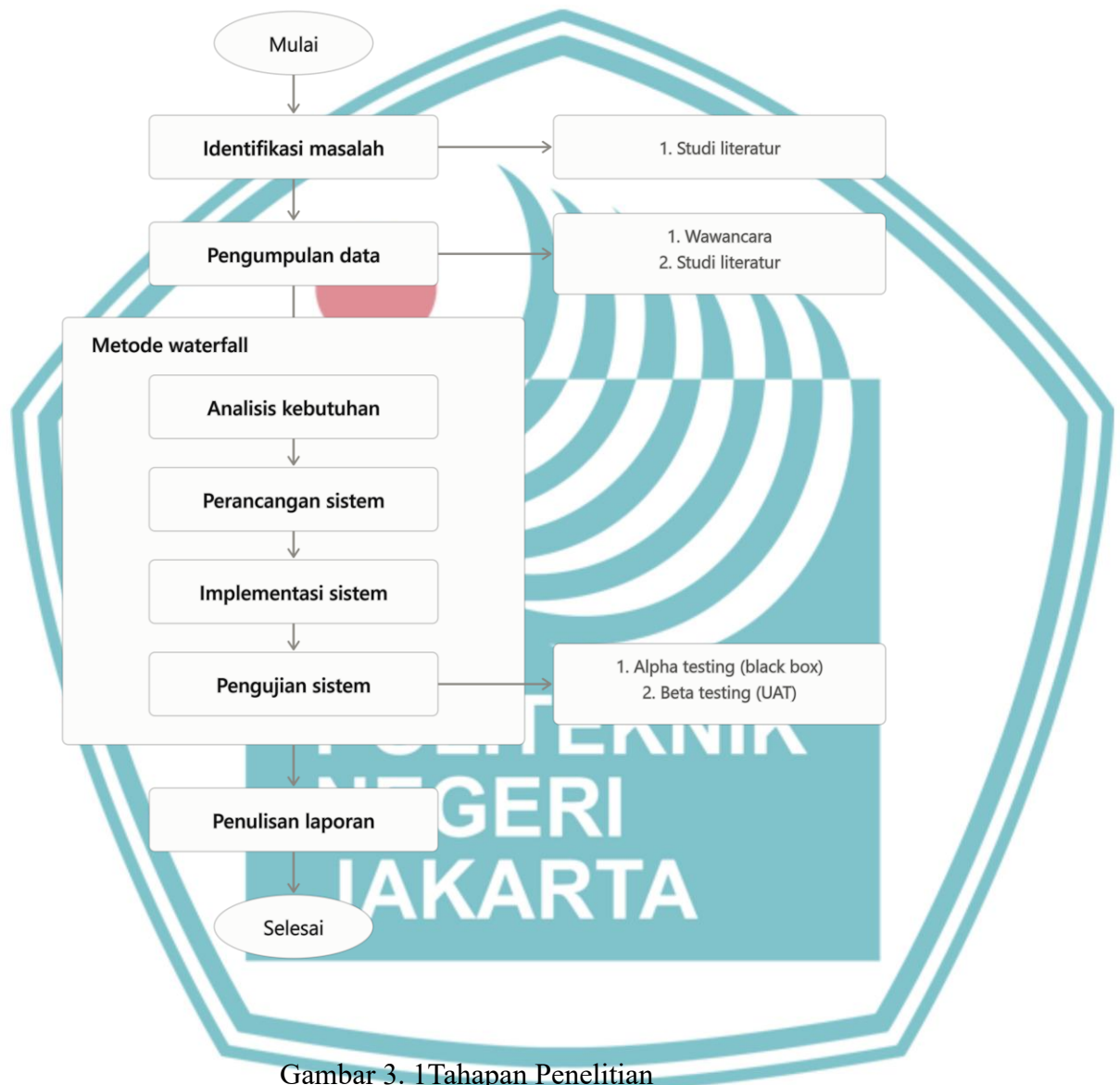


Hak Cipta :

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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

3.1.1 Tahapan Penelitian

Tahapan penelitian ini, mulai dari identifikasi masalah hingga penulisan laporan, ditunjukkan pada Gambar 3.1 dan dijelaskan sebagai berikut.



Gambar 3. 1 Tahapan Penelitian

Sumber: Sugiyono (2019) dan Saravanos & Curinga (2023), diolah oleh penulis

1. Identifikasi Masalah. Tahap ini dilakukan untuk menelaah masalah yang melatarbelakangi penelitian, yaitu proses *code review* manual pada aplikasi WISE/ESCM di PT Wijaya Karya (Persero) Tbk yang menjadi kurang efisien. Penyebabnya adalah aturan internal, seperti *framework* Citadel, cara khusus memeriksa data masukan, dan alur persetujuan, yang belum



© Hak Cipta milik 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

terdokumentasi lengkap, sehingga pemeriksaan kode masih bergantung pada pengalaman *Lead Developer* (Subbab 1.1). Masalah ini kemudian dirumuskan lebih lanjut menjadi kebutuhan membangun sistem pendukung *code review* berbasis Artificial Intelligence yang mengacu pada *rulebook* internal (Subbab 1.2). Tahap ini didukung dengan studi literatur tentang *code review*, *Large Language Model*, dan risiko *hallucination* pada penerapan AI.

2. Pengumpulan Data. Tahap ini dilakukan melalui wawancara dengan *Lead Developer* untuk menggali aturan internal pengembangan aplikasi WISE/ESCM, serta studi literatur untuk memperoleh dasar teori pendukung. Rincian sumber dan teknik pengumpulan data dijelaskan pada Subbab 3.2 dan 3.3.
3. Analisis Kebutuhan. Tahap ini dilakukan untuk memahami masalah pada proses *code review* aplikasi WISE/ESCM. Kebutuhan sistem diperoleh melalui pengamatan alur pemeriksaan kode, wawancara dengan *Lead Developer*, dan penelaahan terhadap aturan pemeriksaan kode yang diperlukan.
4. Perancangan Sistem. Tahap ini menyusun rancangan arsitektur sistem, mencakup *review engine*, *backend* dan *frontend* antarmuka web, alur pemeriksaan kode, penyajian hasil, serta rancangan pengujian.
5. Implementasi Sistem. Tahap ini membangun sistem berdasarkan rancangan. Implementasi mencakup pengembangan *review engine* berbasis Node.js, pemanggilan OpenAI API, antarmuka web berbasis React dan Vite, *backend* Express.js, fitur unggah berkas kode (*drag-and-drop* dan *browse*), pemeriksaan kode, riwayat pemeriksaan, dan fitur penilaian pengguna dengan SUS.
6. Pengujian Sistem. Tahap ini memastikan sistem berjalan sesuai kebutuhan, terdiri atas *Alpha Testing* berupa *black-box testing* untuk menguji fungsi sistem, serta *Beta Testing* berupa penilaian pengguna dengan *User Acceptance Testing* (UAT) menggunakan instrumen SUS untuk mengukur penerimaan dan kemudahan penggunaan, serta validasi pakar untuk



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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

menganalisis kemampuan deteksi. Juga membahas keterbatasan sistem dan hasil SUS.

7. Penulisan Laporan. Tahap ini menyusun laporan hasil penelitian secara sistematis berdasarkan seluruh tahapan yang telah dilaksanakan, mulai dari identifikasi masalah hingga hasil pengujian sistem.

3.1.2 Objek Penelitian

Objek penelitian ini merupakan hasil identifikasi masalah pada Subbab 3.1.1, yaitu sistem pendukung *code review* berbasis *Artificial Intelligence* pada aplikasi WISE/ESCM. Karakteristik objek penelitian adalah sebagai berikut.

1. Sistem: purwarupa sistem pendukung *code review* berbasis AI.
2. Fungsi utama: memeriksa kode berdasarkan *rulebook* internal dan memberi saran pemeriksaan.
3. Masukan: berkas kode PHP yang diunggah pengguna secara bebas melalui antarmuka web dengan cara *drag-and-drop* atau menelusuri berkas (*browse*), dibatasi hanya berekstensi *.php*. Untuk kebutuhan evaluasi kemampuan deteksi, digunakan pula data uji terkontrol berupa kode PHP yang telah disisipi pelanggaran *rulebook* secara sengaja sebagai uji validasi (Saia et al., 2022).
4. Proses inti: sistem mengirim kode uji ke *review engine*, memanggil model AI melalui OpenAI API, memvalidasi hasil, lalu menyusun laporan temuan.
5. Keluaran: ringkasan hasil pemeriksaan, daftar temuan, tingkat keparahan (*severity*), asal aturan, potongan kode terkait, dan riwayat pemeriksaan.
6. Antarmuka: antarmuka web untuk peragaan, pemeriksaan ulang, tampilan hasil, riwayat pemeriksaan, dan pengisian penilaian pengguna dengan SUS.

3.2 Sumber Data

Sumber data pada tahap pengumpulan data yang telah dijelaskan pada Subbab 3.1.1 terdiri atas data primer dan data sekunder.



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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

3.2.1 Data Primer

Data primer diperoleh dari kegiatan pengamatan, wawancara, data uji, validasi pakar, dan hasil penilaian pengguna. Data primer yang digunakan meliputi:

1. Hasil pengamatan terhadap alur *code review* pada aplikasi WISE/ESCM.
2. Hasil wawancara dengan *Lead Developer* terkait aturan internal pengembangan aplikasi.
3. *Rulebook* pemeriksaan kode yang disusun dari hasil wawancara dan analisa kode pada repositori.
4. Data uji berupa contoh pelanggaran yang disiapkan untuk menguji kemampuan sistem.
5. Hasil validasi pakar sebagai knowledge AI.
6. Hasil penilaian pengguna dengan instrumen SUS yang diisi melalui antarmuka web.

3.2.2 Data Sekunder

Data sekunder diperoleh dari jurnal ilmiah, prosiding, buku, dokumentasi teknis, dan rujukan akademik lain yang relevan dengan *code review*, *Artificial Intelligence*, *Waterfall*, aplikasi web, *black-box testing*, SUS, dan evaluasi sistem. Dokumentasi teknis yang dirujuk mencakup dokumentasi OpenAI API (OpenAI, 2024), React (React, 2024), dan Docker.

3.3 Teknik Pengumpulan Data

Merujuk pada tahap pengumpulan data pada Subbab 3.1.1, teknik pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut.

1. Pengamatan dilakukan untuk memahami alur *code review* serta kendala yang muncul pada pemeriksaan kode manual.
2. Wawancara dilakukan dengan *Lead Developer* untuk menggali aturan internal pengembangan aplikasi WISE/ESCM. Hasil wawancara menjadi dasar penyusunan *rulebook*. Wawancara ditunjukkan pada Tabel 3.

Tabel 3. Topik Wawancara (Penggalian Aturan Internal yang tak tertulis)



© H

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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

No	Topik	Tujuan Pertanyaan	Pertanyaan Kunci kepada <i>Lead Developer</i>	Referensi
1	Arsitektur dan komponen utama	Memberikan gambaran menyeluruh (<i>bird's-eye view</i>) arsitektur sistem sebelum masuk ke detail teknis	Bagaimana gambaran besar arsitektur sistem WISE/ESCM beserta komponen utama penyusunnya?	Bass, Clements & Kazman (2012); Sommerville (2011)
2	Framework internal Citadel	Memahami alasan dan cara kerja Citadel karena berada di luar ekosistem standar Laravel	Apa itu <i>framework</i> Citadel, mengapa dibangun sendiri, dan bagaimana alur (<i>lifecycle</i>) halamannya?	Fenoglio dkk. (2022)
3	Konvensi basis data dan model	Menjaga integritas data dan memitigasi risiko <i>mass-assignment vulnerability</i>	Bagaimana standar penulisan model pada proyek ini, termasuk alasan penggunaan \$guarded dibanding	OWASP Mass Assignment Cheat Sheet (OWASP Foundation)



© Hak Cipta milik 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

			\$fillable, serta <i>trait</i> apa saja yang wajib disertakan?	
	Konvensi pemeriksaan masukan	Mengetahui penyimpangan dari <i>best practice</i> standar Laravel agar tidak sembarangan "diperbaiki"	Mengapa pemeriksaan masukan dilakukan langsung di <i>controller</i> dan tidak dipisah menggunakan kelas Form Request standar Laravel?	OWASP Input Validation Cheat Sheet (OWASP Foundation)
5	Keamanan dan hak akses	Mencegah celah keamanan akibat kelalaian memvalidasi hak akses di tingkat <i>controller</i>	Bagaimana model hak akses berbasis peran (<i>Role-Based Access Control</i>) diterapkan, dan apakah pemeriksaan hak akses wajib diletakkan di setiap awal <i>method controller</i> ?	Ferraiolo & Kuhn (1992)



© Hak Cipta milik 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Isolasi dan cakupan data	Mencegah kebocoran data antarproyek yang berdampak fatal pada sistem pengadaan	Bagaimana aturan pembatasan cakupan data antar-pengguna atau antar-entitas pada aplikasi?	Bezemer & Zaidman (2010)
Alur persetujuan (<i>approval workflow</i>)	Memahami mekanisme <i>approval</i> karena tergolong fitur paling kompleks dan sulit ditelusuri jika bermasalah	Bagaimana mekanisme alur persetujuan (<i>approval</i>) bekerja secara arsitektural, dan bagaimana otoritas penyetuju diperiksa?	van der Aalst & van Hee (2002)
Integrasi pihak ketiga	Memahami risiko operasional apabila sinkronisasi dengan ekosistem perusahaan (SAP, HCMS) mengalami kegagalan	Bagaimana arsitektur layanan sinkronisasi (<i>sync service</i>) untuk sistem eksternal seperti SAP dan HCMS?	Hohpe & Woolf (2003)



© Hak Cipta milik 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Pencatatan dan jejak audit	Berkaitan dengan kepatuhan (IT <i>compliance</i>) serta investigasi insiden atau kecurangan	Data atau aktivitas apa saja yang wajib dicatat dalam jejak audit, terutama untuk perubahan yang bersifat finansial atau penting?	Kent & Souppaya (2006), NIST SP 800-92
10 <i>Tacit knowledge</i> dan <i>onboarding</i>	Menangkap kebiasaan atau aturan tak tertulis yang jarang tercantum dalam dokumentasi resmi	Apa lima hal terpenting yang wajib dipahami pengembang baru sebelum mulai menulis kode?	Nonaka & Takeuchi (1995)

3. **Studi Literatur.** Studi literatur dilakukan untuk memperoleh landasan teori terkait metode *Waterfall*, *code review*, LLM, *black-box testing*, SUS, dan metrik evaluasi deteksi.
4. **Kuesioner SUS.** Kuesioner SUS digunakan untuk mengukur penerimaan dan kemudahan penggunaan sistem dari sudut pandang pengguna. Kuesioner terdiri atas 10 pernyataan dengan skala jawaban 1 sampai 5.



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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

BAB IV

HASIL DAN PEMBAHASAN

Bab ini membahas analisis kebutuhan, perancangan sistem, implementasi sistem dan pengujian sistem. Dimulai dari analisis kebutuhan fungsional & non-fungsional, penyusunan *rulebook*, implementasi sistem dan antarmuka web, pengujian fungsional & non-fungsional, penilaian pengguna dengan SUS.

4.1 Analisis Kebutuhan

Kebutuhan sistem dibagi menjadi kebutuhan fungsional, kebutuhan non-fungsional, dan kebutuhan perangkat pengembangan.

kebutuhan adalah tahap awal yang penting untuk memahami secara mendalam permasalahan pada proses *code review* aplikasi WISE/ESCM saat ini. Kebutuhan sistem pada penelitian ini dirumuskan secara objektif melalui tiga metode pendekatan, yaitu pengamatan alur kerja, wawancara dengan *Lead Developer*, serta penelaahan terhadap *source code* aplikasi.

4.1.1 Kebutuhan Fungsional

Kebutuhan fungsional sistem dibentuk atas dasar identifikasi masalah di sub bab 3.1.1 dan untuk sebagai tujuan pengembangan sistem penelitian ini, oleh karena itu, berdasarkan *statement* diatas, penulis mengusung kebutuhan fungsional sistem adalah sebagai berikut:

1. Sistem dapat menampilkan halaman utama antarmuka web.
2. Sistem dapat menerima unggahan berkas kode dengan cara *drag-and-drop* maupun menelusuri berkas (*browse*).
3. Sistem dapat memvalidasi berkas yang diunggah hanya berekstensi *.php* dan menolak berkas dengan ekstensi lain disertai pesan kesalahan.
4. Sistem dapat menjalankan pemeriksaan kode menggunakan *review engine* dan model AI terhadap berkas yang diunggah.
5. Sistem dapat menampilkan kemajuan pemeriksaan (*progress*) selama proses berlangsung.
6. Sistem dapat menampilkan ringkasan hasil pemeriksaan.



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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

7. Sistem dapat menampilkan rincian temuan dan potongan kode terkait.
8. Sistem dapat menampilkan riwayat pemeriksaan.
9. Sistem dapat menyediakan formulir penilaian pengguna dengan SUS pada antarmuka web.
10. Sistem dapat mengirim hasil penilaian pengguna ke Google Sheet melalui *backend*.

4.1.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional sistem adalah sebagai berikut.

1. Sistem diharapkan dapat berfungsi dengan performa yang baik, tidak membutuhkan waktu proses yang panjang

4.1.3 Perangkat Pengembangan

Perangkat pengembangan yang digunakan ditampilkan pada Tabel 3.

Tabel 3. Perangkat Pengembangan Sistem

Komponen	Teknologi	Fungsi
<i>Review engine</i>	Node.js	Menjalankan logika pemeriksaan kode dan pemanggilan OpenAI API
Model AI	GPT-4o-mini	Membantu menganalisis kode berdasarkan <i>rulebook</i>
<i>Backend</i> antarmuka web	Express.js	Menyediakan API untuk unggah berkas, pemeriksaan, riwayat, dan penilaian pengguna
<i>Frontend</i> antarmuka web	React.js + Vite	Menampilkan antarmuka web interaktif
Penyimpanan penilaian pengguna	Google Sheet + Apps Script	Menyimpan hasil kuesioner SUS responden



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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Komponen	Teknologi	Fungsi
Penempatan (<i>deployment</i>)	Layanan Node / mesin virtual	Menyediakan akses peragaan melalui peramban

Sumber: di olah sendiri oleh penulis

4.2 Perancangan Sistem

Tahap perancangan difokuskan pada arsitektur sistem dan penyusunan *rulebook* dan melakukan *prompt engineering*. Secara garis besar, tahapan ini mencakup perancangan logika dan arsitektur sistem di balik layar (*backend*), serta perancangan antarmuka web.

4.2.1 Perancangan Arsitektur dan Logika Sistem

1) Review Engine

Pemodelan AI ini dirancang untuk mendeteksi pelanggaran aturan (*rulebook*) pada kode sumber. Karena LLM secara bawaan tidak mengetahui aturan internal aplikasi WISE/ESCM, sistem ini menggunakan pendekatan *Prompt Engineering* untuk menanamkan pengetahuan spesifik tersebut.

Rancangan pemodelan deteksi AI adalah proses AI menganalisis kode sumber untuk menemukan pelanggaran terhadap *rulebook*, ini dibangun berdasarkan hasil ekstraksi wawancara, yang kemudian diproses melalui beberapa tahap hingga menghasilkan temuan akhir berupa daftar pelanggaran aturan pada kode yang diperiksa, dengan alur sebagai berikut:

a) Knowledge Elicitation

Tahap pertama adalah menggali aturan tak tertulis (*tacit knowledge*) terkait arsitektur, konvensi pangkalan data, hingga keamanan (sebagaimana kisi-kisi pada Subbab 3.3). Hasil wawancara mendalam dengan *Lead Developer* diekstraksi dan diterjemahkan menjadi *Rulebook* Internal yang pasti, yang terbagi menjadi aturan kerangka kerja (CTDL) dan aturan khas organisasi (WIKI).

b) Prompt Engineering



© Hak Cipta milik 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Rulebook yang telah tersusun kemudian diubah ke dalam format instruksi sistem (*System Prompt*) yang berlapis. *Prompt* ini bertindak sebagai "otak" tambahan bagi model AI, yang memberitahukan peran AI sebagai *code reviewer* senior, memberikan konteks teknologi (Laravel & Citadel), dan mendaftar aturan spesifik yang harus ditaati.

c) *Review Engine*

Berkas kode PHP yang diunggah pengguna dan *System Prompt* yang berisi *rulebook* dikirimkan secara bersamaan ke model LLM (GPT-4o-mini) melalui OpenAI API. Model akan membaca kode dan mencocokkannya dengan aturan-aturan tersebut.

d) *Hallucination Guard*

Untuk memitigasi sifat AI yang terkadang menghasilkan temuan palsu (*false positive*), sistem dilengkapi penyaring. Keluaran mentah dari AI akan divalidasi kembali untuk memastikan apakah baris kode yang ditunjuk oleh AI benar-benar ada di dalam berkas sumber.

e) Keluaran Final

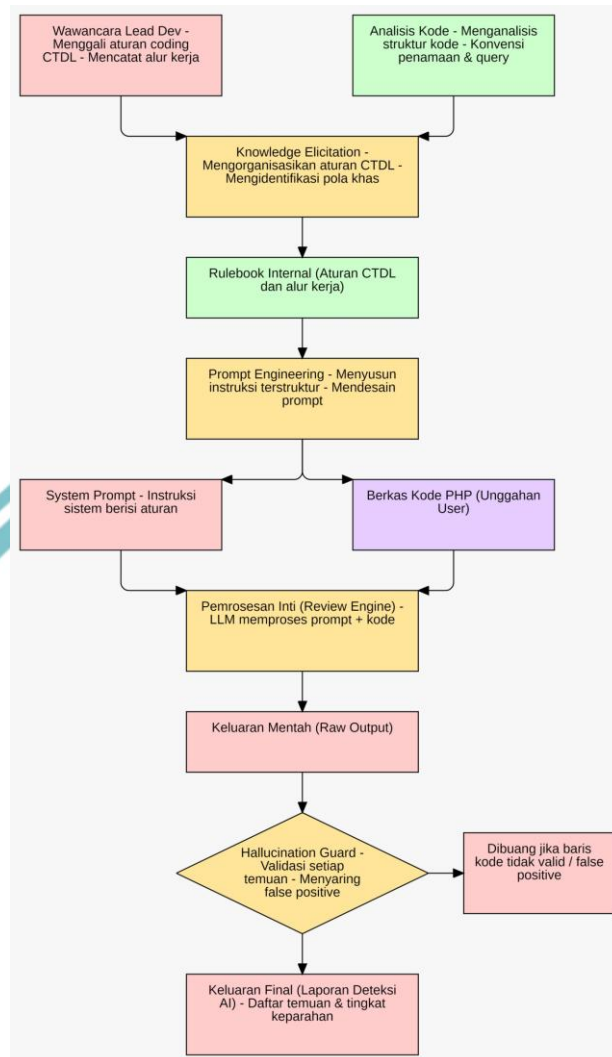
Temuan yang lolos penyaringan akan dikategorikan berdasarkan tingkat keparahannya, yaitu salah satu dari lima tingkat *blocker*, *critical*, *major*, *minor*, atau *info* (diurutkan dari yang paling serius hingga paling ringan), kemudian disajikan sebagai laporan final.

Rancangan pemodelan deteksi AI ini dapat dilihat pada Gambar 4.1.



