Rancang Bangun *Smart Composter* Berbasis IoT dan *Machine Learning* Dengan Sistem *Monitoring* Berbasis *Firebase* dan *Flutter*

Sub Judul

Rancang Bangun *Backend Server Smart Composter* Menggunakan Arsitektur *Client Server*

SKRIPSI

Adipraja Putra Ramadhan

Nim.2207421022

PROGRAM STUDI TEKNIK MUL*TIME*DIA DAN JARINGAN

JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER

POLITEKNIK NEGERI JAKARTA

2026

Rancang Bangun *Smart Composter* Berbasis IoT dan *Machine Learning* Dengan Sistem *Monitoring* Berbasis *Firebase* dan *Flutter*

Sub Judul

Rancang Bangun *Backend Server Smart Composter* Menggunakan Arsitektur *Client Server*

SKRIPSI

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

Adipraja Putra Ramadhan

Nim.2207421022

PROGRAM STUDI TEKNIK MUL*TIME*DIA DAN JARINGAN

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

SURAT PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan di bawah ini:

Nama : Adipraja Putra Ramadhan

NIM : 2207421022

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

Judul skripsi : Rancang Bangun *Backend Server Smart composter* Menggunakan Arsitektur *Client Server*

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, 22 Juli 2026

Yang membuat pernyataan

(Adipraja Putra Ramadhan)

NIM. 2207421022

© 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 : Adipraja Putra Ramadhan

NIM : 2207421022

Program Studi : Teknik Multimedia dan Jaringan

Judul Skripsi : Rancang Bangun *Backend Server* Smart Composter Menggunakan Arsitektur *Client Server*

Telah diuji oleh tim penguji dalam Sidang Skripsi pada hari Rabu, Tanggal 2, Bulan Juli, Tahun 2026, dan dinyatakan **LULUS.**

Disahkan oleh

Pembimbing I : Dr. Indra Hermawan, S.Kom., M.Kom. ( )

Penguji I : Dr. Prihatin Oktivasari,S.Si.,M.Si ( )

Penguji II : Susana Dwi Yulianti, M.Kom ( )

Penguji III : Dwi Ermawati MT ( )

Mengetahui :

Jurusan Teknik Informatika dan Komputer

Ketua

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

NIP. 197908032003122003

Dipindai dengan CamScanner

© 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

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas limpahan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir berjudul "Rancang Bangun *Backend Server Smart Composter* Menggunakan Arsitektur *Client Server*" dengan baik dan tepat waktu. Selama proses penyusunan tugas akhir ini, penulis mendapatkan banyak bantuan, bimbingan, dan dukungan dari berbagai pihak yang berperan penting dalam kelancaran pelaksanaan penelitian hingga penyusunan laporan ini.

Oleh karena itu, penulis menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Tuhan Yang Maha Esa atas segala rahmat, perlindungan, dan kekuatan yang telah diberikan selama proses penelitian dan penyusunan laporan ini.
2. Kedua orang tua penulis yang selalu memberikan doa, semangat, dan dukungan moral maupun materi dalam setiap tahap kehidupan penulis.
3. Bapak Dr. Indra Hermawan, S.Kom., M.Kom., selaku dosen pembimbing yang telah memberikan bimbingan, arahan, dan motivasi secara konsisten selama proses penyusunan Tugas Akhir ini.
4. Rekan satu kelompok yang telah bekerja sama dan berkontribusi secara aktif dalam pelaksanaan dan penyusunan Tugas Akhir ini.
5. Rekan-rekan seperjuangan dan teman-teman di Politeknik Negeri Jakarta yang telah menjadi tempat berbagi ilmu, saling mendukung, dan menemani proses perkuliahan hingga penyusunan tugas akhir.

Penulis menyadari laporan ini masih memiliki kekurangan, sehingga kritik dan saran yang membangun sangat diharapkan. Semoga laporan ini bermanfaat dan dapat memberikan kontribusi di bidang pemantauan lingkungan berbasis IoT.

Depok, 22 Juli 2026

Adipraja Putra Ramadhan

© 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

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS

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

Nama : Adipraja Putra Ramadhan
NIM 2207421022
Jurusan/Program Studi : Teknik Informatika dan Komputer / Teknik Multimedia dan Jaringan

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

Rancang Bangun *Backend Server* Smart Composter Menggunakan Arsitektur *Client Server*

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non - Eksklusif ini Politeknik Negeri Jakarta berhak menyimpan, mengalih mediakan/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, 22 Juli 2026 Yang Menyatakan

(Adipraja Putra Ramadhan)
NIM. 2207421022

© 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

ABSTRAK

Pengelolaan proses pengomposan memerlukan sistem monitoring yang mampu menyediakan data secara real-time, terintegrasi, dan mudah diakses. Penelitian ini bertujuan merancang, mengimplementasikan, dan menguji backend server Smart Composter menggunakan arsitektur Client Server berbasis Firebase. Layanan yang digunakan meliputi Firebase Authentication, Firebase Realtime Database, Firebase Cloud Messaging (FCM), Firebase Cloud Functions, dan Firebase Hosting. Metode penelitian yang diterapkan adalah Research and Development (R&D) yang mencakup identifikasi kebutuhan, perancangan, implementasi, pengujian, dan evaluasi. Pengujian sistem meliputi Black box Testing, Software Quality Assurance (SQA) berdasarkan ISO/IEC 25010 pada aspek usability, Quality of Service (QoS) menggunakan parameter delay, jitter, throughput, dan packet loss, serta reliability Testing menggunakan parameter availability, success rate, failure rate, dan integritas data. Hasil penelitian menunjukkan seluruh fitur backend berfungsi sesuai kebutuhan dengan tingkat keberhasilan Black box Testing sebesar 100% dan nilai usability sebesar 77,5% (kategori baik). Hasil QoS menunjukkan rata-rata delay perubahan data sensor sebesar 48,27 ms, rata-rata jitter sebesar 4,81 ms, rata-rata throughput sebesar 21,7 Kbps, dan packet loss sebesar 0%. Pengujian reliability menghasilkan availability, success rate, dan integritas data sebesar 100% dengan failure rate 0%, sehingga backend server berbasis Firebase mampu mendukung pengelolaan data Smart Composter secara andal.

Kata kunci: *Smart Composter, Firebase, Backend Server, Realtime Database, QoS*

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

DAFTAR ISI

SURAT PERNYATAAN BEBAS PLAGIARISME ii
LEMBAR PENGESAHAN iii
KATA PENGANTAR iv
SURAT PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS v
ABSTRAK vi
DAFTAR ISI vii
DAFTAR GAMBAR x
DAFTAR TABEL xii
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Perumusan Masalah 3
1.3 Batasan Masalah 4
1.4 Tujuan dan Manfaat 4
1.4.1 Tujuan 4
1.4.2 Manfaat 5
1.5 Sistematika Penulisan 5
BAB II TINJAUAN PUSTAKA 6
2.1 Penelitian Terkait 6
2.2 *Server* 9
2.3 *Cloud Computing* 9
2.4 *Cloud Database* 10
2.5 *Client Server* Arsitektur 10
2.6 *Rest API* 11
2.7 *Firebase* 11
2.8 *Firebase* Real*time Database* 12
2.9 *Firebase Authentication* 12
2.10 *Firebase Cloud Messaging* 12
2.11 *Firebase Hosting* 12
2.12 *Firebase Cloud Functions* 13

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

2.13 *Research* and *Development* 13
2.14 Keamanan *Firebase* 13
2.15 Perbandingan Platform *Backend as a Service* (*BaaS*) 15
2.16 *Flutter* 16
2.17 *Dart* 17
2.18 *Black box Testing* 17
2.19 SQA *Testing* 18
2.20 QoS *Testing* 19
2.20.1 *Delay* 19
2.20.2 *Jitter* 20
2.20.3 *Throughput* 20
2.20.4 *Packet loss* 21
2.21 *Reliability Testing* 22
BAB III METODE PENELITIAN 23
3.1 Rancangan Penelitian 23
3.2 Tahapan Penelitian 23
3.3 Objek Penelitian 26
BAB IV HASIL DAN PEMBAHASAN 28
4.1 Analisis Kebutuhan 28
4.1.1 Analisis Kebutuhan Fungsional 28
4.1.2 Analisis Kebutuhan Non-Fungsional 29
4.2 Perancangan Sistem 30
4.2.1 Perancangan Arsitektur Sistem 30
4.2.2 Perancangan *Database* 32
4.2.3 Perancangan Antarmuka Android 34
4.2.4 Perancangan Antarmuka *Websitesite* 34
4.2.5 *Flowchart* Aplikasi 35
4.2.6 *Flowchart* Notifikasi 36
4.2.7 *Flowchart* Siklus Baru 36
4.3 Implementasi Sistem 38
4.3.1 Implementasi *Firebase* 38
4.3.2 Implementasi Keamanan *Firebase* 40