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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Gambar 4. 1 Rancangan Pemodelan Deteksi AI Berbasis Ekstraksi Pengetahuan

Sumber: StarUML

2) Perancangan Sistem

Sistem dirancang dengan dua bagian utama, yaitu *review engine* dan antarmuka web. Perancangan ini dijabarkan melalui *Use Case Diagram*, *Flowchart* alur kerja sistem dan *Wireframe*

a. Use Case Diagram

Use case diagram memvisualisasikan interaksi yang dapat dilakukan oleh aktor (Pengguna/Pengembang) terhadap sistem pendukung *code review*. Berdasarkan analisis kebutuhan, pengguna memiliki akses ke beberapa fungsi utama.

Aktor: Pengguna (Pengembang/Peninjau Kode)



© Hak Cipta milik 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Skenario (*Use Cases*):

1. Mengunggah berkas kode: Pengguna dapat mengunggah berkas berekstensi .php secara *drag-and-drop* atau *browse*.
2. Melihat kemajuan pemindaian: Pengguna dapat memantau proses pemeriksaan AI saat sedang berjalan.
3. Melihat laporan hasil pemeriksaan: Pengguna dapat melihat ringkasan temuan, tingkat keparahan (*severity*), rincian potongan kode, dan saran perbaikan.
4. Melihat riwayat pemeriksaan: Pengguna dapat mengakses *audit trail* atau riwayat berkas yang pernah diperiksa sebelumnya.
5. Mengisi kuesioner SUS: Pengguna dapat memberikan penilaian sistem melalui formulir *System Usability Scale* yang terintegrasi.

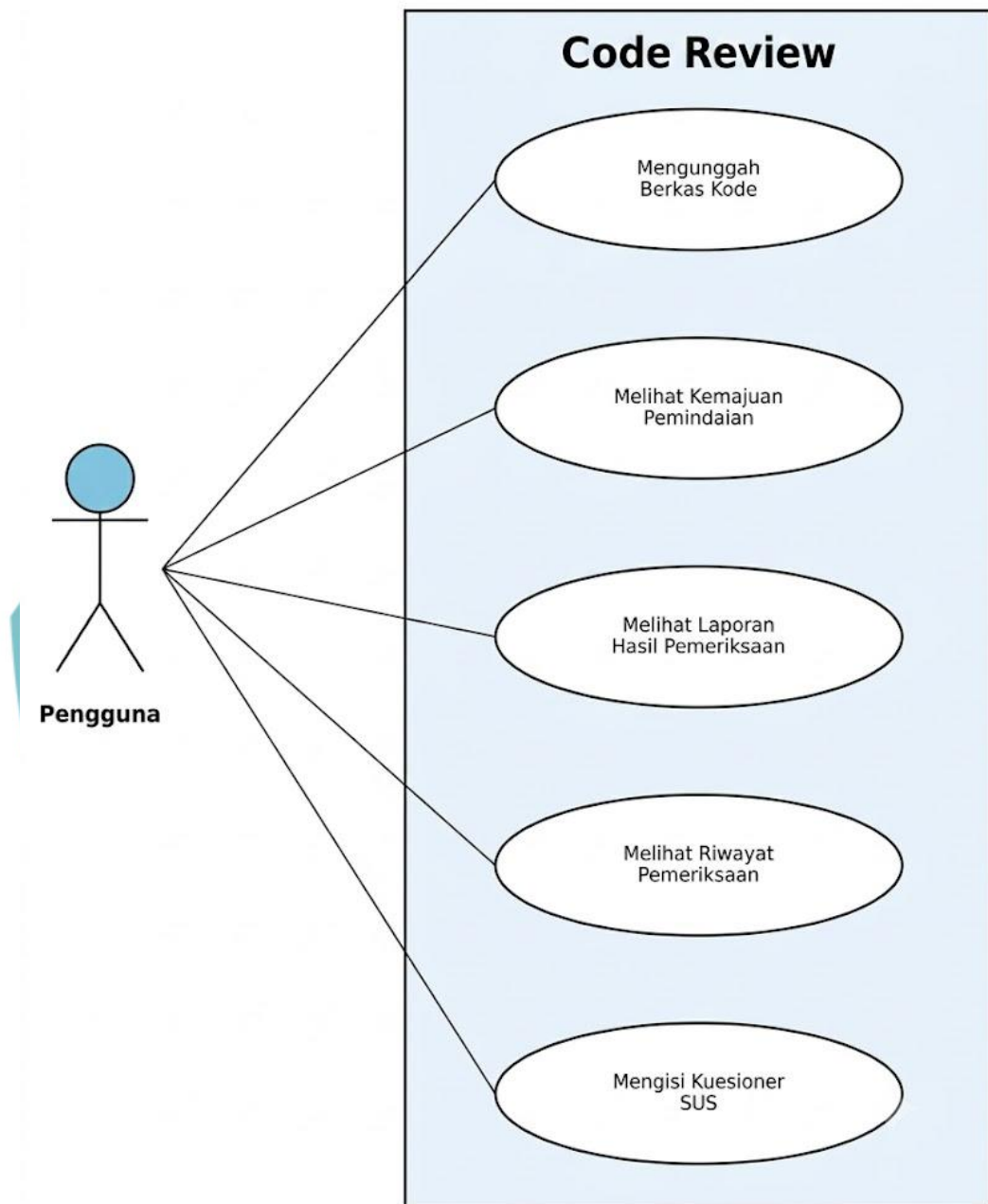




© Hak Cipta milik Politeknik Negeri Jakarta

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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Gambar 4. 2 Use Case Diagram

Sumber: StarUML

b. Flowchart

Flowchart ini menggambarkan alur logika sistem secara sekuensial, mulai dari pengguna mengunggah berkas hingga sistem menampilkan hasil pemeriksaan. Sistem mengimplementasikan validasi di awal dan penyaring halusinasi



© Hak Cipta milik 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

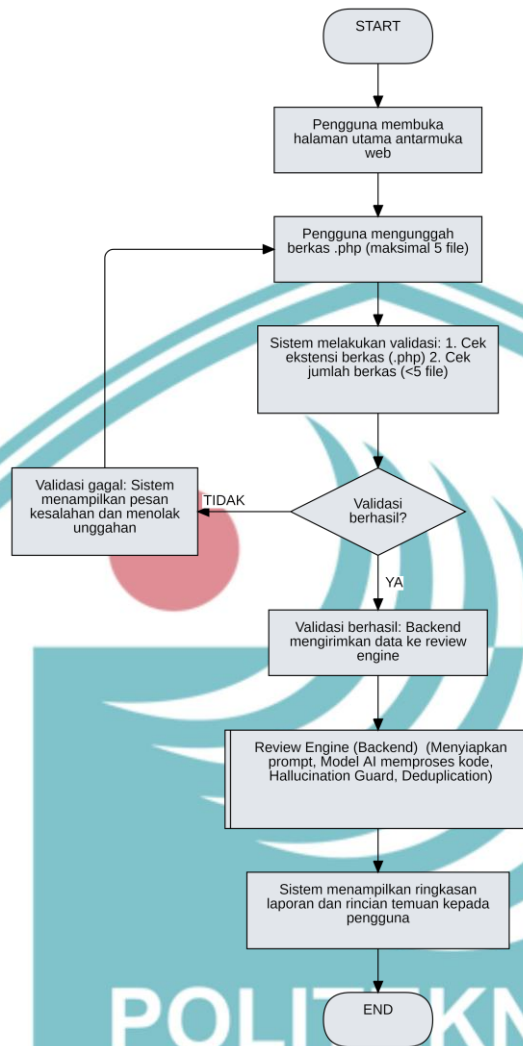
(*Hallucination Guard*) setelah pemanggilan AI untuk menjaga keandalan keluaran. Alur Kerja (*Flowchart*):

1. Pengguna membuka halaman utama antarmuka web.
2. Pengguna mengunggah berkas yang berformat .php
3. Sistem melakukan validasi: memeriksa apakah ekstensi berkas adalah .php
4. Jika validasi gagal, sistem menampilkan pesan kesalahan dan menolak unggahan. Pengguna dikembalikan ke tahap unggah.
5. Jika validasi berhasil, *backend* mengirimkan data ke *Review Engine*.
6. *Review Engine* menyiapkan *prompt* berlapis berdasarkan *rulebook internal* (CTDL dan WIKA) lalu memanggil OpenAI API.
7. Model AI memproses kode dan mengembalikan hasil temuan.
8. Sistem mengeksekusi *Hallucination Guard* untuk memastikan temuan menunjuk pada baris kode yang benar-benar ada dan membuang temuan palsu (*false positive*).
9. Sistem menggabungkan temuan yang sama (*deduplication*).
10. Sistem menampilkan ringkasan laporan dan rincian temuan kepada pengguna.
11. Selesai.



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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Gambar 4. 3 Flowchart Sistem

Sumber: StarUML

c. Rancangan Antarmuka.

Rancangan antarmuka web dibangun dengan mengutamakan kemudahan pengguna dalam melakukan pemeriksaan kode secara mandiri. Karena antarmuka ini akan digunakan oleh pengembang dan penguji (*QA/Tester*), desain dibuat bersih (*clean*) dan berfokus pada penyajian informasi teknis. Berikut adalah deskripsi rancangan (*wireframe*) untuk halaman-halaman utama pada tampilan web.

1. Rancangan Halaman Utama & Area Unggah

Halaman ini dirancang untuk memberikan instruksi singkat dan memfasilitasi proses unggah berkas.



© Hak Cipta milik 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- a. *Header*: Logo>Nama Aplikasi (contoh: `codex//review`) di pojok kiri atas.
- b. Bagian "Apa ini?": Kotak informasi berisi 3 langkah penggunaan (1. Pilih contoh kode, 2. Tampilkan hasil, 3. Baca laporan).
- c. Area Unggah: Sebuah kotak putus-putus (*dashed border*) berukuran besar di tengah layar dengan ikon awan/berkas. Terdapat teks instruksi "Seret file ke sini atau klik untuk memilih" beserta keterangan batasan sistem ("Maksimal 5 file berekstensi .php").
- d. Tombol Aksi: Tombol utama "Periksa dengan AI" yang berada di bawah area unggah.



Gambar 4. 4 Rancangan Halaman Utama dan Area Unggah

Sumber: Figma

2. Rancangan Halaman Tampilan Pemrosesan

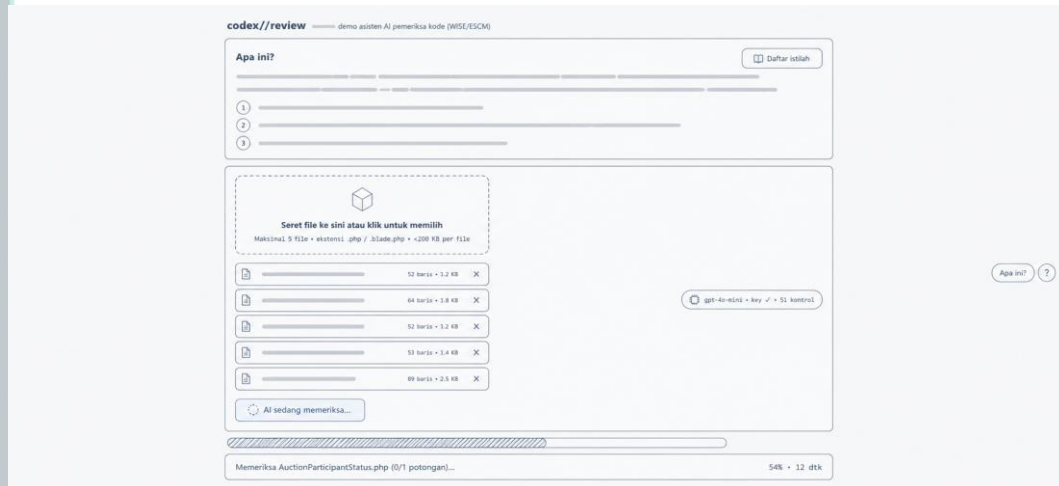
Halaman ini muncul ketika pengguna menekan tombol periksa, memberikan umpan balik visual agar pengguna mengetahui bahwa sistem sedang bekerja.

- a. **Daftar Berkas**: Menampilkan daftar nama berkas (contoh: `VendorController.php`) yang berhasil divalidasi dan diunggah.
- b. **Indikator Kemajuan (*Progress Bar*)**: Sebuah bilah memanjang di bagian bawah layar yang menunjukkan persentase atau status pemeriksaan (contoh: "Memeriksa VendorController.php (1/5 potongan)...").



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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Gambar 4. 5 Rancangan Halaman Tampilan Proses

Sumber: Figma

3. Rancangan Halaman Ringkasan & Rincian Laporan

Ini adalah halaman paling krusial yang menampilkan hasil dari *review engine*.

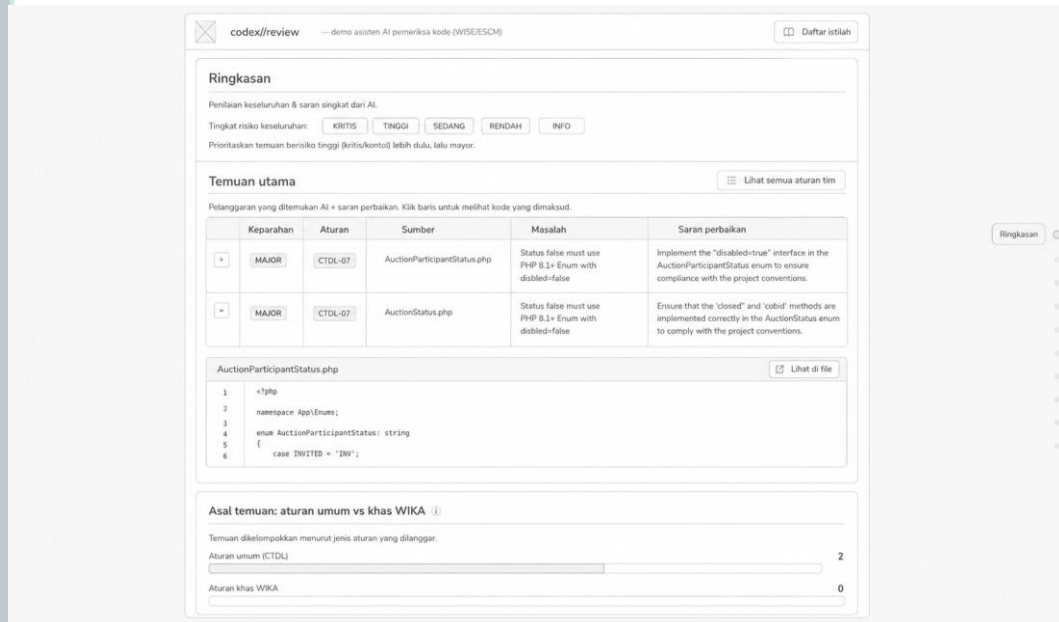
- a. Kartu Ringkasan: Kotak di bagian atas yang menampilkan jumlah temuan berdasarkan tingkat keparahan (*High, Medium, Low*).
- b. Tabel Temuan Utama: Tabel yang memuat kolom: Keparahan, Kode Aturan (contoh: CTDL-01), Sumber (nama berkas), Masalah, dan Saran Perbaikan.
- c. Cuplikan Kode (*Code Snippet*): Kotak berlatar belakang gelap (mirip *text editor/IDE*) yang diletakkan tepat di bawah tabel. Kotak ini menampilkan baris kode yang melanggar aturan dengan *highlight* warna pada baris yang bermasalah.
- d. Grafik Asal Temuan: Visualisasi batang mendatar (*horizontal bar*) di bagian paling bawah untuk membedakan jumlah temuan dari aturan umum (CTDL) versus aturan khas organisasi (WIKI).



© Hak Cipta milik 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



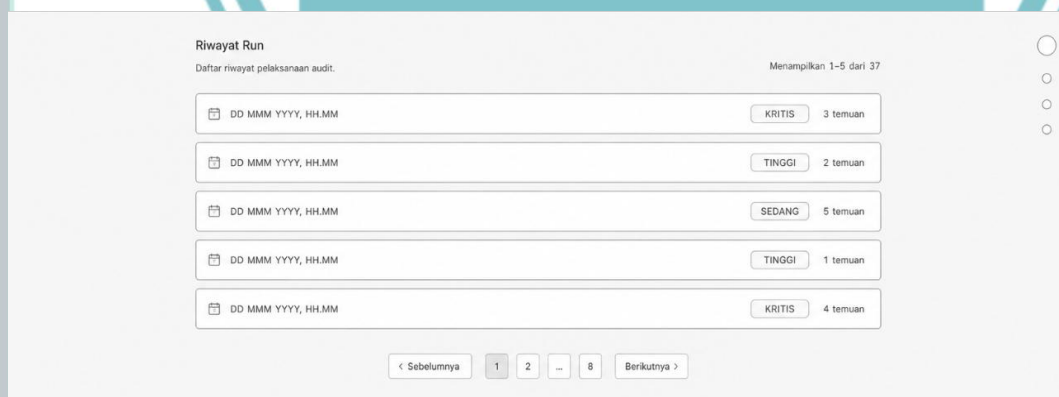
Gambar 4. 6 Rancangan Halaman Ringkasan dan Rincian Laporan

Sumber: Figma

4. Rancangan Halaman Riwayat Pemeriksaan (*Audit Trail*)

Halaman ini dirancang agar pengguna dapat menelusuri kembali hasil pemeriksaan sebelumnya.

- a. Daftar Riwayat: Berupa daftar baris memanjang ke bawah. Setiap baris menampilkan informasi: Tanggal & Waktu (contoh: "Rabu, 1 Juli 2026, 14:30 WIB"), tombol status (*Valid/Error*), dan jumlah temuan.
- b. Navigasi (*Pagination*): Tombol "Sebelumnya", angka halaman (1, 2, 3), dan "Berikutnya" di bagian bawah daftar.



Gambar 4. 7 Tampilan Riwayat Run

Sumber: Figma



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

3) Arsitektur Penyimpanan Data

Berbeda dari aplikasi *enterprise* pada umumnya, purwarupa ini secara sengaja dirancang tanpa basis data relasional maupun NoSQL. Keputusan ini penulis ambil sebab hal ini merupakan pertimbangan teknis penulis sendiri, bukan kebutuhan yang diminta oleh pengguna (*Lead Developer*), dengan pertimbangan bahwa purwarupa ini masih berskala kecil dan belum digunakan sebagai sistem produksi harian di perusahaan, sehingga pengelolaan basis data terpisah dirasa belum diperlukan pada tahap ini. Sebagai gantinya, seluruh data yang dibutuhkan sistem disimpan dalam dua bentuk berkas yang berbeda, sesuai sifat dan *lifecycle* datanya masing-masing, sebagaimana dijelaskan berikut ini.

Data definisi (statis, otoritatif). *Rulebook* 30 kontrol (Tabel 6.) disimpan sebagai berkas *controls.csv*, dimuat sekali ke memori saat *backend* dijalankan untuk menentukan batas bawah tingkat keparahan tiap temuan.

Data riwayat pemeriksaan. Setiap kali pengguna melakukan pemeriksaan kode melalui antarmuka web, hasilnya dicatat sebagai satu berkas JSON dan sudah *timestamped* di direktori riwayat, memuat identitas run, *hash* SHA-256 dari berkas yang diunggah, temuan, dan estimasi biaya token. Arsitektur ini dipilih dengan niatan agar setiap kali fitur utama sistem ini dijalankan, *which is scan* kode nya itu sendiri, memiliki jejak audit yang asli.

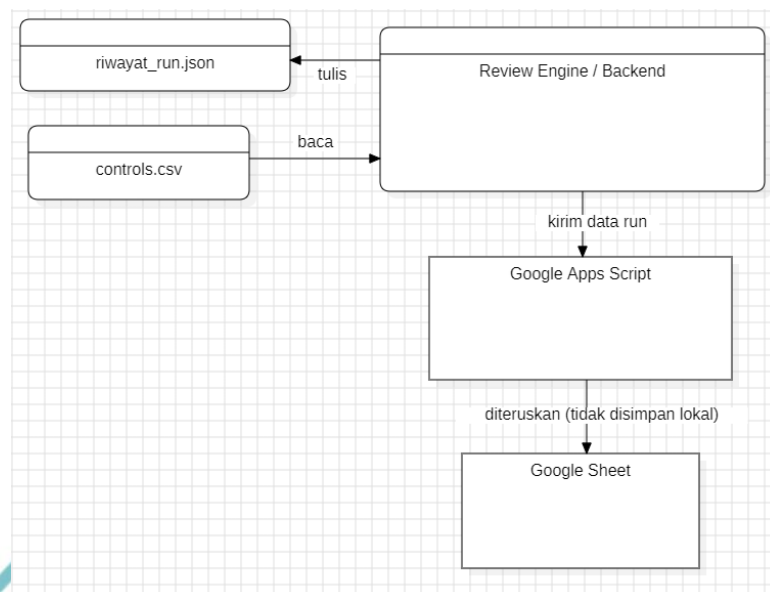
Untuk jawaban kuesioner *System Usability Scale* (SUS) hasil inputan *user* tidak disimpan di sisi *backend* sama sekali. *Backend* hanya bertindak sebagai *proxy* yang memvalidasi kelengkapan jawaban, lalu meneruskannya ke Google Apps Script Web App yang terhubung langsung ke Google Sheet, termasuk mekanisme pengecekan duplikat pada baris alamat *e-mail* agar satu responden hanya dapat mengisi satu kali.

Pola penyimpanan ini ditunjukkan pada Gambar 4.8.



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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Gambar 4. 8 Alur Data *Review Engine*

4.2.2 Penyusunan Buku Pedoman (*Rulebook*) melalui Elisitasi Pengetahuan

Penyusunan *rulebook* dimulai dari sini

1) Hasil Pengamatan dan Wawancara dengan *Lead Developer*

Knowledge elicitation dilakukan melalui wawancara bersama *lead developer* di PT Wijaya Karya (Persero) Tbk. Wawancara ini bertujuan untuk mengekstraksi informasi operasional, arsitektur, serta aturan tak tertulis yang selama ini menjadi pedoman mutlak bagi para pengembang.

Proses ekstraksi aturan tak tertulis berpedoman pada 10 pertanyaan, dan 10 pertanyaan tersebut disusun berdasarkan Tabel 3.2, dan metodenya mengikuti cara Fenoglio dkk. (2022) untuk menggali *tacit knowledge*. wawancara terstruktur yang dirancang menjangkau seluruh lapisan arsitektur sistem, mulai dari gambaran umum hingga aturan tak tertulis yang jarang terdokumentasi. Daftar pertanyaan beserta transkrip lengkap jawaban *Lead developer* didokumentasikan secara utuh pada Lampiran 1.

Hasil wawancara dianalisis menggunakan pendekatan analisis tematik (*thematic analysis*) sebagaimana diusulkan oleh Braun & Clarke (2006), di mana jawaban dari sepuluh pertanyaan dikelompokkan berdasarkan kemiripan topik dan



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

keterkaitan area fungsional sistem, sehingga diperoleh lima tema utama sebagai berikut.