© 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

4.3.3 Implementasi *Firebase Authentication* 41
4.3.4 Implementasi *Firebase* Real*time Database* 43
4.3.5 Implementasi *Firebase Hosting* 47
4.3.6 Implementasi *Firebase Cloud Messaging* 48
4.3.7 Implementasi *Firebase Cloud Functions* 50
4.3.8 Implementasi Antarmuka Android 50
4.3.9 Implementasi Antarmuka *Website* 58
4.4 Pengujian 64
4.4.1 Deskripsi Pengujian 65
4.4.2 Prosedur Pengujian 65
4.4.3 Data Hasil Pengujian 69
4.4.4 Analisis Data / Evaluasi Pengujian 79
BAB V PENUTUP 81
5.1 Kesimpulan 81
5.2 Saran 81
DAFTAR PUSTAKA 82
DAFTAR RIWAYAT HIDUP 87
LAMPIRAN 88

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

DAFTAR GAMBAR

Gambar 2.1 Kategori *Delay* 20
Gambar 2.2 Kategori *Jitter* 20
Gambar 2.3 Kategori *Throughput* 21
Gambar 2.4 Kategori *Packet loss* 22
Gambar 3.1 Tahapan R&D mengacu pada Gusdevi et al. (2023) 23
Gambar 4.1 Arsitektur Sistem 30
Gambar 4.2 *Flowchart* Aplikasi 35
Gambar 4.3 *Flowchart* Notifikasi 36
Gambar 4.4 *Flowchart* Siklus Pengomposan 37
Gambar 4.5 Implementasi *Firebase* 39
Gambar 4.6 Konfigurasi *Firebase* pada lingkungan pengembang 39
Gambar 4.7 Konfigurasi Keamanan *Firebase* 40
Gambar 4.8 Settingan *Firebase Authentication* 42
Gambar 4.9 Penyesuaian *Template Authentication* 42
Gambar 4.10 Kebijakan *Password* 43
Gambar 4.11 Konfigurasi *Firebase* Real*time Database* 44
Gambar 4.12 Struktur Real*time Database Smart Composter* 45
Gambar 4.13 Konfigurasi *Firebase Hosting* 47
Gambar 4.14 Setup *Firebase Cloud Messaging* 49
Gambar 4.15 Konfigurasi *Cloud* Function *Firebase* 50
Gambar 4.16 Halaman *Login* 51
Gambar 4.17 Halaman Register 52
Gambar 4.18 Halaman *Forgot Password* 53
Gambar 4.19 Halaman Reset *Password* 54
Gambar 4.20 Halaman Pemilihan Bin Kompos 55
Gambar 4.21 Halaman *Dashboard Monitoring* 56
Gambar 4.22 Halaman Riwayat Kompos 57
Gambar 4.23 Halaman Kelola Pengguna 58
Gambar 4.24 Halaman *Login Website* 59
Gambar 4.25 Halaman Register *Website* 59
Gambar 4.26 Halaman *Forgot Password Website* 60

Gambar 4.27 Halaman Reset *Password Website* 60
Gambar 4.28 Halaman Pemilihan Bin Kompos *Website* 61
Gambar 4.29 Halaman *Dashboard Monitoring Website* 62
Gambar 4.30 Halaman Riwayat Kompos *Website* 63
Gambar 4.31 Halaman Kelola Pengguna 63
Gambar 4.32 Grafik Hasil Pengujian SQA 72

© 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

© 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

DAFTAR TABEL

Tabel 2.1 Penelitian Terkait 6
Tabel 2.2 Perbandingan *Firebase* dan Supabase 15
Tabel 4.1 Pertanyaan SQA 67
Tabel 4.2 Prosedur Pengujian *Reliability Server* 68
Tabel 4.3 Hasil Pengujian *Black box* 69
Tabel 4.4 Hasil Pengujian *Delay* 73
Tabel 4.5 Hasil Pengujian *Jitter* 74
Tabel 4.6 Hasil Pengujian *Throughput* 75
Tabel 4.7 Hasil Pengujian *Packet loss* 77
Tabel 4.8 Hasil Pengujian *Reliability Server* 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

BAB I
PENDAHULUAN

1.1 Latar Belakang

Berdasarkan data Sistem Informasi Pengelolaan Sampah Nasional (SIPSN) tahun 2025, timbulan sampah di Indonesia yang berasal dari 514 kabupaten/kota masih tergolong tinggi. Total timbulan sampah nasional pada tahun 2025 diperkirakan mencapai lebih dari 144.800 ton per hari, dengan jumlah sampah yang terkelola sebesar 35.747 ton per hari (SIPSN, 2025).

Provinsi Jawa Barat sebagai salah satu wilayah dengan jumlah penduduk tinggi menjadi penyumbang timbulan sampah yang signifikan, termasuk wilayah Kota Depok. Pada tahun 2025 Kota Depok menyumbang sampah sebanyak 1.277 ton per hari (SIPSN, 2025).

Salah satu metode pengelolaan sampah organik yang banyak diterapkan adalah pengomposan. Namun, proses pengomposan memerlukan pemantauan terhadap berbagai parameter seperti suhu, kelembapan, aerasi, dan kandungan nutrisi untuk memastikan proses dekomposisi berlangsung dengan baik dan menghasilkan kompos yang berkualitas. Pada metode pengomposan konvensional, beberapa proses masih melibatkan tenaga manusia secara langsung sehingga memerlukan pengawasan yang berkelanjutan. Oleh karena itu, diperlukan *Smart Composter* yang mampu membantu proses pemantauan kondisi kompos secara lebih efektif dan berkelanjutan (Pillai, 2023).

Kondisi tersebut menunjukkan perlunya suatu sistem *monitoring* yang mampu mengelola dan menyediakan data kondisi kompos secara *real-time*, terintegrasi, dan mudah diakses oleh pengguna. Agar data *monitoring* dapat tersimpan, disinkronkan, dan didistribusikan secara berkelanjutan, diperlukan sebuah *backend server* yang mampu menangani pengelolaan data, autentikasi pengguna, serta mekanisme notifikasi secara terpusat. Pemanfaatan layanan *backend* berbasis *cloud* memungkinkan sinkronisasi data secara *real-time*, pengelolaan pengguna yang aman, dan distribusi informasi yang lebih efektif kepada pengguna sistem (Alsayaydeh, Bacarra, Gani, Mamchenko and Herawan, 2025).

© 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

Firebase merupakan salah satu platform *Backend as a Service* (*BaaS*) yang menyediakan layanan Real*time Database*, *Authentication*, *Cloud Messaging*, *Hosting*, dan *Cloud Functions* yang dapat dimanfaatkan untuk membangun *backend server* tanpa perlu mengelola infrastruktur *server* secara mandiri. Dengan memanfaatkan layanan tersebut, *backend server* dapat mendukung proses penyimpanan data *monitoring*, sinkronisasi data secara *real-time*, manajemen pengguna, serta penyampaian notifikasi kepada pengguna sistem (Prayoga and Yudistira, 2025).

Berdasarkan permasalahan tersebut, penelitian ini berfokus pada rancang bangun *backend server* berbasis *Firebase* pada *Smart Composter* yang digunakan untuk mendukung pengelolaan data *monitoring* pengomposan.

Beberapa penelitian sebelumnya telah mengembangkan sistem *monitoring* kompos berbasis Internet *of Things* (IoT) dengan berbagai platform dan metode komunikasi. Sasmita Prawira Diharja et al. (2025) mengembangkan sistem *monitoring* berbasis arsitektur *Client Server* dengan memanfaatkan Telegram Bot API sebagai media komunikasi. Namun, sistem ini memiliki sejumlah keterbatasan yang cukup signifikan. Fungsi *backend* masih terbatas pada penyampaian informasi dan belum mampu melakukan pengelolaan data secara terpusat. Selain itu, tidak tersedia mekanisme penyimpanan data historis sensor pada *database server*, sehingga data pemantauan tidak dapat dianalisis kembali dalam jangka panjang. Sistem juga belum mendukung sinkronisasi data secara real-*time* antar perangkat atau pengguna, serta fleksibilitas pengembangan layanan *backend* menjadi terbatas karena akses data sepenuhnya bergantung pada platform Telegram.

Rindang Diannita et al. (2025) membangun sistem dengan arsitektur *Client Server* menggunakan platform Blynk IoT berbasis protokol MQTT/HTTP. Meskipun sistem ini mampu melakukan *monitoring* secara dasar, terdapat keterbatasan pada aspek pengelolaan data historis yang belum didukung secara penuh sehingga pengembangan fitur lanjutan menjadi terhambat. Di samping itu, sistem belum dilengkapi dengan mekanisme autentikasi dan pengelolaan hak akses pengguna, yang berpotensi menimbulkan risiko keamanan dalam pengoperasian sistem secara nyata.

© 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

Mohammad Aprizal et al. (2025) merancang sistem berbasis Wireless Sensor *Network* (WSN) dengan kombinasi protokol LoRa dan HTTP. Keterbatasan utama sistem ini terletak pada ketergantungan pengelolaan data terhadap *ESP32 Node*MCU yang difungsikan sebagai *server* lokal, sehingga kapasitas penyimpanan dan kemampuan pengolahan data menjadi lebih terbatas dibandingkan penggunaan *server* terpusat. Ketersediaan layanan *monitoring* pun menjadi tidak sepenuhnya andal karena sangat bergantung pada kondisi fisik dan konektivitas perangkat *ESP32 Node*MCU, sehingga akses data dapat terganggu sewaktu-waktu apabila perangkat mengalami gangguan atau kehilangan koneksi jaringan.

Syaifan Fadlan Riady Bachri (2022) mengimplementasikan sistem WSN dengan mengombinasikan protokol LoRa dan Telegram Bot API untuk keperluan *monitoring* jarak jauh. Sistem ini menghadapi permasalahan *packet loss* yang cukup tinggi, yakni berkisar antara 3 hingga 60 persen pada komunikasi LoRa, terutama ketika terdapat banyak penghalang fisik dan kualitas antena yang digunakan kurang optimal. Selain itu, data sensor disimpan pada *server* lokal berbasis *Raspberry Pi*, sehingga ketersediaan dan aksesibilitas data sangat bergantung pada kondisi operasional perangkat *server* lokal tersebut dan rentan mengalami gangguan apabila perangkat tidak berfungsi dengan baik.

Sasirekha Gurla Venkata Kameswari et al. (2025) mengembangkan sistem WSN dengan memanfaatkan layanan AWS IoT sebagai infrastruktur komunikasi. Keterbatasan utama yang diidentifikasi pada penelitian ini adalah jangkauan protokol jaringan XBee yang digunakan hanya mencapai sekitar 100 meter, sehingga cakupan area *monitoring* menjadi sangat terbatas dan kurang cocok untuk diterapkan pada lingkungan dengan skala yang lebih luas atau lokasi yang tersebar secara geografis.

1.2 Perumusan Masalah

Berdasarkan hal-hal yang sudah disampaikan di atas, maka rumusan permasalahan yang menjadi fokus pada *Smart Composter* yang dibangun:

1. Bagaimana merancang arsitektur *backend server* berbasis *Firebase* pada *Smart Composter*?
2. Bagaimana mengimplementasikan layanan *Firebase* untuk pengelolaan data, autentikasi pengguna, dan notifikasi pada *Smart Composter*?

© 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. Bagaimana menguji *backend server Firebase* yang telah dibangun pada *Smart Composter*?

1.3 Batasan Masalah

Beberapa batasan masalah yang dibatasi untuk memfokuskan dan mengefisiensikan waktu penelitian di antaranya yaitu:

1. Penelitian ini dibatasi pada perancangan dan pengembangan *backend server* berbasis *Firebase* untuk mendukung *Smart Composter*.
2. Implementasi *backend server* dibatasi pada penggunaan layanan *Firebase*, yaitu *Firebase Authentication*, *Firebase* Real*time Database*, *Firebase Cloud Messaging*, *Firebase Hosting*, dan *Firebase Cloud Functions* untuk mendukung autentikasi pengguna, pengelolaan data, sinkronisasi data, layanan *Website monitoring*, dan pengiriman notifikasi.
3. Pengujian sistem dibatasi pada pengujian fungsionalitas layanan *backend Firebase* menggunakan metode *Black box Testing*, pengujian *Software Quality Assurance* (SQA) menggunakan standar ISO/IEC 25010 pada aspek *usability*, pengujian *Quality of Service* (QoS) menggunakan parameter *delay*, *jitter*, *throughput* dan *packet loss*, serta pengujian *reliability server* menggunakan parameter *availability*, *success rate*, *failure rate*, dan integritas data.

1.4 Tujuan dan Manfaat

Berikut ini tujuan dan manfaat dari penelitian ini.

1.4.1 Tujuan

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Merancang dan membangun *backend server* berbasis *Firebase* untuk mendukung *Smart Composter* dalam pengelolaan data *monitoring*.
2. Mengintegrasikan layanan *Firebase* yang meliputi *Firebase Authentication*, *Firebase* Real*time Database*, *Firebase Cloud Messaging*, *Firebase Hosting*, dan *Firebase Cloud Functions* untuk mendukung autentikasi pengguna, pengelolaan data, sinkronisasi data, layanan *Website*, dan pengiriman notifikasi.
3. Mengevaluasi kinerja dan keandalan *backend server Firebase* dalam mendukung proses penyimpanan, pengolahan, sinkronisasi, distribusi data, dan pengiriman notifikasi pada *Smart Composter* melalui pengujian fungsionalitas, *Quality of Service* (QoS), dan *reliability server*.

© 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

1.4.2 Manfaat

1. Menyediakan *backend server* berbasis *Firebase* yang mampu mengelola data *monitoring Smart Composter* secara terintegrasi.
2. Mendukung proses autentikasi pengguna, penyimpanan data, sinkronisasi data, dan pengiriman notifikasi melalui layanan *Firebase*.
3. Meningkatkan kemudahan pengelolaan dan akses data *monitoring Smart Composter* melalui layanan *backend* yang terpusat.
4. Menjadi referensi dalam pengembangan *backend server* berbasis *Firebase* pada sistem *monitoring* lingkungan dan Internet *of Things* (IoT).
5. Memberikan gambaran penerapan platform *Backend as a Service* (*BaaS*) menggunakan *Firebase* untuk mendukung pengembangan sistem *monitoring*.

1.5 Sistematika Penulisan

BAB I PENDAHULUAN

Bab ini membahas latar belakang penelitian, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian, serta sistematika penulisan yang berkaitan dengan perancangan dan implementasi *backend server* berbasis *Firebase* pada *Smart Composter*.

BAB II TINJAUAN PUSTAKA

Bab ini berisi tinjauan pustaka dan landasan teori yang mendukung penelitian, meliputi konsep *backend server*, *Backend as a Service* (*BaaS*), layanan *Firebase* seperti *Firebase Authentication*, *Firebase* Real*time Database*, *Firebase Cloud Messaging*, *Firebase Hosting*, dan *Firebase Cloud Functions*, serta penelitian terdahulu yang relevan.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan metodologi penelitian yang digunakan dalam perancangan dan implementasi *backend server* berbasis *Firebase*. Pembahasan meliputi rancangan penelitian, tahapan penelitian dan objek penelitian.

BAB IV HASIL DAN PEMBAHASAN

Bab ini membahas hasil perancangan, implementasi, dan pengujian *backend server* pada sistem *Smart Composter*. Pembahasan mencakup analisis kebutuhan sistem, perancangan arsitektur *backend*, implementasi layanan *Firebase*, hasil pengujian fungsionalitas sistem, serta analisis hasil pengujian.

© 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 V PENUTUP

Bab ini berisi simpulan dari hasil penelitian yang telah dilakukan serta saran untuk pengembangan *backend server* dan *Smart Composter* pada penelitian selanjutnya.

© 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 II
TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian sebelumnya memiliki peranan penting dalam mewujudkan ide atau gagasan penulis dalam melakukan penelitian, terkhusus apabila penelitian tersebut relevan dengan topik yang akan diamati. Berikut merupakan beberapa referensi yang dapat dijadikan sebagai acuan penulis untuk pembuatan Rancang Bangun *Backend Server Smart Composter* Menggunakan Arsitektur *Client Server*.

Tabel 2.1 Penelitian Terkait

Jurnal	Arsitektur Sistem	Keterbatasan
Sasmita Prawira Diharja et al, 2025	Arsitektur Jaringan: *Client Server* Protokol komunikasi: Telegram Bot API (HTTP) Topologi: *Star*	Sistem hanya memanfaatkan Telegram Bot sebagai media *monitoring* dan notifikasi, sehingga fungsi *backend* masih terbatas pada penyampaian informasi, bukan pengelolaan data secara terpusat. Tidak terdapat penggunaan *database server* untuk penyimpanan data historis sensor, sehingga data pemantauan tidak dapat dianalisis kembali dalam jangka panjang. Belum tersedia fitur sinkronisasi data secara *real-time* antar perangkat atau pengguna karena sistem berfokus pada komunikasi melalui chatbot Telegram. Akses data bergantung pada platform Telegram sehingga fleksibilitas pengembangan layanan *backend* menjadi terbatas.
Rindang Diannita et al, 2025	Arsitektur Jaringan: *Client Server* Protokol komunikasi: Blynk	Belum mendukung pengelolaan data historis dan pengembangan fitur.

© 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

Jurnal	Arsitektur Sistem	Keterbatasan
	IoT Platform (MQTT/HTTP) Topologi: *Star*	Tidak terdapat mekanisme autentikasi dan pengelolaan hak akses pengguna.
Mohammad Aprizal et al, 2025	Arsitektur Jaringan: WSN Protokol komunikasi: LoRa + HTTP Topologi: *Star*	Pengelolaan data masih bergantung pada *ESP32 Node*MCU sebagai *server* sehingga kapasitas penyimpanan dan pengolahan data menjadi lebih terbatas. Ketersediaan layanan *monitoring* bergantung pada kondisi dan konektivitas *ESP32 Node*MCU sehingga akses data dapat terganggu ketika perangkat mengalami gangguan atau tidak terhubung ke jaringan.
Syaifan Fadlan Riady Bachri, 2022	Arsitektur Jaringan: WSN Protokol Komunikasi:LoRa, Telegram Bot API Topologi: *Star*	Terjadinya *packet loss* sebesar 3-60% pada komunikasi LoRa, terutama ketika terdapat banyak hambatan (*obstacle*) dan kualitas antena kurang baik. Data sensor disimpan pada *server* lokal *Raspberry Pi*, sehingga akses dan ketersediaan data bergantung pada kondisi perangkat *server* lokal tersebut.
Sasirekha Gurla Venkata Kameswari et al, 2025	Arsitektur Jaringan: WSN Protokol komunikasi: AWS IoT Topologi: *Star*	Keterbatasan jarak penggunaan protokol jaringan XBee dengan jangkauan sekitar 100 meter.