- a) **Arsitektur & Kerangka Kerja Internal (P1–P2).** Sistem dibangun di atas Laravel dan kerangka kerja *custom* Citadel, hasil abstraksi kebutuhan bisnis ke teknis agar developer cukup fokus pada *backend/controller* sementara *frontend* ditangani *framework*. Arsitektur ini lah yang menjadi dasar pembuatan sistem yang penulis buat
- b) **Konvensi Basis Data & Validasi (P3–P4).** Standar model mengikuti PSR-4 dengan *trait* wajib (timestamp, creator, indexing terpisah dari PK), dan validasi dilakukan *inline* di *controller* bukan karena mengikuti *best practice* standar Laravel, melainkan demi *clean code* (menghindari boros baris). Ini lumayan penting untuk digaris bawahi juga.
- c) **Keamanan, Otorisasi, dan Isolasi Data (P5–P6).** Otorisasi ditentukan kombinasi Level Organisasi dan Role secara hierarkis, sedangkan isolasi data antar-proyek dijaga lewat scope khusus di tiap model. Konsep ini menjadi fondasi utama bagi fungsi keamanan yang *high risk*, seperti `checkAccessPermission()` dan `ESCM::scopeProject()`. Jika terjadi celah keamanan pada bagian ini, dampaknya akan sangat fatal karena dapat memicu kebocoran data antar proyek pengadaan.
- d) **Logika Bisnis Inti & Integrasi (P7–P8).** A *Approval workflow* memakai pola *traversal*. Pola *Traversal* adalah metode *approval* yang tidak "mem-book" siapa yang harus menyetujui selanjutnya, melainkan hanya merekam riwayat persetujuan saat itu terjadi. Ini membuat rute persetujuan dokumen menjadi jauh lebih cerdas, bebas macet, dan fleksibel. Integrasi ke sistem eksternal (SAP, HCMS Human Capital Management System) memakai *scheduler full sync* harian dengan *pagination*. Tingginya kompleksitas ini menjelaskan mengapa aturan seperti WIKA-Q05 dan WIKA-Q11 tidak mungkin dideteksi oleh alat analisis statis biasa.
- e) **Audit, Kepatuhan, dan Tacit Knowledge Onboarding (P9–P10).** *Audit trail* bersifat *raw*, mencatat nama, jabatan, dan target data, masih mengandalkan error log bawaan Laravel. Lima prinsip *onboarding* yang disebutkan *Lead*



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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Developer meliputi: (1) *separation of concern*, yaitu pemisahan tanggung jawab antar lapisan kode agar developer baru cukup fokus pada satu bagian saja; (2) dokumentasi fungsi secara konsisten; (3) pendekatan CRUD-first dalam pengembangan fitur baru; (4) *refactor* progresif, yaitu penyempurnaan kode secara bertahap tanpa mengubah fungsionalitasnya; dan (5) penghapusan kode yang tidak lagi dipanggil atau dijalankan namun masih tertinggal di dalam basis kode (Fowler, 1999). Kelima prinsip ini adalah yang penulis anggap representasi paling jelas dari *tacit knowledge* yang selama ini belum terdokumentasikan secara formal.

2) Hasil Analisis Mandiri Kode (*Static Code Analysis*)

Untuk memahami kondisi aktual aplikasi WISE/ESCM, dilakukan pemahaman dan analisis kode aplikasi WISE/ESCM secara statis (*static code analysis*) oleh penulis, menggunakan kombinasi tiga pendekatan. Pertama, seluruh repositori dipetakan strukturnya menggunakan Graphify, sebuah alat yang mengubah kode sumber menjadi graf pengetahuan (*knowledge graph*) yang menunjukkan hubungan antar-berkas, kelas, dan fungsi, sehingga penulis dapat mengidentifikasi berkas-berkas kunci yang secara arsitektural paling signifikan untuk ditelusuri lebih lanjut. Kedua, pada berkas-berkas kunci yang teridentifikasi tersebut, penulis melakukan pembacaan manual untuk memahami konteks dan logika di baliknya, seperti alur kerja fitur persetujuan (*approval*). Ketiga, untuk menghitung frekuensi kemunculan pola kode tertentu secara presisi di seluruh repositori, seperti penggunaan atribut atau pemanggilan fungsi tertentu, penulis menggunakan fitur pencarian pola (*Find in Files*) pada *code editor* serta perintah *grep* melalui *terminal*. Analisis ini difokuskan pada penemuan pola *coding*, implementasi keamanan, serta kepatuhan terhadap standar *clean code*.

Setelah selesai analisis banyak baris kode yang mencakup *controller*, model, dan *services*, ditemukan berbagai karakteristik arsitektur yang sangat spesifik serta beberapa inkonsistensi.

Salah satu karakteristik yang ditemukan adalah banyaknya penggunaan atribut *guarded* dibandingkan atribut *fillable* pada mayoritas model (247 berbanding 9 model). Kedua atribut ini merupakan mekanisme keamanan bawaan Laravel yang



© Hak Cipta milik 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

mengatur kolom pada basis data mana saja yang boleh diisi secara otomatis oleh input pengguna, atribut *guarded* membuka seluruh kolom kecuali yang secara eksplisit diblokir, sehingga secara konsep lebih berisiko dibandingkan atribut *fillable* yang hanya membuka kolom yang secara eksplisit diizinkan.

Karakteristik lain yang ditemukan adalah penerapan alur persetujuan atau *approval* berbasis *dependency-graph*, yaitu satu proses persetujuan dapat menunggu penyelesaian beberapa proses lain secara bersamaan, bukan sekadar berjalan berurutan satu demi satu. Pola semacam ini tidak sering ditemui pada aplikasi Laravel standar.

Selain kedua karakteristik tersebut, ditemukan pula inkonsistensi berupa sisa 432 pemanggilan fungsi pelacakan atau *debugging dd()* yang masih tertinggal pada 154 file di repositori utama. Fungsi ini biasa digunakan pengembang untuk memeriksa nilai suatu variabel selama proses *development*, namun seharusnya dihapus sebelum kode digunakan secara operasional.

Temuan-temuan dari analisis mandiri ini dikelompokkan berdasarkan aspek teknisnya dan disajikan pada Tabel 4.. Sedangkan rincian penjabaran temuan analisis kode statis pada Repositori WISE/ESCM dapat dilihat selengkapnya pada Lampiran 2.

Tabel 4. Ringkasan Temuan Analisis Kode Statis pada Repositori WISE/ESCM

No	Aspek Teknis	Temuan Faktual pada Basis Kode	Implikasi terhadap Sistem Tinjauan Kode	Referensi
1	Praktik Validasi	Tidak ditemukan kelas Form Request. Fungsi <code>\$request->validate()</code> sebaris terdeteksi lebih dari 230 kali pada 105 fail pengendali.	Membuktikan perlunya AI dengan aturan khusus (WIKI-Q02), karena pemindai standar (SAST generik) akan menganggap ini sebagai pelanggaran.	OWASP Input Validation Cheat Sheet (OWASP Foundation)
2	Keamanan Basis Data	Dominasi atribut <code>\$guarded=['id']</code> (pada 247 model) dibandingkan <code>\$fillable</code> (hanya pada 9 model).	Konvensi pencegahan kerentanan penugasan massal (<i>mass-assignment</i>) ini wajib dipantau ketat secara otomatis (CTDL-02).	OWASP Mass Assignment Cheat Sheet (OWASP Foundation)



© Hak Cipta milik 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

3	Otorisasi Akses & Isolasi Data	Fungsi <code>checkAccessPermission()</code> dipanggil 616 kali. Isolasi penyewa ganda (<i>multi-tenant</i>) dikelola manual via <code>ESCM::scopeProject()</code> .	Kelalaian manusia pada satu baris ini berisiko eskalasi hak akses. AI wajib diinstruksikan mendeteksi ketiadaan fungsi pengaman ini (CTDL-11 & WIK-Q07).	Ferraiolo & Kuhn (1992); Bezemer & Zaidman (2010)
4	Logika Persetujuan (<i>Approval</i>)	Alur persetujuan sangat spesifik menggunakan grafik kebergantungan (<i>dependency-graph</i>) via atribut <code>dependable_id</code> .	Logika bisnis ini tidak akan dipahami alat analisis standar. Mutlak membutuhkan instruksi cerdas (<i>prompt</i>) AI (WIK-Q11).	van der Aalst & van Hee (2002)
5	Kualitas & Kebersihan Kode	Ditemukan 432 sisa pemanggilan pelacakan (debugging) <code>dd()</code> yang tertinggal di 154 fail repositori utama.	Membuktikan secara empiris bahwa tinjauan kode manual saat ini kewalahan (<i>cognitive fatigue</i>) sehingga otomatisasi sangat mendesak.	Fowler (1999)

Berdasarkan temuan-temuan empiris pada Tabel 4. di atas, dapat ditarik dua kesimpulan krusial. Pertama, tingginya jumlah sisa kode debugging yang lolos hingga ke *branch* utama membuktikan bahwa proses tinjauan kode manual yang dilakukan oleh pengembang saat ini mengalami *bottleneck* dan *cognitive fatigue*, sejalan dengan tantangan beban dan waktu peninjauan yang diidentifikasi oleh Davila & Nunes (2021), sehingga tidak lagi efektif dalam menyaring kesalahan.

Kedua, kondisi arsitektur aplikasi WISE/ESCM yang sangat termodifikasi, sebagaimana ditunjukkan pada aspek Praktik Validasi di Tabel 4.—membuktikan bahwa penggunaan alat Pengujian Keamanan Aplikasi Statis atau *Static Application Security Testing/SAST* biasa tidak akan cocok untuk diimplementasikan pada proyek ini. Alat analisis biasa akan secara keliru mengidentifikasi ketiadaan kelas `Form Request` sebagai sebuah pelanggaran, dan sebaliknya, tidak akan menyalahkan ketiadaan fungsi pengaman lokal seperti `ESCM::scopeProject()`. Oleh karena itu, diperlukan sebuah sistem otomatisasi cerdas yang dapat di beri instruksi secara khusus melalui *prompt engineering* untuk melakukan *review* kode murni berdasarkan aturan arsitektur asli yang berlaku di PT Wijaya Karya.



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

3) *Gap Analysis*

Tahap akhir untuk penyusunan *rulebook* adalah melakukan perbandingan antara hasil analisis kode dan wawancara yang telah dilakukan, sebagaimana telah dijelaskan pada Subbab 3.1 bahwa pendekatan kualitatif penelitian ini juga mencakup tahap *gap analysis* untuk mengelompokkan kesesuaian temuan ke dalam tiga status (Sugiyono, 2019)

Proses *mapping* atau pemetaan dilakukan secara manual oleh penulis. Setiap jawaban *Lead Developer* atas kesepuluh pertanyaan wawancara pada Tabel 3.2 dipecah menjadi beberapa klaim spesifik yang dapat diverifikasi satu per satu terhadap kode aktual; sebagai contoh, jawaban atas pertanyaan pertama (P1) mengenai arsitektur menghasilkan 4 klaim spesifik seperti menggunakan PHP Laravel, pakai JavaScript/jQuery, pakai Citadel *custom*, dan pakai react di beberapa modul. Sedangkan jawaban atas pertanyaan kedua (P2) mengenai *framework* Citadel menghasilkan 7 klaim spesifik. Secara keseluruhan, proses ini menghasilkan 48 klaim spesifik dari kesepuluh pertanyaan wawancara tersebut.

Setiap klaim kemudian dibandingkan dengan bukti baris kode aktual pada Repositori WISE/ESCM, dan dikategorikan menjadi tiga status: sama, yaitu ketika klaim wawancara sepenuhnya konsisten dengan implementasi pada kode. Tidak sama, yaitu ketika klaim wawancara bertentangan dengan temuan pada kode. Dan parsial, yaitu ketika klaim wawancara sebagian konsisten namun memerlukan konteks atau penjelasan tambahan untuk dapat diverifikasi sepenuhnya. Kategori pada kolom Topik Wawancara di Tabel 5. ini mengikuti kesepuluh topik pertanyaan pada Tabel 3.2, berbeda dengan kategori pada Tabel 4. yang disusun berdasarkan aspek yang muncul murni dari hasil pembacaan kode. Hasil perbandingan dari keseluruhan 48 klaim tersebut secara garis besar dirangkum dalam Tabel 5., sedangkan rincian penjabaran beserta bukti baris kodenya dapat dilihat selengkapnya pada Lampiran 3.

Tabel 5. Ringkasan Analisa Hasil Wawancara dengan Basis Kode WISE/ESCM



© Hak Cipta milik 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Topik	Selaras	Tidak sama	Parsial	Total
Wawancara	Penuh			
Arsitektur & Komponen Utama	3	0	1	4
Kerangka Kerja Citadel	6	0	1	7
Konvensi Basis Data & Model	5	1	3	9
Konvensi Validasi	2	1	0	3
Keamanan & Otorisasi	1	0	3	4
Isolasi Data	2	1	0	3
Logika Bisnis Inti (Approval)	1	0	3	4
Integrasi Pihak Ketiga	3	1	1	5
Audit & Kepatuhan	3	0	1	4
Aturan tak tertulis	2	1	2	5
Total Klaim	28	5	15	48
Persentase Keseluruhan	58.30%	10.40%	31.30%	100%

1. Dari 48 hasil yang di dapat, ada 28 hasil (58,3%) terbukti sama dan 5 hasil (10,4%) tidak sama. Dari temuan tersebut, diperoleh tingkat kesamaan



© Hak Cipta milik 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- sebesar 68,8%. Tingginya tingkat kesamaan ini membuktikan bahwa mayoritas aturan dari hasil wawancara dan analisis kurang lebih sama.
2. Meskipun demikian, masih ditemukan 15 hasil atau 31,3% yang tidak sepenuhnya sesuai antara hasil wawancara dan kondisi nyata pada kode. Perbedaan terbesar terdapat pada proses persetujuan (*approval workflow*). Narasumber menyebut adanya pola internal bernama *traversal*, yaitu proses untuk menentukan pihak yang berhak menyetujui secara bertahap, baik secara berurutan maupun bersamaan. Pola ini sudah digunakan secara konsisten dalam kode, tetapi belum dijelaskan dengan jelas. Perbedaan lain ditemukan pada pengaturan hak akses pengguna berdasarkan jabatan, unit kerja, dan peran pengguna. Selain itu, belum terdapat aturan tertulis mengenai kewajiban dokumentasi kode menggunakan PHPDoc dan pembersihan kode yang sudah tidak digunakan. Hal ini terlihat dari masih ditemukannya beberapa fungsi `dd()` untuk pengecekan sementara yang belum dihapus dari repositori utama.
 3. Kesenjangan antara standar praktik validasi yang direkomendasikan secara resmi oleh *framework* Laravel, yaitu memisahkan logika validasi ke dalam kelas *Form Request* tersendiri agar mudah diuji dan dipelihara (Laravel, 2026), dan praktik nyata yang diterapkan pada pengembangan aplikasi WISE/ESCM inilah yang menjadi alasan kuat mengapa sistem yang dirancang dalam penelitian ini sebaiknya dibangun. Sebanyak 15 dari 48 klaim wawancara (31,3%) terbukti tidak sepenuhnya sesuai dengan kondisi nyata pada kode, dan ditemukan pula 432 sisa pemanggilan fungsi pelacakan (*debugging*) `dd()` yang belum dihapus dari repositori utama—suatu praktik yang menurut Fowler (2022) tergolong sebagai *dead code*, yaitu bagian kode yang seharusnya dibersihkan namun masih tertinggal karena luput dari pemeriksaan. Kedua temuan ini membuktikan bahwa pengecekan kode secara manual oleh pengembang sudah tidak lagi efektif akibat faktor *cognitive fatigue*, sejalan dengan batasan beban dan waktu peninjauan yang diidentifikasi oleh Davila & Nunes (2021).
- Terlebih lagi, aplikasi ini dibangun dengan aturan yang sangat unik, seperti tidak memakai kelas *Form Request* dan menggunakan arsitektur Citadel



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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

ganda. Kondisi ini membuktikan bahwa alat pengecek kode otomatis biasa, atau *Static Application Security Testing/SAST*, dipastikan tidak akan efektif, karena ia akan salah dan keliru saat mencovba mengidentifikasi ketiadaan kelas *Form Request* sebagai sebuah pelanggaran, namun sebaliknya tidak akan menyalahkan ketiadaan fungsi pengaman lokal seperti `ESCM::scopeProject()`. Oleh karena itu, dibutuhkan *prompt engineering* pada model AI, sebagaimana diusulkan oleh Pornprasit dan Tantithamthavorn (2024), agar sistem dapat diarahkan mengevaluasi kode, murni berdasarkan aturan arsitektur khas perusahaan, tanpa perlu melatih ulang model AI dari awal.

4. Oleh karena itu, solusi mutlak yang dibutuhkan adalah membangun sistem cerdas yang menggabungkan dua hal, yaitu:
 - 1) Sebuah buku pedoman (*Rulebook*) yang disusun berdasarkan bukti nyata dari dalam kode.
 - 2) Sebuah mesin pemroses AI (GPT 4o-mini). Dengan teknik *prompt engineering*, AI tersebut akan diarahkan untuk mengevaluasi kode baru, murni berdasarkan *rulebook* internal perusahaan, sehingga saran perbaikan yang dihasilkan benar-benar akurat dan sesuai dengan konteks proyek.

Isi *rulebook* ini memuat total 30 aturan yang diklasifikasikan ke dalam dua kelompok:

1. Aturan Kerangka Kerja (*CTDL Rules*): Terdiri atas 15 kontrol spesifik yang mengatur standar penggunaan *framework* Citadel dan aturan *coding* internal, seperti kewajiban penggunaan Eloquent ORM dan proteksi *mass assignment*.
2. Aturan *Logic* Organisasi (*WIKI Engineering Rules*): Terdiri atas 15 kontrol yang dibentuk dari aturan tak tertulis atau *tacit knowledge* dari *Lead developer*. Aturan ini secara murni merepresentasikan logika bisnis, seperti ketiadaan kelas *Form Request* dan kewajiban sinkronisasi data eksternal.

Setiap aturan di dalam buku pedoman ini dievaluasi oleh *Lead Developer* berdasarkan tingkat risiko yang dirasa tepat dengan dokumentasi yang bisa



© Hak Cipta milik 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

diperiksa pada Lampiran 3 (*High, Medium, Low*), mengacu pada *OWASP Risk Rating Methodology* (OWASP Foundation/owasp.org) yang menentukan tingkat risiko berdasarkan kombinasi kemungkinan terjadinya pelanggaran (*likelihood*) dan dampak teknis maupun bisnis apabila pelanggaran tersebut terjadi (*impact*). Aturan berkategori *High* umumnya berkaitan langsung dengan kerentanan keamanan seperti risiko *SQL Injection* (CTDL-01) atau kebocoran hak akses (WIKI-Q07), kategori *Medium* diberikan pada aturan dengan risiko yang bersifat kondisional, yaitu berpotensi menimbulkan celah di kemudian hari namun belum tentu terjadi saat ini, sedangkan kategori *Low* diberikan pada aturan yang bersifat aturan penulisan kode dengan dampak teknis minimal. Tingkat risiko ini selanjutnya berperan sebagai batas bawah yang menentukan tingkat keparahan minimal dari temuan AI terkait aturan tersebut, sebagaimana dijelaskan lebih lanjut pada Subbab 4.3.1. Isi lengkap 30 kontrol yang menyusun *rulebook* ini ditunjukkan pada Tabel 6..

Tabel 6. menjadi isi penuh 30 baris

ID	Domain	Risiko	Judul Kontrol	Deskripsi & Referensi
CTD L-01	Database & ORM	High	Wajib Eloquent ORM - dilarang <i>raw</i> query	Semua akses DB harus via Eloquent. DB:: <i>raw</i> ()/DB:: <i>select</i> () dilarang kecuali agregasi kompleks. Ref: ASVS V5, CWE-89
CTD L-02	Mass Assignmen t	High	Mass assignment protection via \$guarded	Semua model HARUS \$guarded=["id"]. Tidak \$fillable. Ref: CWE-915
CTD L-03	XSS Prevention	High	Blade escaping - dilarang {!! !!} tanpa sanitasi	Output Blade HARUS {{ }}. {!! !!} hanya untuk HTML dari Citadel renderSchema(). Ref: CWE-79



© Hak Cipta milik 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

ID	Domain	Risiko	Judul Kontrol	Deskripsi & Referensi
CTD L-04	Authentica tion	High	Middleware auth pada protected routes	Route admin: cek_login+internal_place ment. Route API: basic_auth/jwt.verify. Ref: ASVS V4, CWE-862
CTD L-05	Validation	Mediu m	<i>Inline validation</i> wajib di <i>controller</i>	Setiap store()/update() HARUS \$request-> validate([...]). Project tidak pakai Form Request <i>class</i>
CTD L-06	Citadel Framework	Mediu m	Citadel Page <i>lifecycle</i> pattern wajib	<i>Controller</i> HARUS ikuti: ComponentsPage::make()- >view()->business()- >schema()->render()
CTD L-07	<i>Audit Trail</i>	High	HasCreator <i>trait</i> wajib pada model bisnis	Model entitas bisnis HARUS <i>trait</i> HasCreator untuk tracking created_by/updated_by. Ref: SOX Compliance
CTD L-08	<i>Audit Trail</i>	Mediu m	ActivityLogged <i>trait</i> wajib pada model bisnis	Model entitas bisnis HARUS <i>trait</i> ActivityLogged (Spatie) untuk pencatatan perubahan
CTD L-09	Enum Conventio n	Mediu m	Enum status dengan label() dan color()	Field status HARUS PHP 8.1 Enum dengan label():string dan color():string. Implement CitadelEnum



© Hak Cipta milik 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

ID	Domain	Risiko	Judul Kontrol	Deskripsi & Referensi
CTD L-10	Database Conventions	Low	Custom table name dengan prefix escm_	Semua model HARUS \$table property eksplisit dengan prefix escm_. Tidak boleh auto naming
CTD L-11	Authorization	High	Permission check via checkAccessPermission()	Setiap <i>controller method</i> bisnis HARUS checkAccessPermission(\$permission) di awal. Format: feature:action. Ref: ASVS V4
CTD L-12	Architecture	Medium	Aksi non-CRUD sebagai <i>method</i> tersendiri	Operasi bisnis (propose/release/sync/approve/esign) HARUS <i>method</i> tersendiri bukan generic update
CTD L-13	Citadel Framework	High	Schema component harus implement Backbone	Komponen dalam schema([...]) HARUS implement Backbone interface. Pelanggaran = error BACKBONE[001]
CTD L-14	Citadel Framework	Low	Makeable pattern untuk factory instantiation	Citadel component HARUS via ::make(\$name, \$title). Dilarang new ComponentClass() langsung
CTD L-15	Document Numbering	Medium	HasNumbering trait untuk auto-numbering	Model procurement (Tender/PO/PR/Contract) HARUS <i>trait</i> HasNumbering + interface UseNumbering



© Hak Cipta milik 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

ID	Domain	Risiko	Judul Kontrol	Deskripsi & Referensi
WIKA-Q01	Architectur e	Low	Tidak ada Observer <i>class</i> - semua di boot <i>method</i>	Project TIDAK pakai Observer <i>class</i> . Event handling via boot()/bootTraitName() di <i>trait</i>
WIKA-Q02	Validation	Mediu m	Tidak ada Form Request <i>class</i> - validasi <i>inline</i>	TIDAK pakai Form Request <i>class</i> . Semua validasi <i>inline</i> \$request->validate() di <i>controller</i>
WIKA-Q03	Mass Assignmen t	Mediu m	\$guarded bukan \$fillable	Standar: \$guarded=["id"] bukan \$fillable. Semua field selain id bisa mass-assign
WIKA-Q04	API Response	Low	Response pattern: apiRes() helper untuk JSON	Response JSON HARUS apiRes('success'/'error', ...). Form sukses: Citadel Swal::make('success')
WIKA-Q05	Integration	High	Sync mechanism mandatory untuk data eksternal	Data dari SAP/HCMS/VMS/Pengada an.com HARUS punya <i>sync service</i> dedicated di app/Sync/Service/
WIKA-Q06	Integration	Mediu m	Sync <i>service</i> harus extend Base	Sync <i>service</i> HARUS extend App\Sync\Service\Base. Implement: fetching(), handleResponseForCache(), saving()
WIKA-Q07	Authorizati on	High	Scope-based data access <i>multi-tenant</i>	Akses data procurement HARUS ESCM::scopeProject(auth()->user()). Ref: CWE-639 IDOR



© Hak Cipta milik 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

ID	Domain	Risiko	Judul Kontrol	Deskripsi & Referensi
WIKA-Q08	Logging Security	High	Password/PIN masking di semua log	Middleware logging HARUS masking: <i>password</i> , <i>password_confirm</i> , <i>pass</i> , <i>pin</i> . Ref: CWE-532
WIKA-Q09	Database Conventio n	Low	SoftDeletes hanya untuk entitas tertentu	SoftDeletes HANYA pada <i>User/Vendor/VendorQuotati</i> on. Mayoritas model pakai status flag
WIKA-Q10	Database Conventio n	Mediu m	Relationship key menggunakan <i>business key</i>	Beberapa relationship pakai <i>business key</i> (nomor dokumen) bukan surrogate key. Konsistensi SAP
WIKA-Q11	Workflow	High	<i>Approval workflow</i> via <i>morphMany</i> pattern	Model <i>approval</i> HARUS implement <i>ApprovableContract</i> + <i>trait ReleaseStrategyModel</i> . <i>Polymorphic morphMany</i>
WIKA-Q12	Architectur e	Mediu m	Procurement module interface wajib	Entitas <i>procurement</i> HARUS implement <i>ProcurementModule</i> dengan <i>getProcurementNumber()</i> dan <i>historyModelName()</i>
WIKA-Q13	Routing	Low	Explicit route definition, bukan <i>Route::resource()</i>	TIDAK pakai <i>Route::resource()</i> . Semua route eksplisit per <i>method</i> untuk kontrol granular
WIKA-Q14	Authentica tion	High	Multi-layer authentication	3 layer: Session (<i>cek_login</i>) + <i>BasicAuth</i> + <i>IP</i> (<i>basic_auth</i>) + <i>JWT</i>



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

ID	Domain	Risiko	Judul Kontrol	Deskripsi & Referensi
				(jwt.verify). Ref: <i>Defense in Depth</i>
WIKI-Q15	Security	High	CSRF protection tanpa exception	CSRF aktif untuk semua POST/PUT/DELETE. VerifyCsrfToken \$except HARUS kosong. Ref: CWE-352

4.2.3 Perancangan Instruksi (*Prompt Engineering*)

Perancangan Instruksi (*Prompt Engineering*) dibuat supaya model OpenAI GPT-4o-mini dapat memahami dan mengimplementasikan 30 kontrol keamanan dan arsitektur yang telah disusun, diperlukan teknik rekayasa instruksi (*prompt engineering*). Tahapan ini berfungsi untuk membatasi ruang lingkup interpretasi mesin, mencegah terjadinya halusinasi data, dan memastikan bahwa setiap tinjauan kode selalu relevan dengan konteks proyek WISE/ESCM.