Berdasarkan penelitian-penelitian terdahulu, sistem *monitoring* pengomposan telah berhasil menerapkan berbagai teknologi komunikasi seperti HTTP, Telegram Bot API, Blynk, LoRa, dan XBee untuk mendukung proses pemantauan kondisi

© 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

kompos secara *real-time*. Sistem-sistem tersebut telah mampu melakukan *monitoring* parameter lingkungan seperti suhu, kelembapan, pH, maupun gas serta menyediakan notifikasi dan *dashboard monitoring* bagi pengguna. Namun demikian, masih terdapat beberapa keterbatasan pada aspek *backend*, seperti tidak tersedianya mekanisme autentikasi pengguna, belum optimalnya pengelolaan data historis sensor, serta ketergantungan pada platform pihak ketiga yang membatasi fleksibilitas pengembangan sistem. Selain itu, penggunaan arsitektur Wireless Sensor *Network* (WSN) pada beberapa penelitian juga masih menghadapi kendala berupa keterbatasan jangkauan komunikasi dan terjadinya *packet loss* yang berpotensi menyebabkan hilangnya data selama proses transmisi.

Untuk mengatasi berbagai keterbatasan tersebut, penelitian ini mengusulkan penggunaan *backend* berbasis *Firebase* yang memanfaatkan *Firebase* Real*time Database* sebagai pusat penyimpanan dan sinkronisasi data secara *real-time*. *Firebase* Real*time Database* menggunakan mekanisme komunikasi berbasis *Website*Socket yang memungkinkan koneksi antara *server* dan klien tetap aktif secara berkelanjutan tanpa memerlukan proses polling berulang seperti pada komunikasi HTTP konvensional. Mekanisme tersebut memungkinkan setiap perubahan data dapat didistribusikan secara langsung ke seluruh perangkat yang terhubung sehingga sinkronisasi data menjadi lebih cepat, efisien, dan konsisten. Selain itu, *Firebase Authentication* digunakan untuk menyediakan mekanisme autentikasi dan manajemen pengguna sehingga akses terhadap sistem dapat dikelola secara lebih aman dan terstruktur.

Penelitian ini juga melakukan pengujian *Quality of Service* (QoS) terhadap komunikasi data antara perangkat dan *Firebase* yang meliputi parameter *delay*, *jitter*, *throughput* dan *packet loss*. Pengujian tersebut dilakukan sebagai upaya untuk mengatasi permasalahan *packet loss* yang masih ditemukan pada beberapa penelitian terdahulu, dengan tujuan mengevaluasi tingkat *packet loss* pada komunikasi data sehingga dapat diketahui keandalan proses pengiriman data pada sistem yang dikembangkan. Dengan pendekatan berbasis *cloud* ini, proses penyimpanan, pengelolaan, sinkronisasi, dan distribusi data dapat dilakukan secara lebih terintegrasi serta tidak dibatasi oleh jangkauan komunikasi antar *node* sebagaimana yang umum ditemukan pada arsitektur WSN.

© 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

2.2 *Server*

Server merupakan sistem komputer yang berfungsi untuk menyediakan berbagai layanan dalam suatu jaringan. Umumnya, *server* didukung oleh prosesor yang bersifat scalable, kapasitas RAM yang besar, serta menggunakan sistem operasi khusus yang dikenal sebagai sistem operasi jaringan. Selain itu, *server* menjalankan perangkat lunak administratif untuk mengatur akses terhadap jaringan dan sumber daya yang tersedia, seperti *file* dan printer, sehingga dapat digunakan oleh *client* atau workstation dalam jaringan (Pratama, Siambaton and Haramaini, 2022).

Berdasarkan fungsinya, *server* dapat dibedakan menjadi beberapa jenis, di antaranya application *server* yang digunakan untuk menyediakan aplikasi bagi *client*, data *server* yang berfungsi menyimpan dan mengelola data, serta proxy *server* yang mengatur lalu lintas jaringan. Dalam implementasinya, *server* umumnya menyediakan berbagai layanan berbasis arsitektur *Client Server*, seperti DHCP, DNS, FTP, HTTP, mail *server*, dan lainnya. Setiap layanan tersebut akan merespons permintaan dari *client* sesuai dengan protokol yang digunakan, sehingga memungkinkan komunikasi dan pertukaran data dalam jaringan dapat berjalan dengan baik (Pratama, Siambaton and Haramaini, 2022).

2.3 *Cloud Computing*

Cloud computing merupakan paradigma komputasi yang memungkinkan akses terhadap berbagai sumber daya teknologi, seperti *server*, penyimpanan, basis data, jaringan, perangkat lunak, dan analitik melalui internet. Dengan konsep ini, pengguna tidak perlu lagi memiliki atau mengelola perangkat keras secara langsung di lokasi sendiri, melainkan dapat memanfaatkan layanan yang disediakan oleh pihak ketiga, seperti *Google*, Amazon, dan Micros*oft*. Hal ini memungkinkan penggunaan sumber daya komputasi secara lebih efisien tanpa harus membangun infrastruktur fisik yang mahal (Rezeki and Nasution, 2025).

Cloud computing menawarkan fleksibilitas dan skalabilitas yang tinggi, di mana pengguna dapat menyesuaikan kapasitas layanan sesuai kebutuhan tanpa perlu menambah perangkat baru. Seluruh pengelolaan, termasuk pemeliharaan, pembaruan sistem, dan keamanan, ditangani oleh penyedia layanan, sehingga pengguna dapat lebih fokus pada pengembangan aplikasi atau bisnis. Dengan

© 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

kemudahan akses melalui internet, *cloud computing* menjadi solusi penting dalam mendukung transformasi digital karena memberikan efisiensi, kecepatan, dan kemudahan dalam pemanfaatan teknologi (Rezeki and Nasution, 2025).

2.4 *Cloud Database*

Cloud database merupakan jenis basis data yang dapat diakses oleh klien melalui layanan *cloud* yang didistribusikan oleh penyedia layanan melalui internet. Dengan adanya *cloud database*, pengguna tidak perlu lagi menyimpan data secara lokal pada perangkat keras seperti hard drive, CD, atau media penyimpanan lainnya. Sebagai gantinya, data disimpan pada *server* jarak jauh yang dikelola oleh pihak ketiga dan dapat diakses kapan saja selama tersedia koneksi internet.

Namun demikian, *cloud database* juga memiliki kelemahan, terutama dari aspek keamanan, karena data yang tersimpan di internet berpotensi menghadapi ancaman seperti peretasan. Oleh karena itu, diperlukan sistem keamanan yang baik untuk melindungi data pengguna. *Cloud database* sendiri memiliki keterkaitan erat dengan *Cloud Computing*, yaitu model komputasi yang menyediakan berbagai layanan teknologi informasi berbasis internet (Syaputra and Ratnasari, 2025.

2.5 *Client Server* Arsitektur

Arsitektur *Client Server* merupakan model komputasi terdistribusi di mana satu atau lebih komputer klien meminta sumber daya atau layanan dari komputer *server* melalui jaringan. Dalam model ini, klien bertugas menyediakan antarmuka bagi pengguna, sedangkan *server* bertugas menerima, memproses, dan mengembalikan respons atas permintaan yang dikirimkan. Komunikasi antara klien dan *server* difasilitasi oleh protokol standar seperti HTTP, FTP, dan SMTP yang memungkinkan pertukaran data antar perangkat (Nyabuto et al., 2024).

Arsitektur ini memiliki beberapa karakteristik utama, di antaranya kemampuan satu *server* untuk melayani banyak permintaan secara bersamaan serta skalabilitas baik secara vertikal maupun horizontal. Dari sisi keunggulan, arsitektur *Client Server* memungkinkan penyimpanan data secara terpusat yang dapat diakses dari berbagai lokasi secara *remote*, sehingga memudahkan pengelolaan dan pemeliharaan sumber daya. Selain itu, arsitektur ini mendukung *load balancing* melalui penambahan *server* redundan yang dapat mendistribusikan beban kerja sekaligus menyediakan

© 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

opsi pemulihan apabila salah satu *server* mengalami kegagalan (Nyabuto et al., 2024).

Berdasarkan jumlah tingkatan yang digunakan, arsitektur *Client Server* dapat diklasifikasikan menjadi beberapa jenis. Arsitektur two-tier hanya melibatkan dua lapisan yaitu klien dan *server*, cocok untuk aplikasi sederhana yang tidak membutuhkan banyak sumber daya pemrosesan. Arsitektur three-tier menambahkan satu lapisan tambahan berupa application tier yang memisahkan antarmuka pengguna, logika bisnis, dan penyimpanan data, sehingga meningkatkan kemudahan pemeliharaan, skalabilitas, dan keamanan sistem. Sementara itu, arsitektur n-tier memisahkan presentasi, pemrosesan, dan manajemen data secara logis maupun fisik, sehingga lebih skalabel dan cocok untuk aplikasi yang membutuhkan sumber daya besar serta kecepatan pemrosesan tinggi (Nyabuto et al., 2024).

2.6 *Rest API*

REST API merupakan suatu layanan berbasis *Website* yang digunakan untuk memungkinkan interaksi antar sistem melalui mekanisme komunikasi yang terstruktur. Layanan ini memanfaatkan metode HTTP, seperti GET, POST, PUT, dan DELETE, untuk mengelola pertukaran data antara klien dan *server*. Selain itu, *REST API* mendukung proses pertukaran data antar aplikasi *Website* dengan menggunakan format standar, seperti XML dan JSON, sehingga memudahkan integrasi antar sistem yang berbeda (Aziz, Subandri, and Armada, 2025).

2.7 *Firebase*

Firebase merupakan layanan *Backend-as-a-Service* (*BaaS*) yang menyediakan Real*time Database* untuk penyimpanan dan pengelolaan data berbasis objek. *Google Firebase* adalah platform pengembangan aplikasi berbasis *cloud* yang didukung oleh *Google* dan dirancang untuk membantu pengembang dalam membangun aplikasi Android, iOS, serta *Website*. *Firebase* menyediakan berbagai layanan terintegrasi seperti *realtime database*, *user authentication*, *cloud storage*, *cloud messaging*, *hosting*, dan layanan pendukung lainnya.

Firebase mempermudah proses pertukaran data antara aplikasi dan basis data melalui koneksi yang aman. Selain itu, *Firebase* menyediakan antarmuka *dashboard* yang sederhana sehingga mudah dikonfigurasi dan dikelola. Sistem ini

© 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

dirancang dengan tingkat keamanan yang tinggi serta membutuhkan pengaturan yang relatif minimal. Dengan *Firebase*, setiap unit atau klien dapat mengakses basis data secara langsung, sehingga mempercepat proses integrasi layanan *cloud* pada aplikasi *mobile* maupun *Website* (Saraf, Jadhao, Wanjari, Kolwate, & Patil, 2022).

2.8 *Firebase* Real*time Database*

Firebase Real*time Database* adalah layanan basis data NoSQL berbasis *cloud* yang memungkinkan penyimpanan data dan sinkronisasi antar pengguna secara *Realtime*. Data yang tersimpan akan diperbarui secara otomatis pada seluruh klien yang terhubung. Selain itu, layanan ini tetap dapat diakses meskipun aplikasi berada dalam kondisi *offline*, sehingga menjamin ketersediaan data secara berkelanjutan (Chougale, Yadav, & Gaikwad, 2021).

2.9 *Firebase Authentication*

Firebase Authentication merupakan layanan yang membantu pengembang dalam mengelola sistem autentikasi pengguna secara aman dan efisien. Layanan ini mendukung berbagai metode autentikasi, seperti email dan kata sandi, nomor telepon, serta integrasi dengan penyedia identitas pihak ketiga seperti *Google*, Facebook, GitHub, dan Twitter. Dengan adanya fitur ini, proses *login* dan pendaftaran pengguna dapat dilakukan dengan lebih mudah dan aman (Chougale, Yadav, & Gaikwad, 2021).

2.10 *Firebase Cloud Messaging*

Firebase Cloud Messaging (FCM) adalah layanan pengiriman pesan yang memungkinkan pengembang atau organisasi untuk mengirim notifikasi dan pesan data ke aplikasi berbasis Android, iOS, maupun *Website* tanpa biaya. Layanan ini digunakan untuk mendukung komunikasi aplikasi secara efisien dan *real-time* (Chougale, Yadav, & Gaikwad, 2021).

2.11 *Firebase Hosting*

Firebase Hosting menyediakan layanan *hosting* on*line* beserta domain yang dapat digunakan secara gratis untuk menghubungkan aplikasi ke internet. Layanan ini mempermudah pengembang dalam melakukan *deployment*, pemeliharaan, maupun perubahan pada aplikasi *Website*, bahkan setelah aplikasi dipublikasikan. *Firebase Hosting* juga mendukung pengiriman data menggunakan SSL melalui Content Delivery *Network* (CDN) sehingga koneksi aplikasi menjadi lebih cepat dan aman.

© 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

Selain itu, *Firebase Hosting* memiliki proses konfigurasi yang cukup mudah. Pengguna hanya perlu menginstal *Firebase Command Line Interface* (CLI), kemudian melakukan inisialisasi aplikasi pada direktori lokal sebelum proses *deployment* dilakukan. Seluruh tahapan tersebut dapat dijalankan hanya dengan beberapa perintah melalui *command line* hingga aplikasi berhasil di-host menggunakan *Firebase Hosting* (Nynäs, 2024).

2.12 *Firebase Cloud Functions*

Cloud Functions adalah layanan *serverless* yang memungkinkan eksekusi kode *backend* secara otomatis berdasarkan permintaan atau peristiwa tertentu. Pada *Firebase*, fungsi ini dapat dipicu oleh perubahan data, permintaan HTTPS, maupun jadwal tertentu. Dalam penelitian ini, *Cloud Functions* digunakan untuk memproses perubahan data pada *Firebase* Real*time Database* dan mengirimkan notifikasi kepada pengguna (Amanuel, 2022).

2.13 *Research* and *Development*

Penelitian dan pengembangan atau *Research and Development* (R&D) merupakan metode penelitian yang banyak diadopsi oleh dunia akademik untuk merancang dan menguji efektivitas suatu produk. Metode ini didefinisikan sebagai kegiatan yang bertujuan untuk mengembangkan serta menguji kemanfaatan dan efektivitas produk yang dikembangkan, baik berupa produk teknologi, material, organisasi, metode, maupun alat-alat. Selain itu, R&D juga dipahami sebagai studi sistematis dalam proses desain, pengembangan, dan evaluasi dengan tujuan membangun dasar yang empiris dalam menciptakan produk atau alat baru. Secara umum, metode R&D diawali dari analisis masalah, kemudian dilanjutkan dengan perancangan dan pengembangan produk, validasi oleh para ahli, uji coba, hingga revisi produk sehingga menghasilkan produk yang berkualitas tinggi dan teruji efektivitasnya secara ilmiah (Waruwu, 2024).

2.14 Keamanan *Firebase*

Keamanan merupakan salah satu aspek penting dalam pengembangan backend *server* berbasis *cloud* karena seluruh data aplikasi disimpan dan diakses melalui jaringan internet. Tanpa mekanisme keamanan yang memadai, data yang tersimpan pada *server* berpotensi diakses, dimodifikasi, atau disalahgunakan oleh pihak yang tidak memiliki hak akses. Oleh karena itu, *Firebase* menyediakan berbagai

© 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

mekanisme keamanan yang dirancang untuk melindungi data dan layanan backend, mulai dari autentikasi pengguna, pengaturan hak akses terhadap *database*, hingga pengelolaan izin bagi pengembang yang mengelola proyek *Firebase*. Dengan mekanisme tersebut, *Firebase* memungkinkan pengembang membangun aplikasi yang aman tanpa harus mengembangkan sistem keamanan dari awal.

Salah satu mekanisme keamanan yang disediakan *Firebase* adalah *Firebase Authentication*, yaitu layanan autentikasi yang digunakan untuk memverifikasi identitas pengguna sebelum mengakses layanan backend. Setelah pengguna berhasil melakukan autentikasi, *Firebase* akan menghasilkan identitas pengguna (*user identity*) yang digunakan sebagai dasar dalam proses pemberian hak akses terhadap data. Dengan demikian, setiap permintaan akses ke *database* dapat dikaitkan dengan identitas pengguna sehingga akses terhadap data dapat dikendalikan sesuai dengan kebijakan keamanan yang telah ditentukan (*Google*, 2026).

Untuk mengamankan data yang tersimpan pada *Firebase* Real*time Database*, *Firebase* menyediakan Real*time Database Security Rules*. *Security Rules* merupakan mekanisme kontrol akses yang menentukan siapa yang diizinkan untuk melakukan operasi baca (*read*) maupun tulis (*write*) pada *database*. Aturan keamanan tersebut dijalankan langsung pada *server Firebase* dan diterapkan secara otomatis pada setiap permintaan akses yang diterima. Secara bawaan (*default*), *Firebase* Real*time Database* tidak memberikan hak akses kepada pengguna mana pun sehingga *database* berada dalam kondisi terkunci (*locked mode*) hingga pengembang mengonfigurasi aturan keamanan atau menerapkan mekanisme autentikasi. Selain mengatur hak akses, *Security Rules* juga dapat digunakan untuk melakukan validasi data serta membatasi akses berdasarkan identitas pengguna maupun kondisi tertentu sehingga membantu menjaga kerahasiaan, integritas, dan konsistensi data yang tersimpan pada backend *server* (*Google*, 2026).

Selain keamanan pada tingkat pengguna aplikasi, *Firebase* juga menyediakan mekanisme *User*s and Permissions melalui layanan *Identity* and *Access Management* (IAM). Fitur ini digunakan untuk mengatur hak akses setiap anggota tim pengembang terhadap proyek *Firebase*. Setiap anggota dapat diberikan peran (*role*) yang berbeda, seperti *Owner*, *Editor*, *Firebase Admin*, atau *Viewer*, sehingga

© 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

hanya pengguna dengan hak akses tertentu yang dapat mengubah konfigurasi layanan *Firebase*, mengelola *database*, melakukan *deployment*, atau mengakses pengaturan proyek. Pendekatan ini menerapkan prinsip *least privilege*, yaitu setiap pengguna hanya diberikan izin sesuai dengan tugas dan tanggung jawabnya, sehingga risiko perubahan konfigurasi maupun penyalahgunaan akses dapat diminimalkan (*Google*, 2026).