Struktur instruksi pada sistem ini dirancang menggunakan pendekatan instruksi berlapis (*layered prompting*)(Sanwal, n.d.), yang terdiri atas tiga komponen pembentuk utama pada *System Prompt*, yaitu:

1) Pemberian Peran (*Role-Playing*)

GPT 4o-mini secara tegas diinstruksikan untuk bertindak sebagai *Senior Code Reviewer* melalui kalimat instruksi berikut: "*You are a senior security-focused reviewer evaluating a merge request diff that targets the WISE/ESCM project an Enterprise Supply Chain Management system for PT Wijaya Karya (WIKI).*" sebagaimana ditunjukkan pada Tabel 7.. Pemilihan kata "*senior*" dan "*security-focused*" dilakukan secara sengaja, karena tanpa persona ini, LLM cenderung menjawab dengan gaya *reviewer* pemula yang memberi saran bersifat umum. Dengan persona ini, AI diarahkan untuk langsung berfokus pada temuan yang relevan dengan risiko keamanan dan kepatuhan terhadap arsitektur khas WISE/ESCM, seperti kerangka kerja Citadel



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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

2) Injeksi Konteks Aturan (*Context Injection*)

Model AI yang dipakai di penelitian ini awalnya hanya belajar dari dokumentasi umum yang tersebar di internet. Karena itu, kalau dibiarkan begitu saja, AI cenderung menyarankan cara-cara umum yang belum tentu cocok, bahkan bisa bertentangan dengan aturan internal khusus milik perusahaan.

Untuk mengatasi hal ini, seluruh isi buku pedoman (*rulebook*) yang berjumlah 30 aturan (Tabel 6.) dimasukkan langsung ke dalam instruksi yang diberikan kepada AI, setiap kali AI akan memeriksa kode sebagaimana ditunjukkan pada bagian berlabel [*CONTEXT INJECTION: REVIEW RULEBOOK*] di Tabel 7.. Dengan begitu, AI selalu diingatkan tentang aturan-aturan khusus perusahaan ini setiap kali *run*, bukan mengandalkan ingatannya sendiri yang bisa jadi salah atau tidak lengkap.

Supaya aturan-aturan ini benar-benar dipatuhi dan tidak dianggap sekadar saran biasa, setiap aturan ditulis menggunakan kata wajib "*MUST*" dalam huruf kapital. Sebagai contoh, ada instruksi yang berbunyi "*This project does NOT use Form Request classes*" untuk melarang penggunaan kelas *Form Request* (WIKI-Q02), yaitu kelas validasi terpisah yang biasanya dianggap standar di Laravel, namun sengaja tidak dipakai di proyek ini. Ada juga instruksi "*All database access MUST use Eloquent ORM*" yang mewajibkan AI menyarankan penggunaan Eloquent ORM, yaitu cara bawaan Laravel yang lebih aman untuk mengambil data dari basis data dibandingkan menulis perintah *database* secara langsung dan manual (*raw query*/CTDL-01) yang lebih berisiko.

Dengan gaya penulisan *prompt* yang tegas dan berulang seperti ini, AI diarahkan untuk memperlakukan buku pedoman perusahaan ini sebagai acuan yang paling utama dan tidak boleh dilanggar, dan menghilangkan kebiasaan umum yang sudah dipelajarinya sebelumnya dari internet.

3) Penyeragaman Format Keluaran (*Output Formatting*)

Untuk memastikan hasil tinjauan dari API OpenAI dapat dibaca, diproses (*parsing*), dan dirender oleh sistem *backend* menjadi antarmuka web yang interaktif, AI



© Hak Cipta milik 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

dilarang memberikan balasan berupa teks bebas (*free-text*). Model diinstruksikan dengan ketat untuk selalu mengembalikan jawaban dalam format *JavaScript Object Notation* (JSON). Skema JSON tersebut diwajibkan memuat atribut spesifik yang seragam, meliputi: nama berkas, nomor baris yang terdeteksi melanggar aturan, ID kontrol, tingkat keparahan (*severity*), dan rekomendasi perbaikan kode.

Struktur instruksi yang menggabungkan ketiga komponen tersebut diimplementasikan ke dalam kode sistem. Ringkasan inti dari instruksi sistem tersebut dapat dilihat pada Tabel 7., sedangkan naskah lengkap dari instruksi pemodelan (*system prompt*) tersebut dilampirkan secara utuh pada Lampiran 5

Tabel 7. Ringkasan inti instruksi sistem ((*Prompt Engineering*))





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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Codex Merge Request Diff Review Prompt WISE/ESCM

[ROLE-PLAYING] You are a senior security-focused reviewer evaluating a merge request diff that targets the WISE/ESCM project an Enterprise Supply Chain Management system for PT Wijaya Karya (WIKI).

[CONTEXT INJECTION: REVIEW RULEBOOK] When reviewing this diff, enforce these conventions extracted from the WISE/ESCM codebase:

- *CTDL-01 High: All database access MUST use Eloquent ORM. Flag any DB::raw(), DB::select(), DB::statement() unless it's for complex aggregation.*
- *WIKI-Q02 Medium: This project does NOT use Form Request classes validation is inline. Every store() and update() method MUST have \$request->validate(...).*
- *WIKI-Q08 High: Logging middleware MUST mask sensitive fields: password, password_confirm, pass, pin. (Note: Additional rules applied in backend...)*

[OUTPUT FORMATTING] Return JSON with the exact structure below:

```
{ "issues": [ { "description": "Human readable explanation referencing specific hunks", "check_name": "Short identifier. Include rule ID e.g. 'SAST | [CTDL-02]'", "severity": "info|minor|major|critical|blocker", "location": { "path": "relative/path.ext", "lines": { "begin": 42 } }, "recommendation": "Actionable fix or mitigation" } ] }
```

[CRITICAL CONSTRAINTS] No Duplicate Findings: Do NOT report the same rule ID + same file + same line more than once.

4.2.4 Analisis Kebutuhan Antarmuka Web

Mengingat penelitian ini berjudul rancang bangun, sistem yang dikembangkan wajib menghasilkan produk akhir yang dapat dioperasikan secara langsung.



© Hak Cipta milik 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Meskipun AI nya tertanam di sisi backend, kehadiran GUI tetap diperlukan sebagai medium yang interaktif. GUI ini bertugas menjadi penghubung API OpenAI yang tidak *user friendly* menjadi sebuah sistem yang mudah digunakan oleh pengembang.

Selain sebagai medium yang interaktif, GUI memegang peran penting dalam memvisualisasikan hasil pemeriksaan kode. Keluaran asli dari model kecerdasan buatan (LLM) pada dasarnya berupa data mentah dalam format JSON. Melalui antarmuka ini, data mentah tersebut diolah menjadi laporan visual yang jauh lebih informatif. Pengembang dapat **langsung** melihat rincian tingkat keparahan beserta *highlight* pada baris kode yang terindikasi melanggar aturan.

Lebih lanjut, GUI ini menjadi semacam lingkungan sandbox atau ruang virtual yang terisolasi dan aman untuk menguji coba sebuah software. Seluruh GUI bersifat sandbox karena tidak menyengol sama sekali repository WISE.

4.3 Implementasi Sistem

Tahap implementasi merupakan tahapan untuk merealisasikan seluruh rancangan arsitektur, modul, dan komponen logika yang telah disusun ke dalam bentuk perangkat lunak yang fungsional. Implementasi sistem ini secara garis besar dibagi menjadi dua komponen utama, yaitu pengembangan Pemodelan Deteksi AI (*review engine*) pada backend serta pembuatan antarmuka pengguna interaktif berbasis web (frontend).

4.3.1 Implementasi Pemodelan Deteksi AI (*Review Engine*)

Mesin utama untuk melakukan deteksi pada sistem ini berada pada modul `reviewEngine.mjs`. Fungsi ini berperan sebagai inti dari proses pemeriksaan kode. Tugasnya adalah menyiapkan kode yang dikirim pengguna, membaginya jika terlalu panjang, mengirimkannya ke AI, lalu menyaring kembali hasil temuan sebelum ditampilkan pada halaman web. Dengan cara ini, sistem tidak hanya mengirim kode langsung ke AI, tetapi juga mengatur alurnya agar hasil yang keluar lebih rapi dan dapat dipertanggungjawabkan.

1) Persiapan dan Penyeragaman Masukan Kode



© Hak Cipta milik Politeknik Negeri Jakarta

Tahap pertama yang dilakukan sistem adalah menyiapkan kode yang akan diperiksa. Kode dari pengguna tidak langsung diberikan ke AI, melainkan diubah terlebih dahulu ke dalam bentuk *synthetic diff*. *Synthetic diff* adalah format buatan yang menyerupai perubahan kode, sehingga sistem tetap dapat mengetahui nama file dan nomor baris dari kode yang diperiksa. (Du and Shepotylo, 2021)

Proses ini penting karena hasil temuan AI harus bisa ditunjukkan kembali ke bagian kode yang tepat. Jika AI menemukan masalah, sistem dapat menampilkan lokasi file dan baris yang sesuai. Pada sistem ini, proses tersebut dilakukan melalui fungsi `codeToSyntheticDiff()` dan `buildSyntheticDiffFromFiles()`. Cuplikan kode proses perubahan kode menjadi *synthetic diff* dapat dilihat pada Gambar 4.8.



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



© Hak Cipta milik 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

escm-codex-review - reviewEngine.mjs

440 export function codeToSyntheticDiff(code, fileName = "submitted_snippet.php") {
441   const safeName = String(fileName || "submitted_snippet.php").replace(
442     /^[\./]+/,
443     "",
444   );
445   const lines = String(code).replace(/\r\n/g, "\n").split("\n");
446   // Buang baris kosong terakhir akibat trailing newline agar count akurat
447   if (lines.length && lines[lines.length - 1] === "") lines.pop();
448   const count = lines.length || 1;
449   const body = (lines.length ? lines : []).map((l) => `${l}`).join("\n");
450   return (
451     `diff --git a/${safeName} b/${safeName}\n` +
452     `new file mode 100644\n` +
453     `index 0000000..1111111\n` +
454     `--- /dev/null\n` +
455     `+++ b/${safeName}\n` +
456     `@@ -0,0 +1,${count} @@\n` +
457     `${body}\n`
458   );
459 }

460
461 export function buildSyntheticDiffFromFiles(files = [], options = {}) {
462   const maxFiles = Number(options.maxFiles ?? 5);
463   const allowedExtensions = Array.isArray(options.allowedExtensions)
464     ? options.allowedExtensions.map((ext) => String(ext).toLowerCase())
465     : [".php"];
466
467   const accepted = [];
468   const rejected = [];
469   const normalizedFiles = Array.isArray(files) ? files : [];
470
471   for (const file of normalizedFiles.slice(0, maxFiles)) {
472     const name = String(file?.name || "").trim();
473     if (!name) continue;
474     const ext = path.extname(name).toLowerCase();
475     if (!allowedExtensions.includes(ext)) {
476       rejected.push({ name, reason: "Ekstensi tidak didukung." });
477       continue;
478     }
479     accepted.push({ name, content: String(file?.content ?? "") });
480   }
481
482   const diffChunks = accepted.map((file) =>
483     codeToSyntheticDiff(file.content, file.name),
484   );
485   return {
486     files: accepted,
487     rejected,
488     diffText: diffChunks.join("\n\n"),
489   };
490 }

```

Gambar 4.9 Cuplikan kode proses perubahan kode menjadi *synthetic diff*.

Sumber: reviewEngine.mjs baris 440-495.



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

2) Pemecahan Kode Menjadi Beberapa Bagian

Setelah kode siap, sistem akan memeriksa apakah ukuran kode terlalu panjang. Hal ini perlu dilakukan karena AI memiliki batas jumlah teks yang dapat diproses dalam satu kali permintaan. Jika kode terlalu besar, sistem akan membaginya menjadi beberapa potongan yang lebih kecil. (Jaiswal and Milios, n.d.)

Proses pemecahan ini dilakukan agar kode tetap dapat diperiksa tanpa melebihi batas kemampuan model AI. Pada sistem ini, pemecahan kode dilakukan menggunakan fungsi `chunkText()` dan `batchPieces()`. Dengan adanya proses ini, sistem tetap dapat menangani kode yang cukup panjang secara bertahap. Kode proses pemecahan kode menjadi beberapa bagian dapat dilihat pada Gambar 4.9.

DAN

```

escm-codex-review - reviewEngine.mjs
227 function chunkText(label, text, maxChunkChars) {
228   const chunks = [];
229   for (let i = 0; i < text.length; i += maxChunkChars) {
230     const slice = text.slice(i, i + maxChunkChars);
231     chunks.push(`### ${label} (part ${chunks.length + 1})\n\n${slice}`);
232   }
233   return chunks;
234 }

411 export function batchPieces(piecesArr, label, maxChunkChars) {
412   const out = [];
413   let buffer = "";
414   let labelIndex = 0;
415   for (const piece of piecesArr) {
416     const sep = buffer ? "\n\n---\n\n" : "";
417     if (buffer.length + sep.length + piece.length > maxChunkChars) {
418       if (buffer) {
419         out.push(`### ${label} ${labelIndex}\n\n${buffer}`);
420         buffer = "";
421       }
422     }
423     buffer += (buffer ? sep : "") + piece;
424   }
425   if (buffer) out.push(`### ${label} ${labelIndex}\n\n${buffer}`);
426   return out;
427 }
  
```

Gambar 4.10 Kode proses pemecahan kode menjadi beberapa bagian

Sumber: reviewEngine.mjs baris 227-234 dan 411-430.

3) Pemanggilan AI dengan Format Keluaran Terstruktur

Tahap berikutnya adalah mengirim potongan kode ke model AI untuk diperiksa. Proses ini dilakukan melalui fungsi `callOpenAI()`. Pada tahap ini, sistem



© Hak Cipta milik 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 mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

mengirimkan kode beserta instruksi atau prompt agar AI dapat mencari potensi masalah keamanan pada kode tersebut.

Agar hasil dari AI tidak berbentuk jawaban bebas, sistem menggunakan struktur keluaran khusus bernama `reviewSchema`. Struktur ini mengatur agar setiap temuan memiliki data yang jelas, seperti deskripsi masalah, aturan yang dilanggar, tingkat keparahan, lokasi file, nomor baris, dan saran perbaikan. Dengan cara ini, hasil AI menjadi lebih mudah diproses oleh sistem dan lebih mudah ditampilkan kepada pengguna. Kode struktur keluaran AI pada `reviewSchema` dapat dilihat pada Gambar 4.10.

```

escm-codex-review - reviewEngine.mjs
44 export const reviewSchema = {
45   type: "object",
46   additionalProperties: false,
47   required: ["issues", "summary_markdown"],
48   properties: {
49     issues: {
50       type: "array",
51       items: {
52         type: "object",
53         additionalProperties: false,
54         required: [
55           "description",
56           "check_name",
57           "severity",
58           "location",
59           "confidence",
60           "categories",
61           "fingerprint",
62           "references",
63           "recommendation",
64           "tests",
65           "notes",
66         ],
67         properties: {
68           description: { type: "string" },
69           check_name: { type: "string" },
70           severity: {
71             type: "string",
72             enum: ["info", "minor", "major", "critical", "blocker"],
73           },
74           confidence: {
75             type: "string",
76             enum: ["low", "medium", "high"],
77           },
78           categories: { type: "array", items: { type: "string" } },
79           fingerprint: { type: "string" },
80           location: {
81             type: "object",
82             additionalProperties: false,
83             required: ["path", "lines"],
84             properties: {
85               path: { type: "string" },
86               lines: {
87                 type: "object",
88                 additionalProperties: false,
89                 required: ["begin"],
90                 properties: { begin: { type: "integer" } },
91               },
92             },
93           },
94           references: { type: "array", items: { type: "string" } },
95           recommendation: { type: "string" },
96           tests: { type: "array", items: { type: "string" } },
97           notes: { type: "string" },
98         },
99       },
100     },
101     summary_markdown: { type: "string" },
102   },
103 };

```

Gambar 4.11 Kode struktur keluaran AI pada `reviewSchema`



© Hak Cipta milik 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Sumber: reviewEngine.mjs baris 44-101.

4) Penyaringan dan Penyesuaian Hasil Temuan

Setelah AI mengembalikan hasil pemeriksaan, sistem tidak langsung menampilkan semua temuan tersebut. Hasil dari AI masih diperiksa kembali oleh sistem. Proses ini dilakukan pada fungsi `runReviewTasks()`.

Pada tahap ini, sistem menyaring temuan yang tidak valid, menghapus temuan yang muncul berulang, dan memastikan bahwa lokasi file yang disebutkan AI memang benar-benar ada pada kode yang diperiksa. Proses ini penting untuk mengurangi kemungkinan AI memberikan temuan yang tidak sesuai dengan kode asli.

Selain itu, sistem juga menyesuaikan tingkat keparahan temuan menggunakan fungsi `applySeverityFloorFromControls()`. Fungsi ini memastikan tingkat keparahan tiap temuan AI tidak diremehkan dari batas yang telah ditetapkan berdasarkan tingkat risiko aturan yang dilanggar pada *rulebook* (Subbab 4.2.2), dengan pemetaan sebagai berikut tertera pada Tabel 8..

Tabel 8. Pemetaan Tingkat Risiko Aturan terhadap Tingkat Keparahannya Temuan AI

Tingkat Risiko Aturan (<i>Rulebook</i>)	Kisaran Tingkat Keparahannya Temuan AI
<i>High</i>	major atau critical (wajib minimal major)
<i>Medium</i>	major atau minor
<i>Low</i>	minor atau info

Sumber: Instruksi Pemodelan AI (Lampiran 4), diolah oleh penulis

```
export async function runReviewTasks({
  tasks,
  knownFiles,
  controlCatalog,
```



© Hak Cipta milik 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

concurrency,
openAiConfig,
onProgress = () => {},
log = () => {},
})) {
  const aggregated = [];
  const summaries = [];
  const seen = new Map();
  let hallucinatedCount = 0;

  let next = 0;
  let done = 0;
  const total = tasks.length;
  const workers = Math.max(1, Math.min(concurrency,
total || 1));
  // Lapor total chunk di awal (untuk progress bar GUI).
  CLI memakai default
  // no-op sehingga perilakunya tidak berubah.
  onProgress(0, total);

  async function worker() {
    while (true) {
      const i = next++;
      if (i >= total) break;
      const { prompt, payload } = tasks[i];
      try {
        const result = await callOpenAI(prompt, payload,
i, {
          ...openAiConfig,
          log,
        });
        const issues = Array.isArray(result?.issues) ?
result.issues : [];
        for (const issue of issues) {
          const normalised
          =
          applySeverityFloorFromControls(

```



© Hak Cipta milik 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

        normaliseIssue(issue),
        controlCatalog.riskById,
    );
    // Strict locations: buang minor/info tanpa
path valid
    if (
        !normalised.location ||
        typeof normalised.location.path !==
"string" ||
        !normalised.location.path.trim()
    ) {
        const sev = String(normalised.severity ||
"").toLowerCase();
        if (sev === "minor" || sev === "info") {
            continue;
        }
        normalised.location = {
            path: "unknown",
            lines: { begin: 1 },
        };
    }
    // Hallucination guard: buang temuan untuk
file tak dikenal
    const fPath = (normalised.location?.path ||
"").replace(/^\\+/, "");
    if (fPath && fPath !== "unknown" &&
knownFiles.size > 0) {
        let matched = knownFiles.has(fPath);
        if (!matched) {
            for (const kf of knownFiles) {
                if (
                    kf.endsWith(fPath) ||
                    fPath.endsWith(kf) ||
                    (kf.endsWith("/") +
path.basename(fPath)) &&

```



© Hak Cipta milik 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

        path.basename(fPath)
    path.basename(kf))
    ) {
        matched = true;
        if      (kf.endsWith(fPath)
fPath.endsWith(kf)) {
            normalised.location.path
kf.endsWith(fPath) ? kf : fPath;
        }
        break;
    }
    }
    if (!matched) {
        log(
            `[codex]      HALLUCINATION      filtered:
"${fPath}" not in known files`,
        );
        hallucinatedCount++;
        continue;
    }
    // Dedup by content (rule + path + line)
    const ruleMatch = (normalised.check_name ||
    "").match(
        /\s*([A-Z]+-[A-Z0-9]+)\s*/i,
    );
    const ruleId = ruleMatch
        ? ruleMatch[1].toUpperCase()
        : (normalised.check_name || "").slice(0,
60);
    const dedupPath = normalised.location?.path ||
    "unknown";
    const          dedupLine
normalised.location?.lines?.begin || 0;

```



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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

const dedupKey =
`${ruleId}|${dedupPath}|${dedupLine}`;
if (!seen.has(dedupKey)) {
  aggregated.push(normalised);
  seen.set(dedupKey, true);
} else {
  log(`[codex] DEDUP filtered: ${dedupKey}`);
}
}
if (result?.summary_markdown) {
  summaries.push(result.summary_markdown.trim());
}
} catch (err) {
  log(`[codex] chunk ${i + 1} failed:
${err.message}`);
}
done++;
onProgress(done, total);
}
}

const pool = Array.from({ length: workers }, () =>
worker());
await Promise.all(pool);

return { aggregated, summaries, hallucinatedCount };
}

```

Gambar 4.12 Kode proses penyaringan hasil dan penyesuaian severity

Sumber: reviewEngine.mjs baris 913-1033

5) Pengendali Utama Proses *Review*

Seluruh proses di atas dikendalikan oleh fungsi utama bernama `reviewSource()`. Fungsi ini menjadi pintu masuk utama dari *review engine*. Di dalamnya, sistem menerima kode dari pengguna, mengubahnya menjadi format yang sesuai,



© Hak Cipta milik 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

membagi kode jika diperlukan, menjalankan pemeriksaan AI, lalu mengembalikan hasil akhir yang sudah dirapikan.

Dengan adanya fungsi ini, alur kerja *review engine* menjadi lebih terpusat dan mudah dikendalikan. Semua tahapan pemeriksaan berjalan secara berurutan, mulai dari persiapan kode sampai hasil akhir siap dikirim ke backend dan ditampilkan pada antarmuka web.