2.15 Perbandingan Platform *Backend as a Service* (*BaaS*)

Backend-as-a-*Service* (*BaaS*) merupakan layanan *cloud* yang menyediakan berbagai komponen backend siap pakai, seperti autentikasi pengguna, basis data, penyimpanan, serta layanan lainnya yang dapat diakses melalui *Application Programming Interface* (API) maupun *Software Development Kit* (SDK). Berbagai platform *BaaS* memiliki karakteristik, kelebihan, dan keterbatasan yang berbeda, sehingga pemilihan platform perlu disesuaikan dengan kebutuhan sistem yang akan dikembangkan. Oleh karena itu, dilakukan perbandingan antara *Firebase* dan Supabase sebagai dua platform *BaaS* yang banyak digunakan dalam pengembangan aplikasi modern. Perbandingan ini bertujuan untuk mengidentifikasi karakteristik masing-masing platform berdasarkan layanan yang disediakan, sehingga dapat menjadi dasar dalam menentukan platform yang paling sesuai untuk pengembangan backend *server Smart Composter*. Perbandingan kedua platform tersebut disajikan pada Tabel 2.2.

Tabel 2.2 Perbandingan *Firebase* dan Supabase

Aspek	*Firebase*	*Supabase*
Penyedia	*Google*	*Supabase Inc.*
Database	*NoSQL(RTDB/Firestore)*	*PostgreSQL*
Authentication	Ya	Ya
Real-time	Ya	Ya
Cloud Functions	Ya	Edge *Functions*
Push Notification	*Firebase Cloud Messaging*	Tidak tersedia bawaan
Hosting	*Firebase Hosting*	Tidak tersedia bawaan
Open Source	Tidak	Ya
Vendor Lock-in	Tinggi	Rendah

Berdasarkan Tabel 2.2, *Firebase* dan Supabase sama-sama menyediakan layanan backend yang mendukung autentikasi pengguna, basis data, serta sinkronisasi data

© 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

secara *real-time*. Namun, *Firebase* menawarkan ekosistem layanan yang lebih lengkap, seperti *Firebase Cloud Messaging*, *Firebase Hosting*, dan *Cloud Functions* yang telah terintegrasi dalam satu platform. Sebaliknya, *Supabase* mengunggulkan penggunaan basis data relasional *PostgreSQL*, sifat *open source*, serta tingkat *vendor lock-in* yang lebih rendah. Berdasarkan karakteristik tersebut, penelitian ini memilih *Firebase* karena layanan yang disediakan telah memenuhi seluruh kebutuhan *backend Smart Composter*, meliputi autentikasi pengguna, penyimpanan dan sinkronisasi data secara *real-time*, pengiriman notifikasi, *hosting* halaman autentikasi, serta logika backend menggunakan *Cloud Functions*.

2.16 *Flutter*

Flutter merupakan *framework open-source* yang dikembangkan oleh *Google* dan digunakan untuk membangun antarmuka pengguna (*user interface*) pada aplikasi *mobile*, khususnya untuk platform Android dan iOS.

Flutter memiliki beberapa keunggulan, antara lain sebagai berikut:

1. Cross Platform

   Flutter mendukung pengembangan aplikasi lintas platform, sehingga satu aplikasi dapat dijalankan pada berbagai platform yang berbeda. Dengan menggunakan *Flutter*, pengembang dapat membangun aplikasi Android dan iOS secara bersamaan, bahkan juga untuk platform *Website* dan desktop. Pendekatan ini memungkinkan efisiensi waktu dan biaya karena tidak diperlukan penguasaan bahasa pemrograman native untuk masing-masing platform.

2. Fast *Development* (*Hot Reload*)

   Flutter menyediakan fitur *hot reload* yang berfungsi untuk mempercepat proses pengembangan aplikasi. Fitur ini memungkinkan perubahan kode ditampilkan secara langsung pada aplikasi dalam waktu singkat tanpa harus menjalankan ulang aplikasi secara keseluruhan, sehingga meningkatkan produktivitas pengembang.

3. Kemudahan Pengembangan Antarmuka

   Flutter dirancang untuk mempermudah pengembang dalam membangun tampilan antarmuka pengguna. Seluruh komponen antarmuka disusun menggunakan widget yang dapat dikustomisasi sesuai kebutuhan aplikasi. *Flutter* menyediakan dua jenis widget utama, yaitu *Material Design* untuk

© 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

Android dan *Cupertino* untuk iOS, sehingga tampilan aplikasi dapat menyesuaikan dengan karakteristik masing-masing platform.
(Setiawan, Ramadhan and Labellapansa, 2022)

2.17 *Dart*

Dart merupakan bahasa pemrograman yang dikembangkan oleh *Google* dan digunakan untuk membangun aplikasi sisi *frontend* yang dapat berjalan pada berbagai platform, termasuk *Website* dan perangkat *mobile*. Bahasa pemrograman ini dirancang untuk mendukung proses pengembangan aplikasi secara lebih sederhana dan efisien, dengan sintaks yang menyerupai JavaScript namun memiliki performa eksekusi yang lebih optimal. Salah satu fitur utama *Dart* adalah kemampuan *hot reload*, yang memungkinkan pengembang melihat perubahan kode secara langsung tanpa harus melakukan proses kompilasi ulang secara keseluruhan (Ahmad and Sutabri, 2024).

Selain itu, *Dart* mampu menghasilkan aplikasi dengan kinerja yang baik, terutama pada pengembangan aplikasi berskala besar. Melalui integrasinya dengan *framework Flutter*, *Dart* memungkinkan pembuatan aplikasi lintas platform menggunakan satu basis kode yang sama, sehingga dapat menghemat waktu serta biaya pengembangan. *Dart* juga mendukung fitur *static typing* yang membantu mendeteksi kesalahan sejak tahap kompilasi, sehingga meningkatkan kualitas dan keandalan kode dalam pengembangan aplikasi (Ahmad and Sutabri, 2024).

2.18 *Black box Testing*

Black box Testing merupakan metode pengujian perangkat lunak yang berfokus pada pengujian fungsi sistem berdasarkan input dan keluaran output yang dihasilkan tanpa memperhatikan proses internal maupun struktur kode program. Pengujian dilakukan untuk memastikan bahwa setiap fungsi pada sistem berjalan sesuai dengan kebutuhan dan spesifikasi yang telah ditetapkan (Budiman and Parhusip, 2026).

Pada metode ini, penguji memberikan berbagai skenario masukan ke dalam sistem kemudian mengamati apakah keluaran yang dihasilkan telah sesuai dengan hasil yang diharapkan. Dengan demikian, *Black box Testing* dapat digunakan untuk mengidentifikasi kesalahan fungsional yang mungkin terjadi pada sistem dari sudut pandang pengguna (Budiman and Parhusip, 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Keunggulan metode *Black box Testing* terletak pada kemudahan penerapannya karena tidak memerlukan pemahaman mengenai kode sumber aplikasi. Selain itu, metode ini efektif digunakan untuk memverifikasi fungsi-fungsi yang tersedia pada sistem serta memastikan bahwa seluruh fitur dapat berjalan dengan baik. Namun, metode ini memiliki keterbatasan karena tidak dapat mendeteksi kesalahan yang berkaitan dengan logika internal program maupun struktur data yang tidak terlihat dari sisi pengguna (Budiman and Parhusip, 2026).

Pada penelitian ini, *Black box Testing* digunakan untuk menguji seluruh fitur yang terdapat pada aplikasi *Smart Composter*, seperti *login*, registrasi, verifikasi email, reset *password*, pemilihan bin kompos, *monitoring* data sensor, pengendalian mesin pencacah, pengelolaan siklus kompos, penyimpanan riwayat *monitoring*, notifikasi, dan *logout*. Pengujian dilakukan dengan memberikan berbagai skenario masukan pada setiap fitur dan membandingkan hasil yang diperoleh dengan hasil yang diharapkan (Budiman and Parhusip, 2026).

2.19 SQA *Testing*

Software Quality Assurance (SQA) merupakan serangkaian aktivitas yang dilakukan untuk memastikan bahwa perangkat lunak yang dikembangkan memenuhi standar kualitas serta kebutuhan pengguna. SQA tidak hanya berfokus pada hasil akhir perangkat lunak, tetapi juga mencakup proses pengembangan secara keseluruhan, mulai dari tahap perencanaan, implementasi, pengujian, hingga pemeliharaan sistem.

Penerapan SQA bertujuan untuk menjamin bahwa perangkat lunak yang dihasilkan memiliki kualitas yang baik, dapat berfungsi sesuai kebutuhan, serta mampu memberikan pengalaman penggunaan yang optimal. Dalam penerapannya, SQA melibatkan berbagai aktivitas seperti pengujian perangkat lunak, evaluasi kualitas, dokumentasi, serta verifikasi terhadap kebutuhan sistem yang telah ditetapkan.

Penerapan SQA yang baik dapat membantu mengidentifikasi kesalahan atau kekurangan sistem sejak tahap awal pengembangan sehingga risiko terjadinya kegagalan sistem dapat diminimalkan. Selain itu, SQA juga berperan dalam memastikan bahwa perangkat lunak yang dikembangkan memiliki tingkat keandalan, kemudahan penggunaan, efisiensi, dan keamanan yang sesuai dengan standar kualitas perangkat lunak.

© 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

Pada penelitian ini, SQA digunakan untuk mengevaluasi kualitas aplikasi *Smart Composter* berdasarkan standar ISO/IEC 25010. Aspek yang dievaluasi meliputi *usability*. Hasil evaluasi digunakan untuk mengetahui tingkat kualitas aplikasi serta memastikan bahwa sistem yang dikembangkan telah memenuhi kebutuhan pengguna (Budiman and Parhusip, 2026).

2.20 QoS *Testing*

Quality of Service (QoS) merupakan mekanisme jaringan yang digunakan untuk mengevaluasi apakah suatu aplikasi atau layanan dapat beroperasi sesuai dengan kriteria yang telah ditentukan. QoS digunakan untuk mengukur kualitas layanan jaringan sehingga dapat diketahui tingkat performa komunikasi data yang berlangsung pada suatu sistem (Wahyudi, Mujiono & Fu'ad 2024).

Pengukuran QoS umumnya dilakukan menggunakan standar *Telecommunications* and Internet *Protocol* Harmonization Over *Networks* (TIPHON). Standar TIPHON dikembangkan oleh European *Telecommunications* Standards Institute (ETSI) untuk mengklasifikasikan kualitas layanan jaringan berdasarkan beberapa parameter pengukuran sehingga memudahkan proses analisis dan evaluasi performa jaringan (Wahyudi, Mujiono & Fu'ad 2024).

Dalam standar TIPHON, kualitas layanan jaringan dapat dievaluasi menggunakan beberapa parameter utama yaitu *delay*, *jitter*, *throughput* dan *packet loss*. Parameter-parameter tersebut digunakan untuk mengetahui kemampuan jaringan dalam mengirimkan data, tingkat kehilangan paket, waktu tunda pengiriman data, serta kestabilan transmisi data selama proses komunikasi berlangsung (Wahyudi, Mujiono & Fu'ad 2024).

2.20.1 *Delay*

Delay atau latency merupakan waktu yang dibutuhkan paket data untuk berpindah dari sumber ke tujuan selama proses komunikasi berlangsung. *Delay* diukur dalam satuan milisecond (ms). Semakin kecil nilai *delay* yang diperoleh, maka kualitas layanan jaringan semakin baik karena data dapat diterima lebih cepat oleh pengguna. Kategori kualitas jaringan berdasarkan *delay* ditunjukkan pada gambar 2.1 (Wahyudi, Mujiono & Fu'ad 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Kategori	Delay (ms)	Indeks
Sangat Bagus	< 150	4
Bagus	< 250	3
Sedang	< 450	2
Buruk	>= 450	1

Gambar 2.1 Kategori *Delay*

Berdasarkan gambar 2.1, kualitas *delay* dibagi menjadi empat kategori, yaitu sangat bagus, bagus, sedang, dan buruk. *Delay* menunjukkan waktu yang dibutuhkan paket data untuk mencapai tujuan selama proses transmisi. Semakin kecil nilai *delay* yang diperoleh, maka semakin baik kualitas layanan jaringan karena data dapat diterima lebih cepat oleh pengguna. Menurut standar TIPHON, nilai *delay* kurang dari 150 ms termasuk kategori sangat bagus, sedangkan nilai *delay* lebih dari atau sama dengan 450 ms termasuk kategori buruk (Wahyudi, Mujiono & Fu'ad 2024).

2.20.2 *Jitter*

Jitter merupakan perubahan atau fluktuasi waktu tunda (*delay*) yang terjadi antar paket data selama proses transmisi. Nilai *jitter* menunjukkan tingkat kestabilan waktu kedatangan paket, sehingga semakin kecil nilai *jitter* yang diperoleh, maka kualitas komunikasi jaringan semakin baik karena variasi *delay* yang terjadi antar paket semakin rendah. Kategori kualitas jaringan berdasarkan *jitter* ditunjukkan pada gambar 2.2 (Wahyudi, Mujiono & Fu'ad 2024).

Kategori	Jitter (ms)	Indeks
Sangat Bagus	0	4
Bagus	75	3
Sedang	125	2
Buruk	225	1

Gambar 2.2 Kategori *Jitter*

Berdasarkan kategori pada gambar 2.2, kualitas jaringan dinilai semakin baik apabila nilai *jitter* yang diperoleh semakin kecil. Nilai *jitter* yang rendah menunjukkan bahwa variasi waktu tunda antar paket data relatif kecil sehingga proses komunikasi berlangsung lebih stabil. Sebaliknya, nilai *jitter* yang tinggi mengindikasikan adanya ketidakstabilan dalam pengiriman paket data yang dapat memengaruhi kualitas layanan jaringan (Wahyudi, Mujiono & Fu'ad 2024).

2.20.3 *Throughput*

Throughput adalah rata-rata laju pengiriman data dalam jaringan yang dinyatakan dalam satuan bit per detik (bps). Pengukuran parameter ini penting karena nilai

© 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

throughput yang semakin tinggi menunjukkan kualitas jaringan yang semakin baik. Hal tersebut disebabkan oleh kemampuan jaringan untuk mentransmisikan data dalam jumlah yang lebih besar setiap detiknya (Wahyudi, Mujiono & Fu'ad 2024).

Kategori	Throughput (kbps)	Indeks
Sangat Bagus	>2100	4
Bagus	1200 - 2100	3
Sedang	700 - 1200	2
Buruk	338 - 700	1
Sangat Buruk	0 - 338	0

Gambar 2.3 Kategori *Throughput*

Berdasarkan gambar 2.3, nilai *throughput* diklasifikasikan ke dalam lima kategori. *Throughput* lebih dari 2100 kbps termasuk kategori Sangat Bagus dengan indeks 4. Nilai *throughput* pada rentang 1200–2100 kbps dikategorikan Bagus dengan indeks 3. Selanjutnya, *throughput* sebesar 700–1200 kbps berada pada kategori Sedang dengan indeks 2. Rentang *throughput* 338–700 kbps termasuk kategori Buruk dengan indeks 1, sedangkan *throughput* 0–338 kbps diklasifikasikan sebagai Sangat Buruk dengan indeks 0. Semakin tinggi nilai *throughput* yang diperoleh, semakin baik kemampuan jaringan dalam mentransmisikan data setiap detiknya sehingga kualitas layanan jaringan juga semakin baik (Wahyudi, Mujiono & Fu'ad 2024).

2.20.4 *Packet loss*

Packet loss merupakan jumlah paket data yang hilang selama proses pengiriman dari sumber menuju tujuan. Parameter ini dinyatakan dalam persentase (%). Semakin kecil nilai *packet loss* yang diperoleh, maka kualitas jaringan semakin baik karena menunjukkan bahwa sebagian besar paket berhasil diterima oleh tujuan tanpa mengalami kehilangan data. Kategori kualitas jaringan berdasarkan *packet loss* ditunjukkan pada gambar 2.3 (Wahyudi, Mujiono & Fu'ad 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Kategori	Packet Loss (%)	Indeks
Sangat Bagus	0	4
Bagus	3	3
Sedang	15	2
Buruk	25	1

Gambar 2.4 Kategori *Packet loss*

Berdasarkan gambar 2.4, kualitas *packet loss* dibagi menjadi empat kategori, yaitu sangat bagus, bagus, sedang, dan buruk. Semakin kecil nilai *packet loss* yang diperoleh, maka semakin baik kualitas layanan jaringan karena menunjukkan bahwa sebagian besar paket data berhasil diterima tanpa mengalami kehilangan selama proses transmisi. Menurut standar TIPHON, nilai *packet loss* sebesar 0% termasuk kategori sangat bagus, sedangkan nilai *packet loss* sebesar 25% termasuk kategori buruk (Wahyudi, Mujiono & Fu'ad 2024).

2.21 *Reliability Testing*

Reliability Testing merupakan proses pengujian yang dilakukan untuk mengevaluasi tingkat keandalan suatu sistem dalam menjalankan fungsi yang diharapkan secara konsisten tanpa mengalami kegagalan selama periode operasional tertentu. Pengujian ini bertujuan untuk mengetahui kemampuan sistem dalam mempertahankan ketersediaan layanan, meminimalkan kegagalan, serta memastikan layanan tetap dapat diakses oleh pengguna

Pada penelitian ini, *reliability Testing* digunakan untuk mengevaluasi keandalan *backend server Firebase* dalam menerima, menyimpan, dan menjaga keutuhan data sensor yang dikirim oleh perangkat *Smart Composter*. Pengujian dilakukan menggunakan metrik *availability*, *success rate*, *failure rate*, dan integritas data untuk mengetahui tingkat keandalan layanan *backend server* yang dikembangkan (Bayramova and Malikova, 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

BAB III
METODE PENELITIAN

3.1 Rancangan Penelitian

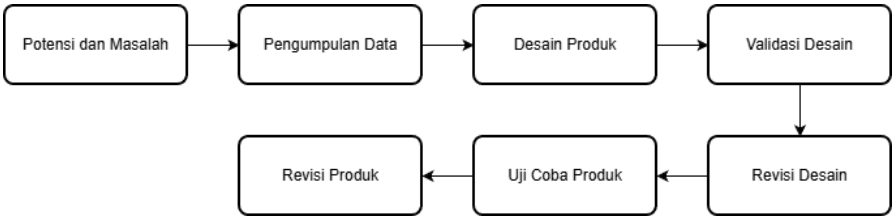
Penelitian ini menggunakan pendekatan kuantitatif dengan metode *Research* and *Development* (R&D). Pendekatan kuantitatif digunakan untuk mengevaluasi sistem yang dikembangkan melalui pengujian fungsionalitas, kualitas aplikasi, dan kinerja jaringan berdasarkan data numerik yang diperoleh dari *Black box Testing*, SQA *Testing*, serta QoS *Testing*.