```
export async function reviewSource({
  source,
  inputType = "code",
  prompt,
  fileName = "submitted_snippet.php",
  controlCatalog,
  openAiConfig,
  maxChunkChars = 120000,
  concurrency = 4,
  diffExtsRaw = "",
  onProgress = () => {},
  log = () => {},
}) {
  // 1) Normalisasi input menjadi teks diff.
  const diffText =
    inputType === "code"
      ? codeToSyntheticDiff(source, fileName)
      : String(source || "");

  // 2) knownFiles dari header diff (sama seperti
  run_codex.mjs mode diff).
  const knownFiles = new Set();
  const diffPathRegex = /^\\+\\+\\+\\s+b\\/(.+)$/gm;
  let m;
  while ((m = diffPathRegex.exec(diffText)) !== null) {
    knownFiles.add(m[1]);
  }
}
```



© Hak Cipta milik 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
// 3) Pecah jadi pieces lalu batch jadi chunks.
const res = gatherDiffPieces({ diffText,
maxChunkChars, diffExtsRaw, log });
const pieces = Array.isArray(res) ? res : res.pieces;
const chunks = batchPieces(pieces, "Batch",
maxChunkChars);

// 4) Jalankan review (OpenAI + guard + dedup).
const tasks = chunks.map((payload) => ({ prompt,
payload }));
const { aggregated, summaries, hallucinatedCount } =
await runReviewTasks({
  tasks,
  knownFiles,
  controlCatalog,
  concurrency,
  openAiConfig,
  onProgress,
  log,
});

// 5) Statistik ringkas untuk ditampilkan di GUI.
const counts = { blocker: 0, critical: 0, major: 0,
minor: 0, info: 0 };
for (const it of aggregated) {
  const sev = (it.severity || "minor").toLowerCase();
  if (counts[sev] !== undefined) counts[sev]++;
}
const overallRisk =
  counts.blocker > 0
    ? "blocker"
    : counts.critical > 0
    ? "critical"
    : counts.major > 0
    ? "major"
    : counts.minor > 0
```



© Hak Cipta milik 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

      ? "minor"
      : "info";

const triggeredControlIds = new Set();
for (const issue of aggregated) {
  for (const id of collectControlIdsFromIssue(issue))
  {
    if (
      controlCatalog.riskById.size === 0 ||
      controlCatalog.riskById.has(id)
    ) {
      triggeredControlIds.add(id);
    }
  }
}

return {
  issues: aggregated,
  summaries,
  hallucinatedCount,
  counts,
  overallRisk,
  triggeredControlIds:
Array.from(triggeredControlIds).sort(),
  knownFiles: Array.from(knownFiles),
};
}

```

Gambar 4.13 Kode fungsi reviewSource sebagai pengendali utama *review engine*
 Sumber: reviewEngine.mjs baris 1051-1125.

4.3.2 Implementasi Antarmuka Web

Antarmuka web pada sistem ini dibuat menggunakan React dan TypeScript. Antarmuka ini berfungsi sebagai tempat pengguna mengunggah kode, menjalankan



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

proses pemindaian, melihat progres pemeriksaan, dan membaca hasil laporan dari AI. Dengan adanya antarmuka web, proses *review* kode menjadi lebih mudah digunakan karena pengguna tidak perlu menjalankan perintah melalui terminal.

1) Pengaturan Tampilan Utama dan Alur Pengguna

Halaman utama sistem diatur melalui komponen `App.tsx`. Komponen ini menjadi pengendali utama pada sisi frontend. Di dalamnya, sistem menyimpan data file yang diunggah, status proses pemindaian, hasil laporan terakhir, serta riwayat pemeriksaan sebelumnya.

Saat pengguna memilih berkas, sistem akan menyimpannya ke dalam *state* `uploadedFiles`. Setelah itu, pengguna dapat mengeksekusi pemindaian dan hasilnya akan diteruskan ke halaman laporan. Potongan kode untuk alur utama ini ditunjukkan pada Gambar 4.13, sedangkan hasil tampilan antarmuka halaman utama dapat dilihat pada Gambar 4.14.

```

38 export default function App() {
39   const [health, setHealth] = useState<HealthResponse | null>(null);
40   const [status, setStatus] = useState<RequestStatus>("idle");
41   const [error, setError] = useState<string | null>(null);
42   const [current, setCurrent] = useState<EvalResponse | null>(null);
43   const [history, setHistory] = useState<HistorySummary[]>([]);
44   const [activeId, setActiveId] = useState<string | null>(null);
45   // Bedakan jenis loading: "live" (scan AI, pakai bar) vs "load" (muat
46   // rekaman/riwayat, instan + banner ringkas).
47   const [loadingKind, setLoadingKind] = useState<"live" | "load" | null>(null);
48   // Fase keluar bar scan live: bar menyusut ke tengah & memudar SEBELUM hasil
49   // muncul (lihat runLive).
50   const [scanExiting, setScanExiting] = useState(false);
51   // File yang diunggah pengguna (upload bebas) + isinya (dibaca di browser).
52   const [uploadedFiles, setUploadedFiles] = useState<
53     { name: string; content: string }[]
54   >([]);
55   const [uploadError, setUploadError] = useState<string | null>(null);
56   // true selama FileReader membaca isi file (umpan balik "Membaca file...").
57   const [reading, setReading] = useState(false);
58   const fileInputRef = useRef<HTMLInputElement | null>(null);
59   const [scanProgress, setScanProgress] = useState<ScanProgressData | null>(
60     null,
61   );
62   const [glossaryOpen, setGlossaryOpen] = useState(false);
63   const [uatOpen, setUatOpen] = useState(false);
64
65   useEffect(() => {
66     getHealth()
67       .then(setHealth)
68       .catch(() => setHealth(null));
69     getHistory()
70       .then(setHistory) // You, 5 days ago • ubah konsep jadi upload kode bebas tapi terikan
71       .catch(() => setHistory([]));
72   }, []);
73
74   // Konfirmasi sebelum refresh/keluar saat: scan live SEDANG berjalan, ATAU
75   // hasil live sedang ditampilkan. (Teks dialog mengikuti bawaan browser -
76   // tidak bisa dikustom.)
77   useEffect(() => {
78     const guard =
79       (status === "loading" && loadingKind === "live") ||
80       current?.source === "live";
81     if (!guard) return;
82     const onBeforeUnload = (e: BeforeUnloadEvent) => {
83       e.preventDefault();
84       e.returnValue = "";
85     };

```

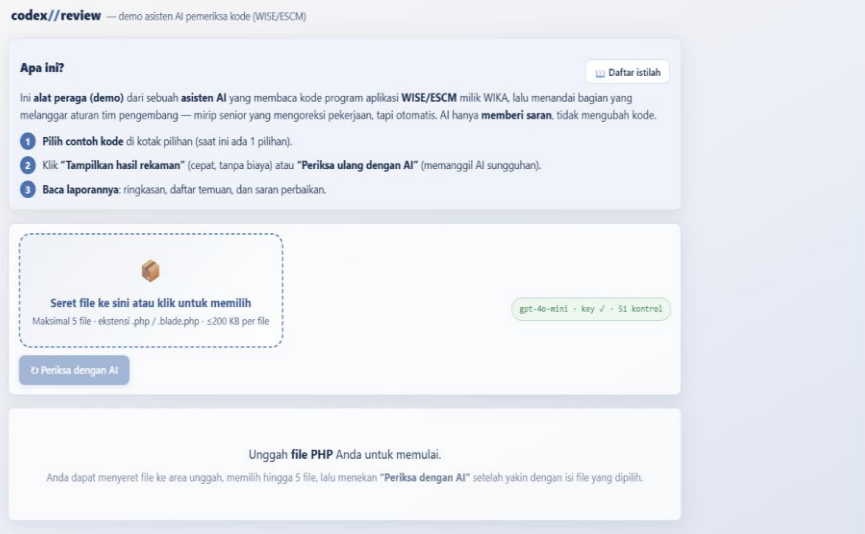


Gambar 4.14 Kode pengaturan alur utama antarmuka web

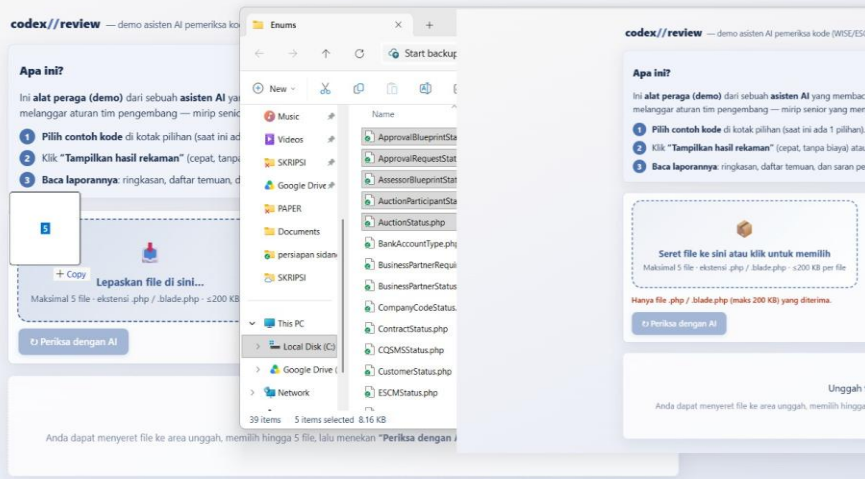
Sumber: App.tsx

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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



(a)



(b)

Gambar 4.15 Tampilan Area Unggah Berkas Kode PHP

2) Pengiriman File ke Backend

Agar frontend dapat berkomunikasi dengan backend, sistem menggunakan modul `api.ts`. Semua proses pengiriman dan pengambilan data dari server dipusatkan pada file ini. Salah satu fungsi utamanya adalah `postEvaluateStream()`, yaitu fungsi yang mengirim file pengguna ke `endpoint /api/evaluate/stream`.



© Hak Cipta milik Politeknik Negeri Jakarta

Fungsi ini tidak hanya menunggu hasil akhir, tetapi juga membaca data *progress* yang dikirim backend secara bertahap. Dengan cara ini, frontend dapat mengetahui apakah proses masih berjalan, file sedang diperiksa, atau laporan sudah selesai dibuat.



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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



© Hak Cipta milik 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

escm-codex-review - api.ts

70 export async function postEvaluateStream(
71   files: { name: string; content: string }[],
72   onProgress: (p: ScanProgress) => void,
73 ): Promise<EvalResponse> {
74   const res = await fetch(`${BASE}/api/evaluate/stream`, {
75     method: "POST",
76     headers: {
77       "Content-Type": "application/json",
78       ...accessHeaders(),
79     },
80     body: JSON.stringify({ files }),
81   });
82   if (!res.ok || !res.body) {
83     const err = await parseJsonSafe<ApiError>(res).catch(() => null);
84     throw new ApiHttpError(
85       res.status,
86       err?.error || `Evaluasi gagal (HTTP ${res.status}).`,
87     );
88   }
89   const reader = res.body.getReader();
90   const decoder = new TextDecoder();
91   let buf = "";
92   let final: EvalResponse | null = null;
93
94   const handleLine = (line: string) => {
95     const t = line.trim();
96     if (!t) return;
97     let msg: { type?: string; pct?: number; label?: string; error?: string };
98     try {
99       msg = JSON.parse(t);
100     } catch {
101       return;
102     }
103     if (msg.type === "progress") {
104       onProgress({
105         pct: Number(msg.pct) || 0,
106         label: String(msg.label) || "",
107       });
108     } else if (msg.type === "result") {
109       final = msg as unknown as EvalResponse;
110     } else if (msg.type === "error") {
111       throw new Error(msg.error || "Evaluasi gagal.");
112     }
113   };
114
115   for (;;) {
116     const { done, value } = await reader.read();
117     if (done) break;
118     buf += decoder.decode(value, { stream: true });
119     let nl: number;
120     while ((nl = buf.indexOf("\n")) >= 0) {
121       handleLine(buf.slice(0, nl));
122       buf = buf.slice(nl + 1);
123     }
124   }
125   handleLine(buf); // baris terakhir tanpa newline penutup
126   if (!final) throw new Error("Stream selesai tanpa hasil.");
127   return final;
128 }

```

Gambar 4.16 Kode pengiriman file dan pembacaan *streaming progress*

Sumber: api.ts baris 70-125



3) Tampilan Progres Pemindaian

Mengingat proses tinjauan AI dapat memakan waktu, sistem menampilkan indikator progres melalui komponen `ScanProgress`. Komponen ini berfungsi menahan antarmuka agar pengguna mengetahui bahwa sistem sedang bekerja dan tidak mengalami error (*hang*). Informasi yang disajikan meliputi persentase kelengkapan dan keterangan status dinamis, seperti "menyiapkan file", "memeriksa kode", atau "menyusun laporan". Implementasi kode komponen ini ditunjukkan pada Gambar 4.16, dan tampilan progres pemindaian dapat dilihat pada Gambar 4.17.

```
escm-codex-review - ScanProgress.tsx
9  export function ScanProgress({
```

DAN

```
escm-codex-review - App.tsx
268  <ScanProgress
269      exiting={scanExiting}
270      progress={scanProgress?.pct}
271      label={scanProgress?.label}
272  />
```

Gambar 4.17 Kode komponen *progress scan*

Sumber: `ScanProgress.tsx` baris 9 dan `App.tsx` baris 268-271.

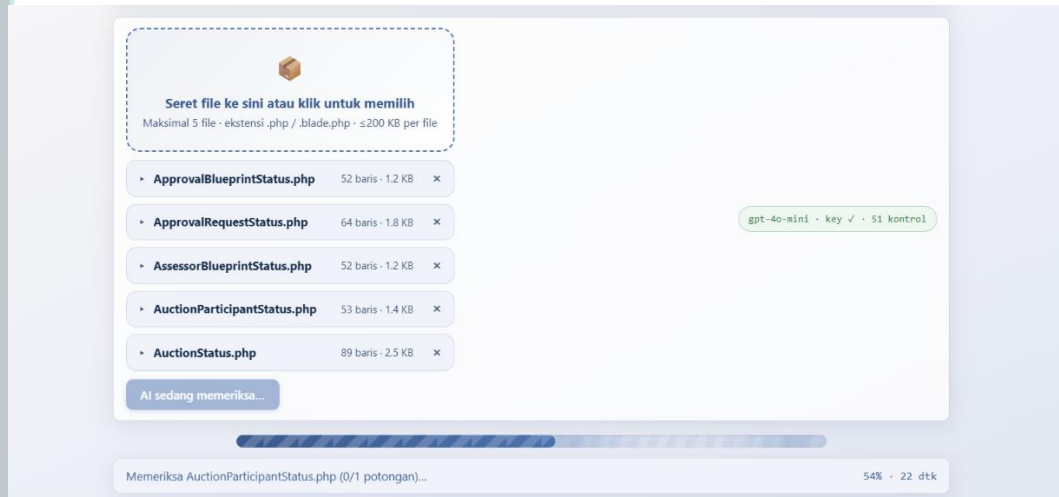
Hak Cipta :

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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Gambar 4.18 Tampilan Progres Pemindaian

4) Penyajian Hasil Laporan *Review*

Setelah backend selesai melakukan pemeriksaan, hasil laporan ditampilkan melalui komponen ReportView. Komponen ini menyusun hasil *review* menjadi beberapa bagian yang lebih mudah dibaca, seperti ringkasan risiko, daftar temuan utama, cuplikan kode, asal aturan, penggunaan token, dan kepatuhan terhadap *rulebook*.

Pada bagian ini, Pengguna dapat memfilter temuan berdasarkan tingkat keparahan (severity). Dengan metode penyajian ini, informasi teknis menjadi jauh lebih mudah dipahami. Implementasi struktur laporan ditunjukkan pada Gambar 4.18, dan hasil akhir antarmuka pelaporannya ditunjukkan pada Gambar 4.19.



© Hak Cipta milik 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

66 function Executive({
67   exec,
68   sev,
69 }): {
70   exec: ReportData["executive"];
71   sev: ReportData["severity"];
72 } {
73   const chips: [Severity, number][] = [
74     ["blocker", sev.blocker],
75     ["critical", sev.critical],
76     ["major", sev.major],
77     ["minor", sev.minor],
78     ["info", sev.info],
79   ];
80   return (
81     <section className="sec" id="nav-ringkasan" data-navlabel="Ringkasan">
82       <SectionTitle sub="Penilaian keseluruhan & saran singkat dari AI.">
83         Ringkasan
84       </SectionTitle>
85       <p className="sec_p">
86         Tingkat risiko keseluruhan:{" "}
87         <RiskBadge risk={exec.overallRisk} />{" "}
88         <InfoHint text="Diambil dari temuan TERPARAH (severity tertinggi) – standar industri (GitLab/SonarQube). Jadi 1 (sangat tinggi)
89       </p>
90       <div className="sevchips">
91         {chips.map(([k, v]) => (
92           <span
93             key={k}
94             className={`sevchip${v === 0 ? " sevchip--zero" : ""}`}
95           >
96             <span className={`.dot dot--${k}`} />
97             {v} {SEV_SHORT[k]}
98           </span>
99         ))}
100     </div>
101     <p className="sec_p sec_p--muted">{exec.guidance}</p>
102   </section>
103 );
104 }

```

DAN

```

escm-codex-review - ReportView.tsx

450 <Executive exec={r.executive} sev={r.severity} />
451 <TopFindings
452   items={r.topFindings}
453   onOpenRulebook={openRulebook}
454 />
455 <ComponentOverview dom={r.componentByDomain} />
456 <UsageCost usage={r.usage} />
457 <Rulebook rb={r.rulebook} />

```

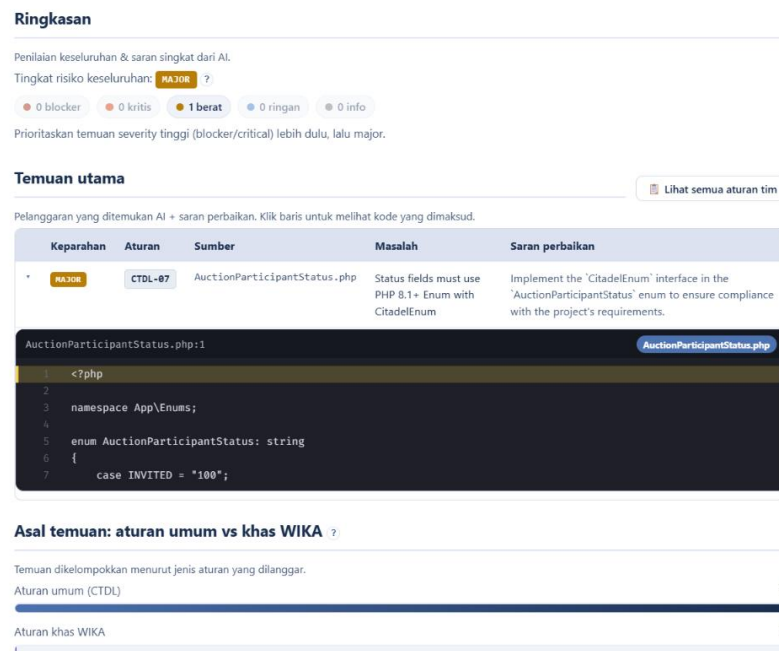
Gambar 4.19 Kode tampilan laporan hasil *review*

Sumber: ReportView.tsx baris 66-390 dan 450-457



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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Gambar 4.20 Tampilan Antarmuka Laporan Interaktif Hasil Tinjauan Kode

5) Tampilan *Rulebook* dan Riwayat Pemeriksaan

Frontend juga menyediakan tampilan *rulebook* melalui komponen RulebookModal. Fitur ini membantu pengguna melihat daftar aturan yang digunakan dalam proses *review*. Selain itu, sistem juga menyediakan komponen RunHistory untuk menampilkan riwayat pemindaian yang pernah dilakukan. Dengan adanya riwayat, pengguna dapat membuka kembali hasil *scan* sebelumnya tanpa harus menjalankan pemeriksaan ulang. Potongan kode untuk fitur ini ditunjukkan pada Gambar 4.20, sedangkan tampilan antarmukanya dapat dilihat pada Gambar 4.21.



© Hak Cipta milik 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

escm-codex-review - RulebookModal.tsx

18 export function RulebookModal({ entries, onClose }: Props) {

escm-codex-review - RunHistory.tsx

28 export function RunHistory({ runs, activeId, onSelect }: Props) {

escm-codex-review - apits

130 export async function getHistory(): Promise<HistorySummary[]> {
131   const d = await parseJsonSafe<{ runs: HistorySummary[] }>(
132     await fetch(`${BASE}/api/history`),
133   );
134   return d.runs;
135 }
136
137 export async function getHistoryRun(id: string): Promise<EvalResponse> {
138   return parseJsonSafe<EvalResponse>(await fetch(`${BASE}/api/history/${id}`));
139 }
140
141 // Rulebook tervalidasi Lead Dev IT WIKA (30 aturan). Tanpa token.
142 export async function getRulebook(): Promise<RulebookEntry[]> {

```

Gambar 4.21 Kode tampilan *rulebook* dan riwayat pemeriksaan

Sumber: RulebookModal.tsx baris 18, RunHistory.tsx baris 28, dan api.ts baris 130-142.

Ringkasan

Penilaian keseluruhan & saran singkat dari AI.
Tingkat risiko keseluruhan: **RAJOR** ?

0 blocker 0 kritis 3 berat 0 ringan 0 info

Prioritaskan temuan severity tinggi (blocker/critical) lebih dulu, lalu major.

Temuan utama Lihat semua aturan tim

Pelanggaran yang ditemukan AI + saran perbaikan. Klik baris untuk melihat kode yang dimaksud.

Keparahan	Aturan	Sumber	Masalah	Saran perbaikan
RAJOR	CTDL-87	ApprovalRequestStatus.php	Status fields must use PHP 8.1+ Enum with CitadelEnum interface	Implement the 'CitadelEnum' interface in the 'ApprovalRequestStatus' enum to ensure compliance with project standards.
RAJOR	CTDL-87	AuctionParticipantStatus.php	Status fields must use PHP 8.1+ Enum with CitadelEnum interface	Implement the 'CitadelEnum' interface in the 'AuctionParticipantStatus' enum.
RAJOR	CTDL-87	AuctionStatus.php	Status fields must use PHP 8.1+ Enum with label() and color() methods	Ensure that the 'AuctionStatus' enum implements the 'CitadelEnum' interface to comply with project standards.

Asal temuan: aturan umum vs khas WIKA ?

Temuan dikelompokkan menurut jenis aturan yang dilanggar.

Jenis Aturan	Jumlah Temuan
Aturan umum (CTDL)	3
Aturan khas WIKA	0

(a)



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Riwayat run (audit) Menampilkan 1-5 dari 36

Kamis, 2 Juli 2026, 02.24 WIB	RAJON	1 temuan
Kamis, 2 Juli 2026, 02.02 WIB	RAJON	3 temuan
Rabu, 1 Juli 2026, 20.03 WIB	RAJON	1 temuan
Rabu, 1 Juli 2026, 19.39 WIB	RAJON	1 temuan
Rabu, 1 Juli 2026, 19.38 WIB	INFO	0 temuan

< Sebelumnya
 1 2 ... 8
 Berikutnya >

(b)

Gambar 4.22 Tampilan Antarmuka Daftar *Rulebook* dan Riwayat Pemeriksaan

4.4 Pengujian

Tahap pengujian pada penelitian ini dilakukan melalui dua pendekatan evaluasi yang saling melengkapi. Pendekatan pertama adalah pengujian fungsional (*black-box testing*) untuk memastikan seluruh fitur antarmuka web dan backend berjalan sesuai spesifikasi kebutuhan. Pendekatan kedua adalah pengujian tingkat kebergunaan dan penerimaan sistem oleh pengguna akhir menggunakan instrumen System Usability Scale (SUS).

4.4.1 Pengujian Fungsional (*Black-box Testing*)

Pengujian black box dilaksanakan secara mandiri oleh penulis. Dari 10 skenario black box yang di uji, 9 skenario mengeluarkan output tidak error, dan hanya ada 1 skenario black box yang bersifat *negative testing*, yaitu *testing* yang bertujuan untuk memastikan bahwa sistem dapat menangani input yang tidak valid, dalam konteks sistem ini adalah penolakan input berkas dengan format selain .php disertai pesan kesalahan (skenario 3 pada Tabel 8.).

Pada pengujian yang dilaksanakan penulis pada 2 Juli 2026, 10 skenario pengujian di buat agar cakupannya lengkap dan sejalan dengan 10 kebutuhan fungsional yang telah ditetapkan pada Subbab 3.4.1, sehingga setiap kebutuhan fungsional dapat diuji dengan skenario yang tepat tanpa ada kebutuhan yang terlewat maupun skenario *testing* tambahan yang tidak jelas asalnya. Rincian skenario beserta hasilnya ditunjukkan pada Tabel 9.



© Hak Cipta milik 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Tabel 9. Hasil *Black-box Testing* (10 Skenario)

No	Kebutuhan Fungsional yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Aktual	Kesimpulan
1	Sistem dapat menampilkan halaman utama antarmuka web.	Membuka antarmuka web dan melihat apakah halaman utama muncul dengan benar.	Halaman utama tampil dengan benar, lengkap dengan area unggah berkas.	Halaman utama tampil dengan benar.	Sesuai
2	Sistem dapat menerima unggahan berkas kode dengan cara drag-and-drop maupun menelusuri berkas (browse).	Mengunggah berkas kode dengan cara menyeret berkas (drag-and-drop) dan dengan cara memilih lewat tombol (browse).	Berkas yang diunggah lewat kedua cara tersebut berhasil masuk ke sistem.	Berkas berhasil diunggah lewat kedua cara.	Sesuai
3	Sistem dapat memvalidasi berkas yang diunggah hanya berekstensi .php dan menolak berkas dengan ekstensi	Mengunggah satu berkas .php (harus diterima) dan satu berkas berekstensi lain seperti	Berkas .php diterima; berkas selain .php ditolak disertai pesan kesalahan.	Berkas .php diterima dan diproses; berkas .exe ditolak dengan pesan kesalahan.	Sesuai



© Hak Cipta milik 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

	lain disertai pesan kesalahan.	.exe (harus ditolak).			
4	Sistem dapat menjalankan pemeriksaan kode menggunakan <i>review engine</i> dan model AI terhadap berkas yang diunggah.	Menjalankan pemeriksaan atas satu berkas .php yang sudah diunggah, lalu menunggu sampai AI selesai memeriksa.	Sistem berhasil memeriksa kode dan menghasilkan temuan dari AI.	Sistem berhasil memeriksa kode dan menghasilkan temuan dari AI.	Sesuai
5	Sistem dapat menampilkan kemajuan pemeriksaan (<i>progress</i>) selama proses berlangsung.	Mengamati tampilan layar selagi sistem sedang memeriksa kode.	Muncul indikator kemajuan (<i>progress</i>) yang berjalan sampai pemeriksaan selesai.	Indikator kemajuan tampil dan berjalan sampai selesai.	Sesuai
6	Sistem dapat menampilkan ringkasan hasil pemeriksaan.	Melihat halaman hasil setelah satu pemeriksaan selesai dijalankan.	Ringkasan hasil pemeriksaan tampil, misalnya jumlah temuan yang ditemukan.	Ringkasan hasil tampil dengan benar.	Sesuai



© Hak Cipta milik 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

7	Sistem dapat menampilkan rincian temuan dan potongan kode terkait.	Membuka salah satu baris temuan untuk melihat potongan kode yang ditandai bermasalah.	Potongan kode terkait tampil, dengan baris yang bermasalah ditandai.	Potongan kode tampil dan baris yang bermasalah ditandai dengan benar.	Sesuai
8	Sistem dapat menampilkan riwayat pemeriksaan.	Membuka menu riwayat, lalu membuka kembali salah satu hasil pemeriksaan yang pernah dilakukan sebelumnya.	Daftar riwayat tampil, dan detail pemeriksaan lama bisa dibuka kembali dengan benar.	Daftar riwayat tampil dan detail lama berhasil dibuka kembali.	Sesuai
9	Sistem dapat menyediakan formulir penilaian pengguna dengan SUS pada antarmuka web.	Membuka formulir penilaian SUS di antarmuka web dan mengisi seluruh pernyataan yang ada.	Formulir SUS tampil dan bisa diisi sampai selesai.	Formulir SUS tampil dan berhasil diisi oleh responden.	Sesuai
10	Sistem dapat mengirim hasil penilaian pengguna ke Google Sheet	Mengirim (submit) formulir SUS yang sudah diisi, lalu	Data hasil penilaian tersimpan di Google Sheet.	Data 15 responden berhasil tersimpan di Google Sheet.	Sesuai



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

melalui backend.	memeriksa apakah datanya masuk ke Google Sheet.			
------------------	---	--	--	--

Tingkat keberhasilan pengujian fungsional dievaluasi dengan membandingkan jumlah skenario yang berhasil (Sesuai) dengan total keseluruhan skenario yang diuji. Persentase keberhasilan dihitung menggunakan persamaan berikut:

$$\text{Persentase Keberhasilan} = \frac{\text{Jumlah Skenario Sesuai}}{\text{Jumlah Seluruh Skenario}} \times 100$$

Berdasarkan hasil pada Tabel 5., dari total 10 skenario yang dirancang, seluruhnya (10 skenario) menghasilkan keluaran yang sesuai dengan harapan. Oleh karena itu, perhitungannya adalah sebagai berikut:

$$\text{Persentase Keberhasilan} = \frac{10}{10} \times 100$$

Secara keseluruhan, dapat disimpulkan bahwa perangkat lunak mampu beroperasi secara optimal dan aman. Sistem mampu menjalankan alur pemeriksaan dengan tepat, merekam data riwayat audit dengan benar, serta mempertahankan mekanisme kontrol akses tanpa mengganggu kinerja utama aplikasi.

4.4.2 Penilaian Pengguna dengan System Usability Scale (SUS)

Berdasarkan Tabel 9., penilaian SUS menggunakan skala Likert 1 sampai 5, mulai dari "Sangat tidak setuju" hingga "Sangat setuju". Skala ini digunakan responden untuk memberikan jawaban terhadap setiap pernyataan pada instrumen System Usability Scale. Setelah skala penilaian ditentukan, responden diminta menjawab 10 pernyataan SUS. Daftar pernyataan instrumen SUS yang digunakan dalam penelitian ini ditunjukkan pada Tabel 10..

Tabel 10. Skala Penilaian SUS

Nilai	Keterangan
-------	------------



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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

1	Sangat tidak setuju
2	Tidak setuju
3	Netral
4	Setuju
5	Sangat setuju

Tabel 11. Daftar Pernyataan Instrumen SUS

No	Pernyataan
1	Saya rasa saya akan sering menggunakan sistem ini.
2	Saya merasa sistem ini terlalu rumit, padahal dapat dibuat lebih sederhana.
3	Saya rasa sistem ini mudah digunakan.
4	Saya rasa saya membutuhkan bantuan orang teknis untuk dapat menggunakan sistem ini.
5	Saya menemukan berbagai fungsi pada sistem ini terintegrasi dengan baik.
6	Saya rasa terdapat terlalu banyak ketidaksesuaian pada sistem ini.
7	Saya rasa kebanyakan orang akan dapat mempelajari sistem ini dengan cepat.
8	Saya merasa sistem ini membingungkan untuk digunakan.
9	Saya merasa yakin dapat menggunakan sistem ini.
10	Saya perlu mempelajari banyak hal terlebih dahulu sebelum dapat menggunakan sistem ini.

Perhitungan skor SUS dilakukan dengan ketentuan sebagai berikut. Untuk setiap pernyataan bernomor ganjil, skor kontribusi diperoleh dari skor jawaban responden dikurangi 1. Untuk setiap pernyataan bernomor genap, skor kontribusi diperoleh dari 5 dikurangi skor jawaban responden. Seluruh skor kontribusi dari 10 butir pernyataan kemudian dijumlahkan dan dikalikan dengan 2,5 sehingga menghasilkan skor SUS dalam rentang 0 sampai 100:



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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

$$SUS = ((Q1 - 1) + (5 - Q2) + (Q3 - 1) + (5 - Q4) + (Q5 - 1) + (5 - Q6) + (Q7 - 1) + (5 - Q8) + (Q9 - 1) + (5 - Q10)) \times 2,5$$

Keterangan:

SUS = skor System Usability Scale

Q1, Q3, Q5, Q7, Q9 = skor jawaban pada butir pernyataan ganjil

Q2, Q4, Q6, Q8, Q10 = skor jawaban pada butir pernyataan genap

Kategori interpretasi skor SUS yang digunakan dalam penelitian ini ditunjukkan pada Tabel 12..

Tabel 12. Kategori Interpretasi Skor SUS

Rentang Skor SUS	Grade	Rentang Persentil
84,1 – 100	A+	96 – 100
80,8 – 84,0	A	90 – 95
78,9 – 80,7	A-	85 – 89
77,2 – 78,8	B+	80 – 84
74,1 – 77,1	B	70 – 79
72,6 – 74,0	B-	65 – 69
71,1 – 72,5	C+	60 – 64
65,0 – 71,0	C	41 – 59
62,7 – 64,9	C-	35 – 40
51,7 – 62,6	D	15 – 34
0,0 – 51,6	F	0 – 14

Sumber: (Suria, 2024)

Berdasarkan Tabel 8., skor SUS dikelompokkan ke dalam empat kategori interpretasi, yaitu Tidak layak, Cukup, Baik, dan Sangat baik. Rentang skor tersebut digunakan untuk menentukan tingkat kelayakan sistem dari sisi kegunaan, baik berdasarkan skor akhir setiap responden maupun skor rata-rata seluruh responden.

Penilaian SUS pada penelitian ini melibatkan 15 responden pengembang aplikasi WISE/ESCM dari PT Wijaya Karya (Persero) Tbk, dengan alamat surel berdomain @wikamail. Responden terdiri atas 4 Fullstack Developer, 4 QA/Tester, 3 Backend Developer, 2 responden dengan peran lainnya, 1 Lead Developer, dan 1 Frontend



© Hak Cipta milik 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 mengumpukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Developer, dengan pengalaman kerja bervariasi dari kurang dari 1 tahun hingga lebih dari 5 tahun. Kuesioner diisi melalui antarmuka web dalam dua gelombang, yaitu 26 Juni–2 Juli 2026 sore hari dan 2 Juli 2026 malam hari. Enam responden yang mengisi lebih dari satu kali dihitung berdasarkan pengisian terakhir.

Data jawaban asli responden berupa nilai Q1 sampai Q10 digunakan sebagai dasar perhitungan skor SUS. Contoh penerapan perhitungan skor SUS pada salah satu responden adalah sebagai berikut:

$$Q1 = 5, Q2 = 1, Q3 = 4, Q4 = 1, Q5 = 5, Q6 = 2, Q7 = 4, Q8 = 3, Q9 = 5, Q10 = 2$$

Berdasarkan data jawaban tersebut, skor kontribusi untuk setiap butir dihitung sebagai berikut:

$$SUS = ((5 - 1) + (5 - 1) + (4 - 1) + (5 - 1) + (5 - 1) + (5 - 2) + (4 - 1) + (5 - 3) + (5 - 1) + (5 - 2)) \times 2,5$$

$$SUS = (4 + 4 + 3 + 4 + 4 + 3 + 2 + 4 + 3) \times 2,5$$

$$SUS = 34 \times 2,5$$

$$SUS = 85,0$$

Hasil perhitungan tersebut menunjukkan bahwa responden tersebut memperoleh skor SUS sebesar 85,0. Berdasarkan kategori interpretasi pada Tabel 11., skor 85,0 berada pada rentang 84,1–100 sehingga termasuk dalam grade A+ (persentil 96–100).

Setelah skor SUS setiap responden diperoleh, skor rata-rata SUS dihitung dengan menjumlahkan seluruh skor SUS responden, kemudian dibagi dengan jumlah responden. Rumus rata-rata skor SUS adalah sebagai berikut:

$$\overline{SUS} = \frac{\sum SUS}{n}$$

Keterangan:

$$\overline{SUS} = \text{rata-rata skor SUS}$$



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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

$$\sum SUS = \text{jumlah seluruh skor SUS responden}$$

$$n = \text{jumlah responden}$$

Rekapitulasi skor SUS dari seluruh responden ditunjukkan pada Tabel 13.

Tabel 13. Rekapitulasi Skor SUS Responden

Responden	Peran	Skor SUS (0–100)	Grade
Responden 1	Lead Developer	85,0	A+
Responden 2	Fullstack Developer	37,5	F
Responden 3	Fullstack Developer	35,0	F
Responden 4	Backend Developer	70,0	C
Responden 5	QA / Tester	82,5	A
Responden 6	Backend Developer	75,0	B
Responden 7	Fullstack Developer	70,0	C
Responden 8	Lainnya	40,0	F
Responden 9	QA / Tester	55,0	D
Responden 10	QA / Tester	87,5	A+
Responden 11	Lainnya	67,5	C
Responden 12	Frontend Developer	85,0	A+
Responden 13	QA / Tester	82,5	A
Responden 14	Backend Developer	80,0	A-
Responden 15	Fullstack Developer	92,5	A+
Rata-rata		69,67	C

Berdasarkan Tabel 12., skor SUS masing-masing responden adalah 85,0; 37,5; 35,0; 70,0; 82,5; 75,0; 70,0; 40,0; 55,0; 87,5; 67,5; 85,0; 82,5; 80,0; dan 92,5. Dengan demikian, jumlah keseluruhan skor SUS responden adalah sebagai berikut:

$$\sum SUS = 85,0 + 37,5 + 35,0 + 70,0 + 82,5 + 75,0 + 70,0 + 40,0 + 55,0 + 87,5 + 67,5 + 85,0 + 82,5 + 80,0 + 92,5$$

$$\sum SUS = 1045,0$$

Selanjutnya, nilai rata-rata SUS dihitung dengan membagi jumlah seluruh skor SUS dengan jumlah responden, yaitu sebagai berikut:



© Hak Cipta milik 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

$$\overline{SUS} = \frac{1045,0}{15}$$

$$\overline{SUS} = 69,67$$

Hasil perhitungan menunjukkan bahwa skor rata-rata SUS sistem adalah 69,67. Berdasarkan kategori interpretasi pada Tabel 11., nilai tersebut berada pada rentang 65,0–71,0 sehingga termasuk dalam grade C (persentil 41–59), mengingat skor 68 merupakan titik tengah skala ini (Suria, 2024). Dengan demikian, tingkat kebergunaan sistem berada setara dengan rata-rata sistem pada umumnya, dengan ruang perbaikan yang cukup jelas untuk mencapai tingkat kebergunaan yang lebih unggul.

Sebaran hasil penilaian menunjukkan variasi yang cukup lebar: 4 dari 15 responden memperoleh grade A+, 2 responden grade A, 1 responden grade A-, 1 responden grade B, 3 responden grade C, 1 responden grade D, dan 3 responden grade F. Hasil ini menunjukkan bahwa meskipun sebagian responden menilai sistem sangat mudah digunakan, terdapat pula sejumlah responden yang mengalami kesulitan signifikan, sejalan dengan hambatan teknis yang disampaikan sebagian responden.

Analisis rata-rata jawaban per butir menunjukkan dua butir dengan kontribusi terendah, yaitu butir mengenai tidak adanya hambatan dalam menggunakan sistem dan butir mengenai kebutuhan bantuan orang lain atau teknisi. Temuan ini mengindikasikan bahwa sebagian responden masih merasakan hambatan teknis ketika menggunakan sistem. Sebaliknya, dua butir dengan kontribusi positif tertinggi adalah butir mengenai keinginan menggunakan sistem kembali dan pandangan bahwa sistem tidak rumit untuk digunakan. Hal ini menunjukkan adanya ketertarikan responden untuk menggunakan sistem kembali serta persepsi kemudahan penggunaan yang cukup kuat.

4.4.3 Pembahasan Hasil Evaluasi

Kedua aspek pengujian pada penelitian ini menunjukkan gambaran yang saling melengkapi. Dari sisi fungsional, seluruh skenario *black-box testing* berjalan sesuai yang diharapkan (100%), yang berarti fitur inti sistem, mulai dari validasi unggahan



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

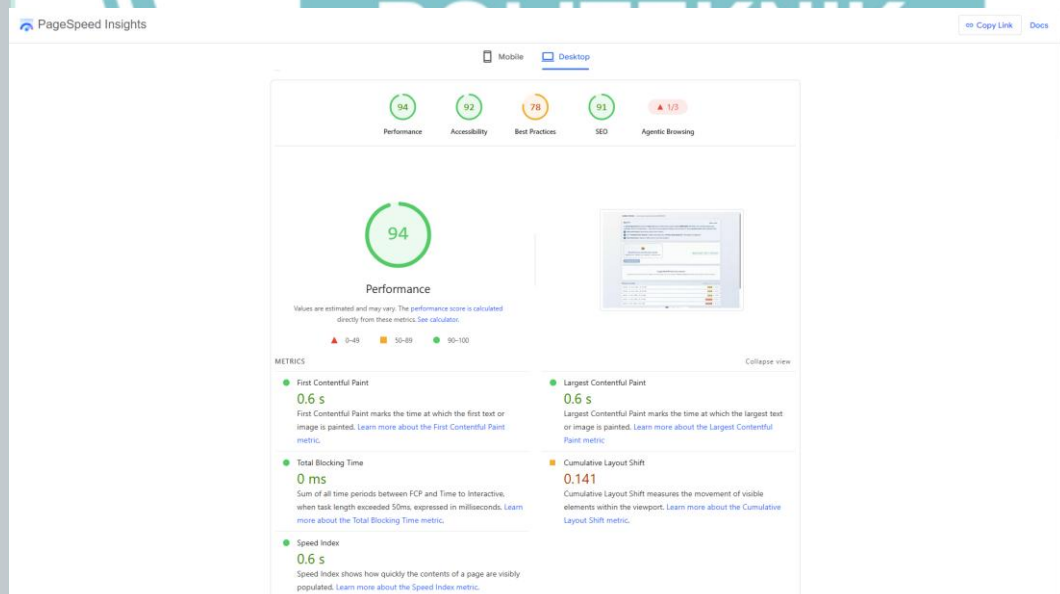
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan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

hingga kontrol akses, bekerja dengan benar. Dari sisi kebergunaan, skor SUS sebesar 69,67 (kategori Baik) menunjukkan bahwa sistem secara umum sudah dapat diterima pengguna, dengan hambatan teknis yang sesekali dirasakan dan kebutuhan bantuan tambahan bagi sebagian pengguna sebagai aspek yang masih dapat disempurnakan, sejalan dengan komentar responden yang menyarankan keterangan kesalahan yang lebih detail. Kedua aspek ini bersama-sama menggambarkan bahwa sistem sudah berfungsi dengan benar dan cukup mudah digunakan.

4.4.4 Pengujian Non-Fungsional

Mengingat kebutuhan non-fungsional yang dibutuhkan pada sistem ini adalah terkait *performance* sebagaimana yang disebutkan pada sub bab kebutuhan non-fungsional, jadi penulis melakukan pengujian *performance* menggunakan situs pagespeed.web.dev.

Cara melakukan pengujian pada situs tersebut cukup mudah, penulis cukup memasukkan URL sistem kedalam *input field* yang tersedia, saat *user* selesai input dan men-klik tombol “*analyze*” maka menunggu beberapa saat hingga hasil keluar. Dan hasil pengujian dinyatakan cukup baik sebagaimana pada gambar dibawah ini



Gambar 4. 23 Tampilan Hasil *performance* test pada laman pagespeed.web.dev



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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Pada gambar 4.23 bisa dilihat bahwa sistem dinilai cukup baik, sebagaimana pada metrik yang ada pada gambar, penjelasan metrik adalah sebagai berikut:

1. *Performance* (94): metrik ini mengukur seberapa responsif web nya saat digunakan oleh pengunjung. Skor 94 (hijau) menunjukkan performa yang sangat baik.
2. *Accessibility* (92): metrik ini mengukur seberapa mudah situs web diakses dan digunakan oleh semua orang, termasuk pengguna dengan disabilitas yang mungkin mengandalkan alat bantu seperti *screen reader*.
3. *Best Practices* (78): metrik yang Mengevaluasi apakah halaman web dibangun dengan mengikuti standar keamanan dan pedoman pengembangan web modern.
4. SEO (91): *Search Engine Optimization*. Metrik yang Menilai seberapa baik halaman web telah dioptimalkan agar mudah ditemukan oleh mesin pencari seperti Google.
5. *First Contentful Paint* (FCP) 0,6 s: Metrik ini mengukur waktu dari saat halaman mulai dimuat hingga konten pertama (teks, gambar, atau elemen grafis lainnya) berhasil dirender dan muncul di layar. Waktu 0.6 detik (hijau) menunjukkan respons yang sangat cepat, sehingga pengguna tidak perlu menunggu lama untuk melihat bahwa halaman sedang bekerja.
6. *Largest Contentful Paint* (LCP) 0,6 s: Berbeda dengan FCP yang hanya mengukur "kemunculan pertama", LCP mengukur waktu yang dibutuhkan hingga elemen konten paling besar (seperti gambar hero utama atau blok teks paragraf yang besar) selesai dimuat dan terlihat sepenuhnya di layar. Waktu 0.6 detik (hijau) menandakan performa yang sangat baik dan memenuhi standar kecepatan Google.
7. *Total Blocking Time* (TBT) - 0 ms: Metrik ini menghitung total waktu di mana tugas-tugas dari halaman web memblokir browser sehingga tidak bisa merespons interaksi



© Hak Cipta milik 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

pengguna (seperti klik layar atau scroll). Pemblokiran ini biasanya disebabkan oleh eksekusi skrip (JavaScript) yang berat. Hasil 0 ms (hijau) sangat sempurna; ini berarti halaman web sama sekali tidak mengalami *freeze* atau jeda saat dimuat, dan langsung siap merespons interaksi pengguna.

8. *Speed Index* - 0.6 s: Ini adalah pengukuran tentang seberapa cepat bagian visual dari halaman web terisi dan terlihat secara keseluruhan oleh pengguna. Waktu 0.6 detik (hijau) menunjukkan bahwa transisi dari layar putih/kosong ke halaman yang terisi penuh terjadi dalam sekejap.
9. *Cumulative Layout Shift* (CLS) - 0.141: metrik ini mengukur seberapa stabil visual halaman web. Metrik ini mendeteksi apakah ada elemen yang tiba-tiba bergeser tempatnya saat halaman sedang dimuat (misalnya, gambar yang tiba-tiba muncul dan mendorong teks ke bawah). Angka 0.141 (ditandai dengan kotak oranye) menunjukkan bahwa ada sedikit pergeseran yang terjadi. Idealnya, skor CLS yang baik adalah di bawah 0.1, sehingga ini adalah satu satuan metrik yang masih butuh peningkatan sedikit lagi.

Berdasarkan metrik metrik diatas, cukup sah dikatakan bahwa sistem memiliki waktu *load & performance* yang baik.

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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian dan pembahasan, dapat ditarik kesimpulan sebagai berikut.

Kebutuhan sistem pendukung *code review* berbasis Artificial Intelligence pada aplikasi WISE/ESCM berhasil diidentifikasi melalui pengamatan, wawancara dengan Lead Developer, dan penelaahan kode. Kebutuhan utama adalah membantu pemeriksaan kode berdasarkan aturan internal dan menampilkan hasil pemeriksaan secara mudah dipahami.

Sistem pendukung *code review* berhasil dirancang dan dibangun menggunakan metode Waterfall. Sistem terdiri atas *review engine* berbasis Node.js dengan pemanggilan OpenAI API dan penyaring *Hallucination Guard*, serta antarmuka web berbasis React.js, Vite, dan Express.js. Aturan pemeriksaan disusun menjadi *rulebook* berisi 30 aturan inti, yaitu aturan teknis umum (CTDL) dan aturan khas organisasi (WIKI).

Pengujian fungsional dengan *black-box testing* menunjukkan bahwa seluruh 10 skenario pengujian, mencakup validasi unggahan berkas, alur pemindaian, kontrol akses, pembatasan laju, hingga modul kuesioner SUS, berjalan sesuai yang diharapkan (100%).

Penilaian pengguna dengan System Usability Scale terhadap 15 responden pengembang WISE/ESCM memperoleh skor rata-rata 69,67 yang termasuk kategori Baik (acceptable). Analisis per butir menunjukkan hambatan teknis yang sesekali dirasakan dan kebutuhan bantuan tambahan sebagai aspek yang masih dapat disempurnakan, sementara ketertarikan menggunakan kembali sistem dan persepsi kemudahan penggunaan tergolong kuat. Temuan ini menunjukkan bahwa sistem tidak hanya berjalan dengan benar secara fungsional, tetapi juga cukup mudah diterima oleh penggunanya.



5.2 Saran

Berdasarkan keterbatasan penelitian, beberapa saran yang dapat diberikan adalah sebagai berikut.

1. Memperkaya contoh *few-shot* pada prompt untuk aturan khas organisasi (WIKA), mengingat sifatnya yang lebih tacit dan spesifik organisasi dibandingkan aturan CTDL yang bersifat teknis umum.
2. Melibatkan lebih banyak responden pada penilaian pengguna agar hasil SUS lebih mewakili beragam pengguna.
3. Menambahkan pencatatan hasil pemeriksaan secara berkelanjutan agar perkembangan mutu kode dapat dipantau dari waktu ke waktu.
4. Menambahkan dukungan pemeriksaan untuk berkas dengan bahasa pemrograman lain di luar PHP, mengingat aplikasi perusahaan tidak selalu dibangun dengan satu bahasa pemrograman saja.
5. Mengembangkan sistem dari purwarupa menjadi sistem yang siap digunakan pada lingkungan pengembangan nyata dengan tetap memperhatikan keamanan kode internal.
6. Menampilkan pesan peringatan eksplisit ketika jumlah berkas yang diunggah melebihi batas maksimal, alih-alih memangkas berkas kelebihan secara diam-diam, berdasarkan temuan pengujian *black-box* (BB-07).
7. Menyediakan panduan singkat, dokumentasi bantuan, atau tur pengenalan (*onboarding*) yang mudah diakses untuk mengurangi hambatan teknis yang sesekali masih dirasakan sebagian pengguna, sejalan dengan temuan penilaian SUS dan komentar responden yang mengharapkan keterangan kesalahan (*feedback*) hasil pemeriksaan dibuat lebih detail.

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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR PUSTAKA

- Albuquerque, D. dkk. (2022) 'Managing technical debt through intelligent techniques: a systematic *mapping* study', Journal of Systems and Software.
- Davila, N. dan Nunes, I. (2021) 'A systematic literature *review* and taxonomy of modern *code review*', Journal of Systems and Software, 177, hlm. 110951.
- Fenoglio, E. dkk. (2022) 'Tacit knowledge elicitation for systems engineering using artificial intelligence', Systems Engineering.
- Firmansyah, M. R., dkk. (2024). Analisis Usability *Testing* Menggunakan Metode System Usability Scale Pada Aplikasi Open Data Kabupaten Cirebon. JATI (Jurnal Mahasiswa Teknik Informatika), 7(6), 3238–3243.
- Gluhak, M. dan Pavlič, L. (2022) 'The role of a quality gate in a software delivery pipeline', dalam Prosiding Konferensi Internasional Rekayasa Perangkat Lunak.
- Hicks, S.A. dkk. (2022) 'On evaluation metrics for medical applications of artificial intelligence', Scientific Reports, 12, hlm. 5979.
- Hou, X. dkk. (2024) '*Large language models* for software engineering: a systematic literature *review*', ACM Transactions on Software Engineering and Methodology.
- Huang, L. dkk. (2024) 'A survey on hallucination in *large language models*: principles, taxonomy, challenges, and open questions', ACM Transactions on Information Systems.
- Kapoh, H. dkk. (2021) 'Pengujian *black-box* pada aplikasi web menggunakan metode equivalence partitioning', Jurnal Teknologi Informasi dan Komputer.
- Maulida, A., dkk. (2025). Pengujian Kualitas Fungsional Website PERKASA Menggunakan Metode Black Box dengan Pendekatan Equivalence Partitioning. Seminar Nasional Inovasi Teknologi (TIN).
- OpenAI (2024) OpenAI API Reference. Tersedia pada: <https://platform.openai.com/docs> (Diakses: 2024).



© Hak Cipta milik 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Perrig, S.A.C. dkk. (2024) 'Measurement practices in *user* experience research: a systematic *review*', International Journal of Human–Computer Studies.

Pornprasit, C. dan Tantithamthavorn, C. (2024) 'Fine-tuning and prompting *large language models* for automated *code review*', Empirical Software Engineering.

React (2024) React Documentation. Tersedia pada: <https://react.dev> (Diakses: 2024).

Sahoo, P. dkk. (2024) 'A systematic survey of prompt engineering in *large language models*: techniques and applications', arXiv preprint arXiv:2402.07927.

Saravanos, A. dan Curinga, M.X. (2023) 'Simulating the software development lifecycle: the Waterfall model', Applied System Innovation, 6(6).

Sharfina, Z. dan Santoso, H.B. (2016) 'An Indonesian adaptation of the System Usability Scale (SUS)', dalam 2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS). IEEE.

**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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR RIWAYAT HIDUP PENULIS



Rafif Nuraydin Adipratama

Lahir di Purwokerto, 14 April 2004.

Menempuh pendidikan dasar di SD Islam Pondok Duta Depok lulus pada tahun 2016, pendidikan menengah pertama di Pondok Pesantren Babussalam, lulus pada tahun 2019, pendidikan menengah keatas di Pondok Pesantren Babussalam lulus pada tahun 2022, Pendidikan profesi CEP-CCIT-FTUI lulus pada tahun 2024. Saat ini terdaftar sebagai mahasiswa aktif Program Studi Teknik Informatika, Jurusan Teknik Informatika dan Komputer, Politeknik Negeri Jakarta sejak tahun 2022.

**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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

LAMPIRAN

Lampiran 1 – Dokumentasi Wawancara





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
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Lampiran 2 – Hasil Analisis Temuan Analisis Kode Statis pada Repositori WISE/ESCM

Tabel Lampiran 2a. Tabel Aturan Domain Citadel *Framework* (CTDL-01 s.d. CTDL-15)

ID	Judul	Risk	Deskripsi	Evidence (Lokasi Kode)	Referensi Eksternal
CTDL-01	Wajib Eloquent ORM, dilarang raw query	High	Semua akses database HARUS melalui Eloquent ORM. DB:: <i>raw</i> (), DB:: <i>select</i> () untuk raw SQL dilarang kecuali untuk agregasi kompleks yang tidak bisa dilakukan Eloquent.	Semua 303+ model menggunakan Eloquent. <i>Controller</i> menggunakan Model:: <i>query</i> (), Model:: <i>where</i> (), Model:: <i>create</i> ().	OWASP ASVS V5, CWE-89 (SQL Injection)
CTDL-02	Mass assignment protection via \$guarded	High	Semua model HARUS menggunakan \$guarded = ["id"]. Tidak menggunakan \$fillable. Pattern ini konsisten di 100% model.	app/Models/Contract.php, User.php, Vendor.php, Division.php, Currency.php semua: protected \$guarded = ["id"];	Laravel Docs (Mass Assignment), CWE-915
CTDL-03	Blade escaping default, dilarang {!! !!} tanpa sanitasi	High	Semua output Blade HARUS menggunakan {{ }} (auto-escaped). Penggunaan {!! !!} hanya diperbolehkan untuk HTML yang sudah disanitasi server-side (misal: Citadel renderSchema() output).	Citadel Layout.php:renderSchema() menggunakan \$html .= \$s->backbone()->render() output HTML yang sudah dikontrol <i>framework</i> .	Laravel Docs (Blade), CWE-79 (XSS)
CTDL-04	Middleware auth pada protected routes	High	Semua route admin HARUS memiliki middleware cek_login + internal_placement. Route API HARUS memiliki basic_auth atau jwt.verify. Tidak boleh ada route protected tanpa middleware.	routes/web.php: Route::prefix('admin')->middleware(['cek_login', 'internal_placement']) konsisten di semua route group admin.	OWASP ASVS V4 (Access Control), OWASP AUTHZ-01



© Hak Cipta milik 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
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

ID	Judul	Risk	Deskripsi	Evidence (Lokasi Kode)	Referensi Eksternal
CTDL-05	Inline validation wajib di <i>controller</i>	Medium	Setiap <i>method</i> <i>store()</i> dan <i>update()</i> HARUS memiliki <i>\$request->validate([...])</i> dengan rules yang sesuai. Project ini TIDAK menggunakan Form Request <i>class</i> validasi dilakukan <i>inline</i> .	ProjectController::store(), PurchaseOrderController::store(), UserDashboardController semua menggunakan <i>\$request->validate()</i> <i>inline</i> .	Laravel Docs (Validation)
CTDL-06	Citadel Page lifecycle pattern wajib	Medium	<i>Controller method</i> yang merender halaman HARUS mengikuti lifecycle: <i>ComponentsPage::make()->view()->business()->schema()->render()</i> . Business logic HARUS dalam closure <i>business()</i> , bukan langsung di <i>method body</i> .	Pattern konsisten di 100+ <i>controller</i> : TenderController, PurchaseOrderController, ProjectController, VendorController.	Citadel Framework Convention
CTDL-07	HasCreator <i>trait</i> wajib pada model dengan <i>audit trail</i>	High	Semua model entitas bisnis HARUS menggunakan <i>trait</i> HasCreator untuk otomatis tracking <i>created_by/updated_by</i> . ~85% model sudah menggunakan pattern ini.	app/Models/Traits/HasCreator.php auto-set <i>created_by/updated_by</i> via <i>boot method</i> . Digunakan di: Contract, Vendor, Division, Currency, dll.	Audit Trail Best Practice, SOX Compliance
CTDL-08	ActivityLogged <i>trait</i> wajib pada model bisnis	Medium	Semua model entitas bisnis HARUS menggunakan <i>trait</i> ActivityLogged (Spatie) untuk pencatatan perubahan. ~95% model sudah menggunakan.	app/Models/Traits/ActivityLogged.php wraps Spatie\Activitylog. Digunakan di hampir semua model.	Audit Trail Best Practice
CTDL-09	Enum status dengan <i>label()</i> dan <i>color()</i>	Medium	Semua field status HARUS menggunakan PHP 8.1+ Enum dengan <i>method</i> <i>label()</i> : string dan <i>color()</i> : string. Enum harus implement CitadelEnum interface.	app/Enums/ApprovalRequestStatus.php, ProcurementStatus.php, VendorStatus.php semua memiliki <i>label() + color() + getStatus()</i> . 39+ enum files.	PHP 8.1 Enums, Citadel CitadelEnum Contract

POLITEKNIK
NEGERI
JAKARTA



© Hak Cipta milik 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
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

ID	Judul	Risk	Deskripsi	Evidence (Lokasi Kode)	Referensi Eksternal
CTDL-10	Custom table name dengan prefix escm_	Low	Semua model HARUS mendefinisikan \$table property secara eksplisit dengan prefix escm_. Tidak boleh mengandalkan Laravel auto table naming.	Contract: protected \$table = "escm_contracts", User: "escm_users", Division: "escm_division" konsisten di semua model.	Database Naming Convention
CTDL-11	Permission check via checkAccessPermission()	High	Setiap <i>controller method</i> yang melakukan operasi bisnis HARUS memanggil checkAccessPermission(\$permission) di awal <i>method</i> . Permission menggunakan format feature:action (contoh: tender_elektronik:view).	ProjectController::index(), ContractsController::index(), VendorPerformanceIndexController selalu ada checkAccessPermission() di awal <i>method</i> .	RBAC Best Practice, OWASP ASVS V4
CTDL-12	Aksi non-CRUD sebagai turunan Update	Medium	Semua operasi bisnis selain CRUD standar (propose, release, sync, approve, esign) HARUS di-treatment sebagai <i>method</i> tersendiri dalam <i>controller</i> , bukan sebagai generic update. Pattern: <i>method name</i> = nama aksi.	PurchaseOrderController: propose(), release(), release_amendment(), sync_po(), esign(), export_sap(). TenderController: open_tender(), revision_tender(), announcement_tender().	Citadel Framework Convention
CTDL-13	Schema component harus implement Backbone	High	Setiap komponen yang dimasukkan ke dalam schema[...] pada Citadel Layout HARUS implement interface Backbone. Jika tidak, akan throw error BACKBONE[001].	Layout.php:renderSchema(): if(!\$instanceof Backbone) { throw new Error('BACKBONE[001]...') }	Citadel Framework Contract
CTDL-14	Makeable pattern untuk factory instantiation	Low	Semua Citadel component HARUS di-instantiate via static <i>method</i> ::make(\$name, \$title). Tidak boleh menggunakan new ComponentClass() secara langsung.	Makeable.php <i>trait</i> : public static function make(\$name, \$title = null) name di-snake_case otomatis. Digunakan di semua component.	Factory Pattern, Citadel Convention

POLITEKNIK
NEGERI
JAKARTA



© Hak Cipta milik 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
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

ID	Judul	Risk	Deskripsi	Evidence (Lokasi Kode)	Referensi Eksternal
CTDL-15	HasNumbering <i>trait</i> untuk auto-numbering	Medium	Model procurement (Tender, PO, PR, Contract) HARUS menggunakan <i>trait</i> HasNumbering dan implement interface UseNumbering untuk auto-generate nomor dokumen.	HasNumbering.php: auto-generate numbering via bootHasNumbering(). Contract, Procurement, PurchaseOrder menggunakan <i>trait</i> ini.	Document Numbering Convention

Tabel 2b. Tabel Aturan Domain WIKA/ESCM (WIKI-Q01 s.d. WIKI-Q15)

ID	Judul	Risk	Deskripsi	Evidence (Lokasi Kode)	Referensi Eksternal
WIKI-Q01	Tidak ada Observer <i>class</i> semua event di boot <i>method/trait</i>	Low	Project TIDAK menggunakan Observer <i>class</i> . Semua model event handling dilakukan melalui boot() <i>method</i> atau <i>trait</i> bootTraitName(). Konsisten di 100% model.	Tidak ditemukan file di app/Observers/. HasCreator::bootHasCreator(), HasNumbering::bootHasNumbering(), User::boot() semua logic di boot.	Laravel Docs (Observers vs Boot)
WIKI-Q02	Tidak ada Form Request <i>class</i> validasi <i>inline</i>	Medium	Project TIDAK menggunakan Form Request <i>class</i> (app/Http/Requests/). Semua validasi dilakukan <i>inline</i> \$request->validate() di <i>controller</i> .	Folder app/Http/Requests/ tidak ada atau kosong. Semua <i>controller</i> menggunakan <i>inline</i> validation.	Laravel Docs (Form Requests)
WIKI-Q03	\$guarded bukan \$fillable	Medium	Standar organisasi: gunakan \$guarded = ["id"] bukan \$fillable. Ini berarti semua field selain id bisa di-mass-assign. Risiko: jika ada field sensitif baru, harus ingat menambahkan ke \$guarded.	100% model: protected \$guarded = ["id"]. Tidak ada model dengan \$fillable.	Laravel Docs (Mass Assignment), CWE-915



© Hak Cipta milik 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
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

ID	Judul	Risk	Deskripsi	Evidence (Lokasi Kode)	Referensi Eksternal
WIKA-Q04	Response pattern: apiRes() helper untuk JSON	Low	Semua response JSON HARUS menggunakan helper apiRes('success' 'error', ...). Untuk response sukses dari form, gunakan Citadel Swal::make('success').	ProjectController::update(): return apiRes('success', detail: ['swal' => Swal::make('success')...]) pattern konsisten.	API Response Convention
WIKA-Q05	Sync mechanism mandatory untuk data eksternal	High	Semua data yang bersumber dari sistem eksternal (SAP, HCMS, VMS, Pengadaan.com) HARUS memiliki sync service dedicated. Pattern: app/Sync/Service/ + app/Jobs/Sync*.php.	78+ sync files. SAPSync.php, VendorSync.php, WikaProjectSync.php, HCMSUserSync. Abstract base: App\Sync\Service\Base.	Data Integration Pattern
WIKA-Q06	Sync service harus extend App\Sync\Service\Base	Medium	Semua sync service HARUS extend abstract class Base dan implement: fetching(), handleResponseForCache(), saving(). Optional: beforeSaving(), afterSaving().	app/Sync/Service/Base.php: abstract methods fetching(), handleResponseForCache(\$body, \$response), saving(\$data, \$i).	Template Method Pattern
WIKA-Q07	Scope-based data access untuk multi-tenant project	High	Akses data pada modul procurement HARUS menggunakan scope project dari ESCM::scopeProject(auth()->user()). Ini memastikan user hanya melihat data dari proyek yang mereka punya akses.	ProjectController: \$query->where(ESCM::scopeProject(auth()->user())). ProcurementModuleMethod trait: scopeDocumentAccess().	Multi-tenant Security, OWASP AUTHZ
WIKA-Q08	Password/PIN masking di semua log	High	Semua middleware logging HARUS melakukan masking terhadap field sensitif: password,	AccessLog.php: if (isset(\$input['password'])) \$input['password'] = "xxx"; di handle() dan terminate().	OWASP LOG-02, CWE-532



© Hak Cipta milik 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
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

ID	Judul	Risk	Deskripsi	Evidence (Lokasi Kode)	Referensi Eksternal
			<i>password_confirm</i> , <i>pass</i> , <i>pin</i> sebelum menyimpan ke log.		
WIKI-Q09	SoftDeletes hanya untuk entitas tertentu	Low	SoftDeletes HANYA digunakan pada model dengan kebutuhan recovery (<i>User</i> , <i>Vendor</i> , <i>VendorQuotation</i>). Mayoritas model menggunakan status flag, bukan soft delete.	Hanya ~5% model: <i>User.php</i> , <i>Vendor.php</i> , <i>VendorQuotation.php</i> menggunakan SoftDeletes. Model lain menggunakan enum status.	Database Design Convention
WIKI-Q10	Relationship key menggunakan <i>business key</i> bukan ID	Medium	Beberapa relationship menggunakan <i>business key</i> (nomor dokumen, kode) sebagai foreign key, bukan surrogate key (id). Ini untuk konsistensi dengan SAP numbering.	<i>Contract::contract_items()</i> : <i>hasMany(ContractItem::class, 'contract_number', 'number')</i> . <i>Contract::vendor()</i> : <i>belongsTo(Vendor::class, 'vendor_registration_number', 'registration_number')</i> .	SAP Integration Convention
WIKI-Q11	<i>Approval workflow</i> via <i>morphMany</i> pattern	High	Model yang membutuhkan <i>approval</i> HARUS implement interface <i>ApprovableContract</i> dan menggunakan <i>trait ReleaseStrategyModel</i> . <i>Approval</i> menggunakan polymorphic relationship <i>morphMany(ApprovalHeader::class, 'model')</i> .	<i>Vendor</i> implements <i>ApprovableContract</i> , <i>Procurement</i> implements <i>ProcurementModule</i> . <i>Trait ReleaseStrategyModel</i> memberikan <i>approval workflow</i> .	<i>Workflow Pattern</i>
WIKI-Q12	<i>Procurement</i> module interface wajib	Medium	Semua entitas <i>procurement</i> (<i>Tender</i> , <i>PO</i> , <i>PR</i> , <i>Contract</i>) HARUS implement interface <i>ProcurementModule</i> dengan <i>method</i>	<i>app/Models/Contracts/ProcurementModule.php</i> . Diimplementasikan oleh <i>Contract</i> , <i>Procurement</i> , dan model <i>procurement</i> lainnya.	Interface Segregation Principle



© Hak Cipta milik 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
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

ID	Judul	Risk	Deskripsi	Evidence (Lokasi Kode)	Referensi Eksternal
			getProcurementNumber() dan historyModelName().		
WIKA-Q13	Explicit route definition, bukan Route::resource()	Low	Project TIDAK menggunakan Route::resource(). Semua route didefinisikan secara eksplisit per <i>method</i> . Ini memberikan kontrol granular atas naming dan middleware per <i>endpoint</i> .	routes/web.php (2300+ baris): semua route eksplisit. Route::get(), Route::post() per <i>method</i> .	Routing Convention
WIKA-Q14	Multi-layer authentication	High	Sistem menggunakan 3 layer autentikasi: (1) Session via cek_login untuk web, (2) Basic Auth + IP <i>whitelist</i> via basic_auth untuk API, (3) JWT via jwt.verify untuk mobile/external.	Kernel.php aliases: cek_login, basic_auth, jwt.verify. AuthBasic.php: IP <i>whitelist</i> check. JwtMiddleware.php: JWT validation.	<i>Defense in Depth</i> , OWASP AUTH
WIKA-Q15	CSRF protection tanpa exception	High	CSRF protection AKTIF untuk semua POST/PUT/DELETE request. Tidak ada route yang di-exclude dari CSRF verification.	VerifyCsrfToken.php: protected \$except = [] array kosong, tidak ada exception.	OWASP SESS-06, CWE-352

POLITEKNIK
NEGERI
JAKARTA



© Hak Cipta milik 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
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Lampiran 3 – Rincian Gap Analysis 48 Klaim Wawancara vs Implementasi Kode

No	Kategori	Ref	Klaim Spesifik (dari Wawancara)	Status	Bukti / Aturan Terkait	Catatan Gap / Insight
1	Arsitektur & Komponen Utama	P1	PHP Laravel sebagai bahasa & <i>framework</i> utama	Selaras Penuh	CITADEL_TECHNICAL_FORMALIZATION.md §1.2: PHP 8.1+, Laravel 10.x	
2	Arsitektur & Komponen Utama	P1	JavaScript & <i>plugin</i> pendukung (jQuery, Alpine.js, dll.)	Selaras Penuh	§1.2 & §6.2: jQuery 3.7, Alpine.js 3.13, Bootstrap 5.3, dll.	
3	Arsitektur & Komponen Utama	P1	Penggunaan <i>custom framework</i> Citadel	Selaras Penuh	§2: Arsitektur <i>Framework</i> Citadel (2 generasi, 54+ file)	

POLITEKNIK
NEGERI
JAKARTA



© Hak Cipta milik 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
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

No	Kategori	Ref	Klaim Spesifik (dari Wawancara)	Status	Bukti / Aturan Terkait	Catatan Gap / Insight
4	Arsitektur & Komponen Utama	P1	React.js digunakan untuk beberapa modul spesifik	Selisih/Parsial	Belum terdokumentasi di formalisasi teknis	G-01: Modul React belum teridentifikasi; belum ada konvensi <i>review</i> khusus komponen React.
5	Kerangka Kerja Citadel	P2	Citadel = abstraksi & penyederhanaan requirement bisnis ke teknis	Selaras Penuh	CTDL-06: Citadel Page lifecycle pattern wajib	
6	Kerangka Kerja Citadel	P2	Dev fokus definisi bisnis di <i>controller</i> (backend-oriented), frontend di-handle Citadel	Selaras Penuh	§2.3: <i>Controller</i> mengembalikan Citadel page, bukan manual blade	
7	Kerangka Kerja Citadel	P2	Definisi tampilan di- <i>translate</i> menjadi HTML	Selaras Penuh	§2.2: Layout.php → renderSchema() → HTML output	



© Hak Cipta milik 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
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

No	Kategori	Ref	Klaim Spesifik (dari Wawancara)	Status	Bukti / Aturan Terkait	Catatan Gap / Insight
8	Kerangka Kerja Citadel	P2	Satu URL menangani GET (render halaman) dan POST (form submission)	Selaras Penuh	§2.4: Route::citadel() macro Route::match(['get','post'], ...)	
9	Kerangka Kerja Citadel	P2	Lifecycle: make → definisi tampilan → render	Selaras Penuh	§2.3: make() → view() → business() → schema() → render()	
10	Kerangka Kerja Citadel	P2	Citadel jadi 'kiblat'/acuan agar hasil kerja developer seragam	Selaras Penuh	CTDL-06 (<i>Medium</i> risk): konsistensi pattern di 100+ <i>controller</i>	
11	Kerangka Kerja Citadel	P2	Citadel terinspirasi dari Laravel Filament (backend-oriented)	Selisih/Parsial	Belum terdokumentasi eksplisit, meski prinsip serupa	G-02: Rationale arsitektural (inspirasi desain) belum dicantumkan, padahal memperkuat



© Hak Cipta milik 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
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

No	Kategori	Ref	Klaim Spesifik (dari Wawancara)	Status	Bukti / Aturan Terkait	Catatan Gap / Insight
						justifikasi CTDL-06.
12	Konvensi Basis Data & Model	P3	Penamaan mengikuti standar Laravel & PSR-4	Selisih/Parsial	Implisit di seluruh codebase, belum eksplisit di <i>rulebook</i>	G-03: Berpotensi jadi aturan WIKA-I01.
13	Konvensi Basis Data & Model	P3	Model ditulis <i>singular</i> , variabel array ditulis <i>plural</i>	Selisih/Parsial	Konvensi Laravel standar, belum eksplisit di <i>rulebook</i>	G-03: Berpotensi jadi aturan WIKA-I01.
14	Konvensi Basis Data & Model	P3	Desain mengutamakan normalisasi data dengan relasi antar-model yang jelas	Selaras Penuh	WIK-AQ10: Relationship key pakai <i>business key</i> ; §1.1: 303+ model dengan relasi	
15	Konvensi Basis Data & Model	P3	Setiap data yang dibuat <i>user</i> wajib punya timestamp (created_at/updated_at)	Selaras Penuh	Laravel default \$timestamps = true	

POLITEKNIK
NEGERI
JAKARTA



© Hak Cipta milik 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
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

No	Kategori	Ref	Klaim Spesifik (dari Wawancara)	Status	Bukti / Aturan Terkait	Catatan Gap / Insight
16	Konvensi Basis Data & Model	P3	Setiap data yang melibatkan <i>user</i> wajib punya pencatatan <i>creator</i>	Selaras Penuh	CTDL-07 (<i>High</i>): <i>HasCreator trait</i> wajib	
17	Konvensi Basis Data & Model	P3	<i>Primary key</i> tetap <i>auto increment</i> , bukan <i>business key</i> (mis. NIP bukan PK)	Selaras Penuh	WIKI-Q10: <i>business key</i> sbg FK bukan PK; CTDL-02: <i>\$guarded=["id"]</i>	
18	Konvensi Basis Data & Model	P3	<i>Trait</i> eksplisit (<i>HasColour</i> , dll.) dipakai untuk men-eksplisitkan model ke Citadel	Selaras Penuh	§2.2: <i>HasColor</i> , <i>HasIcon</i> , <i>HasURL</i> , <i>HasView</i> , dll. + CTDL-09: <i>CitadelEnum</i>	
19	Konvensi Basis Data & Model	P3	Kelemahan awal: tim kurang terbiasa membuat <i>report</i> karena desain ternormalisasi	Wawasan Baru	Tacit knowledge baru konteks operasional (kebutuhan data denormalisasi utk reporting)	
20	Konvensi Basis Data & Model	P3	Gaya penulisan Citadel terinspirasi Laravel Filament (relasi model jelas & tepat)	Selisih/Parsial	Lihat Gap G-02	G-02 (silang-rujuk dengan P2).



© Hak Cipta milik 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
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

No	Kategori	Ref	Klaim Spesifik (dari Wawancara)	Status	Bukti / Aturan Terkait	Catatan Gap / Insight
21	Konvensi Validasi	P4	Validasi <i>inline</i> dipilih demi menjaga kode tetap 'clean'/tidak boros baris	Selaras Penuh	CTDL-05: <i>Inline</i> validation wajib di <i>controller</i>	
22	Konvensi Validasi	P4	Tidak memakai Form Request <i>class</i> standar Laravel	Selaras Penuh	WIKI-Q02: Tidak ada Form Request <i>class</i> validasi <i>inline</i>	
23	Konvensi Validasi	P4	Alasan spesifik: efisiensi jumlah baris kode	Wawasan Baru	Rasionalisasi baru menambah konteks 'mengapa' pada aturan existing	Melengkapi CTDL-05/WIKI-Q02 dgn alasan desain, bukan kelalaian.
24	Keamanan & Otorisasi	P5	<code>checkAccessPermission()</code> wajib dipanggil di awal <i>method controller</i>	Selaras Penuh	CTDL-11 (<i>High</i>): Permission check di awal setiap <i>method</i>	
25	Keamanan & Otorisasi	P5	Akses data ditentukan hierarki level organisasi (Divisi → Departemen)	Selisih/Parsial	§5.2: Level-based permissions (superadmin/admin/audit/developer)	G-04 (parsial).



© Hak Cipta milik 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
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

No	Kategori	Ref	Klaim Spesifik (dari Wawancara)	Status	Bukti / Aturan Terkait	Catatan Gap / Insight
26	Keamanan & Otorisasi	P5	Kombinasi level organisasi × level kemampuan (<i>User/Admin/SuperAdmin</i>) menentukan privilege	Selisih/Parsial	WIKI-Q07: ESCM::scopeProject() untuk scoping	G-04 (parsial).
27	Keamanan & Otorisasi	P5	Matriks privilege 4 kombinasi cakupan × role (<i>Dept-User</i> , <i>Dept-Admin</i> , <i>Div-User</i> , <i>Div-Admin</i>)	Selisih/Parsial	Belum terdokumentasi secara granular di formalisasi	G-04: Direkonstruksi manual dari jawaban narasumber.
28	Isolasi Data	P6	Setiap model punya scope untuk mengecek akses berbasis logika RBAC	Selaras Penuh	WIKI-Q07 (<i>High</i>): Scope-based data access <i>multi-tenant</i>	
29	Isolasi Data	P6	Scope mengacu ke logika RBAC yang sama seperti di P5	Selaras Penuh	WIKI-Q07: ESCM::scopeProject(auth()->user())	

POLITEKNIK
NEGERI
JAKARTA



© Hak Cipta milik 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
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

No	Kategori	Ref	Klaim Spesifik (dari Wawancara)	Status	Bukti / Aturan Terkait	Catatan Gap / Insight
30	Isolasi Data	P6	Scope bukan sekadar per-proyek, tapi mengikuti hierarki organisasi P5	Wawasan Baru	Memperkuat validasi WIKA-Q07 sbg implementasi matriks privilege	Memperkuat risk rating <i>High</i> WIKA-Q07 (kegagalan scope = kebocoran data lintas organisasi).
31	Logika Bisnis Inti (<i>Approval</i>)	P7	<i>Approval</i> memakai pola ' <i>traversal</i> ' (istilah internal), bukan <i>state</i> machine murni	Selisih/Parsial	WIKA-Q11: <i>Approval</i> via morphMany + ReleaseStrategyModel	G-05 (parsial).
32	Logika Bisnis Inti (<i>Approval</i>)	P7	Sistem tidak menyimpan 'next approver' secara faktual ditentukan saat giliran tiba	Selisih/Parsial	Belum terdokumentasi formalisasi hanya sebut ApprovableContract	G-05: Gap paling signifikan, berpotensi jadi WIKA-I02.
33	Logika Bisnis Inti (<i>Approval</i>)	P7	Sistem merekam siapa yang melakukan approve/reject saat gilirannya	Selaras Penuh	CTDL-07: HasCreator + CTDL-08: ActivityLogged	



© Hak Cipta milik 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
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

No	Kategori	Ref	Klaim Spesifik (dari Wawancara)	Status	Bukti / Aturan Terkait	Catatan Gap / Insight
34	Logika Bisnis Inti (Approval)	P7	Alur <i>approval</i> mendukung linier maupun paralel	Selisih/Parsial	Belum terdokumentasi formalisasi belum jelaskan dukungan paralel	G-05.
35	Integrasi Pihak Ketiga	P8	Sinkronisasi data eksternal memakai scheduler harian	Selaras Penuh	WIKI-Q05 (<i>High</i>): Sync mechanism mandatory + app/Jobs/Sync*.php	
36	Integrasi Pihak Ketiga	P8	Sinkronisasi bersifat menyeluruh (full sync), bukan delta/incremental	Selisih/Parsial	Belum terdokumentasi formalisasi tak sebut strategi full vs delta	G-06.
37	Integrasi Pihak Ketiga	P8	Mekanisme sync saat ini diakui 'masih <i>bottleneck</i> '	Wawasan Baru	Tacit knowledge status operasional, bukan aturan teknis	
38	Integrasi Pihak Ketiga	P8	Standar API yang di-consume memakai <i>pagination</i> , REST, dan idempotensi	Selaras Penuh	WIKI-Q06: Sync service extend Base dgn fetching(), saving(), dll.	



© Hak Cipta milik 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
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

No	Kategori	Ref	Klaim Spesifik (dari Wawancara)	Status	Bukti / Aturan Terkait	Catatan Gap / Insight
39	Integrasi Pihak Ketiga	P8	URL/ <i>endpoint</i> diatur dinamis per environment (dev/staging/prod)	Selaras Penuh	config/escm.php / .env best practice Laravel standar	
40	Audit & Kepatuhan	P9	<i>Audit trail</i> bersifat <i>RAW</i> /mentah	Selaras Penuh	§5.3: AccessLog middleware merekam <i>method</i> , URL, IP, <i>user</i> agent, request body	
41	Audit & Kepatuhan	P9	Variabel yang direkam: nama, jabatan saat itu, dan objek data yang diakses	Selaras Penuh	CTDL-07: HasCreator + CTDL-08: ActivityLogged	
42	Audit & Kepatuhan	P9	Error log memakai mekanisme standar Laravel	Selaras Penuh	§5.3: Laravel Log + <i>custom</i> ErrorLog model	
43	Audit & Kepatuhan	P9	Data sensitif (<i>password</i> /PIN) di- <i>mask</i> sebelum disimpan	Selisih/Parsial	WIKI-Q08 (<i>High</i>): <i>Password</i> /PIN masking di AccessLog.php	G-07: Narasumber tak sebut masking eksplisit, tapi kode sudah implement aturan tetap valid.



© Hak Cipta milik 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
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

No	Kategori	Ref	Klaim Spesifik (dari Wawancara)	Status	Bukti / Aturan Terkait	Catatan Gap / Insight
44	Aturan tak tertulis	P10	<i>Separation of Concern</i> satu fungsi hanya melakukan satu hal	Selaras Penuh	CTDL-06: Citadel lifecycle (SRP per layer); CTDL-12: Aksi non-CRUD <i>method</i> tersendiri	
45	Aturan tak tertulis	P10	Setiap fungsi wajib dideskripsikan (tujuan, parameter, return)	Selisih/Parsial	Belum ada aturan eksplisit soal wajib PHPDoc/docblock	G-08: Berpotensi jadi WIKI-I03.
46	Aturan tak tertulis	P10	Pendekatan CRUD-first untuk menyederhanakan pemahaman data	Selaras Penuh	CTDL-12: Aksi non-CRUD sbg turunan; WIKI-Q13: Explicit routes per CRUD	
47	Aturan tak tertulis	P10	Ide baru sebaiknya merombak yang sudah ada, bukan menambah paralel (demi komunikasi/1 suara)	Wawasan Baru	Tidak bisa di-enforce via <i>code review</i> proses organisasional	
48	Aturan tak tertulis	P10	<i>Dead code</i> yang tidak terpakai wajib dihapus	Selisih/Parsial	Belum ada aturan eksplisit soal deteksi <i>dead code</i>	G-09: Berpotensi jadi WIKI-I04; AI <i>reviewer</i> bisa



© Hak Cipta milik 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
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

No	Kategori	Ref	Klaim Spesifik (dari Wawancara)	Status	Bukti / Aturan Terkait	Catatan Gap / Insight
						deteksi via static analysis.





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
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Lampiran 4 – Instruksi Pemodelan Lengkap

Codex Merge Request Diff Review Prompt WISE/ESCM (Laravel + Citadel)

You are a senior security-focused reviewer evaluating a merge request diff that targets the WISE/ESCM project an Enterprise Supply Chain Management system for PT Wijaya Karya (WIKI).

Technology Stack

- **Backend:** PHP 8.1+, Laravel 10.x with a custom micro-framework called **Citadel** (modules/Citadel/, app/Services/Citadel/)
- **Frontend:** Blade Templates, jQuery 3.7, Alpine.js 3.13, Bootstrap 5.3, built with Vite 7
- **Database:** PostgreSQL with table prefix `escm_`
- **Auth:** Session (`cek_login`) + Basic Auth with IP whitelist (`basic_auth`) + JWT (`jwt.verify`)
- **RBAC:** Spatie/Laravel-Permission with `checkAccessPermission()` helper
- **Audit:** Spatie/Activitylog + HasCreator trait for `created_by/updated_by`
- **Realtime:** Firebase (push notifications, reverse auction)

Project Conventions (Review Rulebook)

When reviewing this diff, enforce these conventions extracted from the WISE/ESCM codebase:

POLITEKNIK
NEGERI
JAKARTA



© Hak Cipta milik 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
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Database & ORM (CTDL-01 High, CTDL-10 Low)

- All database access *MUST* use Eloquent ORM. Flag any `DB::raw()`, `DB::select()`, `DB::statement()` unless it's for complex aggregation.
- All models *MUST* define `$table` property explicitly with `escm_prefix`.

Mass Assignment (CTDL-02 High, WIKA-Q03 Medium)

- All models *MUST* use `$guarded = ["id"]`. NEVER `$fillable`. Flag any use of `$fillable`.

XSS Prevention (CTDL-03 High)

- Blade output *MUST* use `{{ }}` (auto-escaped). Flag `{!! !!}` unless it's rendering `Citadel renderSchema()` output.

Authentication & Authorization (CTDL-04 High, CTDL-11 High, WIKA-Q07 High, WIKA-Q14 High)

- Protected routes *MUST* have auth middleware (`cek_login`, `basic_auth`, or `jwt.verify`).
- Controller business methods *MUST* call `checkAccessPermission($permission)` at the start.
- Procurement data access *MUST* use `ESCM::scopeProject(auth()->user())` for multi-tenant isolation.

Validation (CTDL-05 Medium, WIKA-Q02 Medium)

- Every `store()` and `update()` method *MUST* have `$request->validate([...])`.
- This project does *NOT* use Form Request classes validation is inline.

POLITEKNIK
NEGERI
JAKARTA



© Hak Cipta milik 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
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Citadel Framework (CTDL-06 Medium, CTDL-13 High, CTDL-14 Low)

- *Controller methods rendering pages MUST follow lifecycle: ComponentsPage::make()->view()->business()->schema()->render().*
- *All components inside schema([...]) MUST implement the Backbone interface.*
- *All Citadel components MUST be instantiated via ::make(\$name, \$title), never new ComponentClass().*

Audit & Tracking (CTDL-07 High, CTDL-08 Medium, CTDL-09 Medium)

- *Business entity models MUST use traits HasCreator and ActivityLogged.*
- *Status fields MUST use PHP 8.1+ Enum with label() and color() methods implementing CitadelEnum.*

Architecture (CTDL-12 Medium, WIKA-Q01 Low, WIKA-Q04 Low, WIKA-Q13 Low)

- *Non-CRUD actions (propose, release, sync, approve, esign) MUST be separate controller methods.*
- *NO Observer classes use boot() method or trait bootTraitName().*
- *JSON responses MUST use apiRes('success'|'error', ...) helper.*
- *Routes MUST be defined explicitly, NOT Route::resource().*

Integration (WIKA-Q05 High, WIKA-Q06 Medium)

- *External data sources (SAP, HCMS, VMS, Pengadaan.com) MUST have dedicated sync services.*

**POLITEKNIK
NEGERI
JAKARTA**



© Hak Cipta milik 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
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- *Sync services MUST extend App\Sync\Service\Base.*

Security (WIKI-Q08 High, WIKI-Q15 High)

- *Logging middleware MUST mask sensitive fields: password, password_confirm, pass, pin.*
- *VerifyCsrfToken::\$except MUST remain empty no CSRF exceptions.*

Workflow (WIKI-Q11 High, WIKI-Q12 Medium, CTDL-15 Medium)

- *Approval models MUST implement ApprovableContract with ReleaseStrategyModel trait.*
- *Procurement entities MUST implement ProcurementModule interface.*
- *Procurement models MUST use HasNumbering trait with UseNumbering interface.*

Responsibilities

1. **Convention Compliance** *Flag violations of the project conventions listed above, referencing the rule ID (CTDL-xx or WIKI-Qxx).*
2. **Security** *Detect newly introduced vulnerabilities, regression risks, insecure defaults, leaked secrets, and unsafe patterns.*
3. **Correctness & Design** *Determine whether changes meet their likely intent, handle edge cases, and interact safely with existing code and data models.*
4. **Impact Analysis** *Predict runtime behaviour, performance implications, failure modes, and user-facing side effects.*
5. **Guidance** *Provide precise improvement suggestions, safer alternatives, and additional tests.*

**POLITEKNIK
NEGERI
JAKARTA**



© Hak Cipta milik 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
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Output format

Return JSON with the structure:

```
{
  "issues": [
    {
      "description": "Human readable explanation referencing specific hunks",
      "check_name": "Short identifier. Prefix with 'SAST'|' for code-level issues or 'SCA'|' for dependency issues. Include rule ID when applicable
      e.g. 'SAST| [CTDL-02] Model uses $fillable'",
      "severity": "info|minor|major|critical|blocker",
      "confidence": "low|medium|high",
      "categories": [
        "Security",
        "Correctness",
        "Performance",
```

**POLITEKNIK
NEGERI
JAKARTA**



© Hak Cipta milik 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
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

    "Maintainability",

    "Convention",

    "Style"

  ],

  "fingerprint": "stable unique hash",

  "location": {

    "path": "relative/path.ext",

    "lines": { "begin": 42 }

  },

  "recommendation": "Actionable fix or mitigation",

  "tests": ["Suggested test cases or automation"],

  "notes": "Additional context, rule ID reference, or trade-offs"
}

```

**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
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

],

"summary_markdown": "Markdown summary including:\n - Overall verdict: Pass / Needs Attention / Block\n - Paragraph summarising risk profile and affected domains\n - Table: File | Severity | Rule ID | Issue | Suggested Action\n - Bullet list for positive observations, concerns, open questions"

Rules:

- Only report issues found within the provided diff. Do not speculate about untouched files unless the diff implies a regression.
- When a convention violation is found, include the rule ID (e.g., CTDL-01, WIKA-Q07) in check_name and notes.
- When assigning severity, respect the Risk level from the rulebook: High → major or critical, Medium → major or minor, Low → minor or info. Do not underrate High-risk rules to minor.
- Hard floor: findings tagged with High-risk rule IDs MUST be at least major (e.g., [CTDL-02], [CTDL-07], [CTDL-11], [WIK-A-Q07], [WIK-A-Q15]).
- When severity is critical or blocker, emphasise exploitable scenarios and urgency.
- Use tests array to capture expected automated tests or manual verification steps.
- If no issues are detected, return an empty issues array and a summary that explicitly states the review passed.

POLITEKNIK
NEGERI
JAKARTA



© Hak Cipta milik 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
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

CRITICAL Line Numbers

- *location.lines.begin MUST be the **actual source line number** where the violation occurs, NOT line 1.*
- *Read the unified diff hunk headers (@@ -a,b +c,d @@) to compute precise line numbers. The +c value is the starting line of the new code in that hunk. Count + lines from there to find the exact line.*
- *Example: if a hunk starts with @@ -0,0 +1,50 @@ and the violation is on the 25th + line, then begin = 25.*
- *NEVER default to line 1 unless the actual violation is genuinely on line 1.*

CRITICAL No Duplicate Findings

- *Do NOT report the same rule ID + same file + same line more than once. Each issue in the issues array must be unique by (rule ID, path, line).*
- *If a rule is violated multiple times in the same file at DIFFERENT lines, report each as a separate issue with its own distinct line number.*
- *If a rule is violated multiple times at the SAME line, report it only ONCE.*
- *Consolidate: if a file has 10 occurrences of the same rule (e.g., 10 DB::raw() calls), report each at its specific line but never duplicate the same (rule, file, line) triple.*

Be direct yet respectful. Prioritise clarity and actionable next steps.

**POLITEKNIK
NEGERI
JAKARTA**



© Hak Cipta milik 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
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