Jenis penelitian yang digunakan adalah *Research* and *Development* (R&D), karena penelitian ini bertujuan untuk merancang, mengembangkan, dan menguji *Smart Composter* yang terdiri atas *backend* berbasis *Firebase* dan aplikasi *monitoring* berbasis *Flutter*.

Teknik pengumpulan data yang digunakan meliputi studi literatur, observasi kebutuhan sistem, serta pengujian sistem. Analisis data dilakukan dengan mengolah dan menganalisis hasil pengujian *Black box Testing*, SQA, dan QoS untuk mengetahui tingkat keberhasilan fungsi, kualitas aplikasi, serta kinerja komunikasi data pada sistem yang dikembangkan (Gusdevi, Hadhiwibowo, Agustina, Fatah, A. and Naseer, 2023).

3.2 Tahapan Penelitian

Tahapan penelitian merupakan rangkaian kegiatan yang dilakukan untuk mencapai tujuan penelitian yang telah ditetapkan. Pada penelitian ini, pengembangan *backend server Smart Composter* dilakukan menggunakan metode *Research* and *Development* (R&D). Metode ini digunakan karena memungkinkan proses pengembangan sistem dilakukan secara sistematis mulai dari identifikasi kebutuhan, perancangan, implementasi, hingga pengujian produk yang dikembangkan. Adapun tahapan penelitian yang dilakukan dapat dijelaskan sebagai berikut.



Gambar 3.1 Tahapan R&D mengacu pada Gusdevi et al. (2023)

© 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

Berdasarkan gambar 3.1 merupakan tahapan penelitian dalam pengembangan *backend server Smart Composter* menggunakan metode *Research* and *Development* (R&D) yang terdiri dari beberapa langkah berikut:

1. Potensi dan Masalah

   Tahap potensi dan masalah dilakukan untuk mengidentifikasi berbagai permasalahan yang terdapat pada proses pengelolaan dan *monitoring* kompos, khususnya yang berkaitan dengan pengelolaan data secara real-*time*. Berdasarkan hasil observasi dan studi terhadap sistem yang telah ada, diketahui bahwa proses penyimpanan data *monitoring*, sinkronisasi data antar perangkat, autentikasi pengguna, serta pengiriman notifikasi masih memiliki berbagai keterbatasan. Selain itu, beberapa penelitian terdahulu juga menunjukkan bahwa sistem *backend* yang digunakan belum mampu menyediakan pengelolaan data secara terpusat dan terintegrasi. Oleh karena itu, pada tahap ini dirumuskan kebutuhan pengembangan *backend server* berbasis *Firebase* yang mampu mendukung penyimpanan data, sinkronisasi data secara real-*time*, autentikasi pengguna, pengelolaan riwayat *monitoring*, serta pengiriman notifikasi pada *Smart Composter*.

2. Pengumpulan Data

   Tahap pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan sebagai dasar dalam proses perancangan *backend server*. Pengumpulan data dilakukan melalui studi literatur terhadap buku, jurnal ilmiah, dan dokumentasi resmi *Firebase* yang berkaitan dengan *backend server*, arsitektur *Client Server*, *Firebase Authentication*, *Firebase* Real*time Database*, *Firebase Cloud Messaging*, *Firebase Hosting*, dan *Firebase Cloud Functions*. Selain itu, dilakukan observasi terhadap kebutuhan sistem *Smart Composter* serta analisis kebutuhan fungsional dan nonfungsional agar layanan *backend* yang dikembangkan sesuai dengan kebutuhan pengguna.

3. Desain Produk

   Tahap desain produk dilakukan dengan merancang *backend server* yang akan digunakan pada sistem *Smart Composter* menggunakan arsitektur *Client Server* berbasis *Firebase*. Pada tahap ini disusun rancangan arsitektur sistem 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

menggambarkan hubungan antara perangkat IoT, *backend server*, serta aplikasi Android dan *Website* sebagai *client*. Selain itu, dilakukan perancangan struktur *Firebase* Real*time Database*, mekanisme autentikasi pengguna, alur pengiriman notifikasi menggunakan *Firebase Cloud Messaging*, implementasi *Firebase Cloud Functions*, serta layanan *Firebase Hosting* untuk aplikasi *Website*. Perancangan juga mencakup penyusunan use case diagram, *flowchart* sistem, serta rancangan antarmuka aplikasi sebagai acuan pada tahap implementasi.

4. Validasi Desain

   Tahap validasi desain dilakukan untuk memastikan bahwa rancangan *backend server* yang telah dibuat telah sesuai dengan kebutuhan sistem *Smart Composter*. Proses validasi dilakukan melalui peninjauan terhadap arsitektur *Client Server*, struktur basis data, mekanisme autentikasi pengguna, sinkronisasi data secara real-*time*, serta mekanisme pengiriman notifikasi. Selain itu, hasil perancangan dibandingkan dengan kebutuhan sistem yang telah diperoleh pada tahap analisis serta dikonsultasikan kepada dosen pembimbing untuk memperoleh masukan sebelum proses implementasi dilakukan. Tahap ini bertujuan untuk meminimalkan kesalahan desain sehingga *backend server* yang dibangun dapat memenuhi kebutuhan sistem.

5. Uji Coba Produk

   Setelah proses implementasi selesai dilakukan, *backend server* diuji untuk mengetahui apakah seluruh layanan telah berjalan sesuai dengan kebutuhan sistem. Pengujian dilakukan menggunakan *Black box Testing* untuk memastikan setiap fungsi *backend* dapat bekerja dengan benar berdasarkan skenario pengujian yang telah ditentukan. Selain itu, dilakukan pengujian SQA berdasarkan standar ISO/IEC 25010 pada aspek usability untuk mengevaluasi kualitas aplikasi dari sudut pandang pengguna. Kinerja komunikasi data juga dianalisis melalui pengujian QoS menggunakan parameter *delay*, *jitter*, *throughput*, dan *packet loss*. Selanjutnya dilakukan *reliability Testing* menggunakan parameter *availability*, *success rate*, *failure rate*, dan integritas data untuk mengetahui tingkat keandalan *backend server* dalam mengelola proses komunikasi dan penyimpanan data *Smart Composter*.

6. Revisi Produk

© 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

Setelah proses implementasi selesai dilakukan, *backend server* diuji untuk mengetahui apakah seluruh layanan telah berjalan sesuai dengan kebutuhan sistem. Pengujian dilakukan menggunakan *Black box Testing* untuk memastikan setiap fungsi *backend* dapat bekerja dengan benar berdasarkan skenario pengujian yang telah ditentukan. Selain itu, dilakukan pengujian *Software Quality Assurance* (SQA) berdasarkan standar ISO/IEC 25010 pada aspek usability untuk mengevaluasi kualitas aplikasi dari sudut pandang pengguna. Kinerja komunikasi data juga dianalisis melalui pengujian *Quality of Service* (QoS) menggunakan parameter *delay*, *jitter*, *throughput*, dan *packet loss*. Selanjutnya dilakukan *reliability Testing* menggunakan parameter *availability*, *success rate*, *failure rate*, dan integritas data untuk mengetahui tingkat keandalan *backend server* dalam mengelola proses komunikasi dan penyimpanan data *Smart Composter*.

3.3 Objek Penelitian

Objek penelitian dalam penelitian ini adalah *backend server Smart Composter* yang dibangun menggunakan layanan *Firebase*. *Backend server* tersebut berfungsi untuk mengelola autentikasi pengguna, menyimpan dan menyinkronkan data *monitoring* kompos secara *real-time*, mengelola riwayat pengomposan, serta mengirimkan notifikasi kepada pengguna melalui layanan *Firebase Cloud Messaging* (FCM).

Penelitian ini dilaksanakan di Bank Sampah Induk Rumah Harum yang berlokasi di Jalan Merdeka, Depok 2, Kota Depok, Jawa Barat. Lokasi ini dipilih sebagai tempat implementasi dan pengujian sistem karena mendukung kegiatan pengelolaan sampah organik dan pengomposan yang berkelanjutan. Pada penelitian ini, lokasi tersebut digunakan sebagai lingkungan implementasi *Smart Composter* yang terintegrasi dengan *backend server* berbasis *Firebase*. Hasil implementasi digunakan untuk mendukung proses perancangan, pembangunan, dan pengujian *backend server Smart Composter* menggunakan arsitektur *Client Server*.

Penelitian difokuskan pada perancangan, implementasi, dan pengujian layanan *backend* yang terdiri dari *Firebase Authentication*, *Firebase* Real*time Database*, *Firebase Cloud Functions*, *Firebase Cloud Messaging* (FCM), dan *Firebase Hosting*. Layanan-layanan tersebut diintegrasikan dalam arsitektur *Client Server*

untuk mendukung proses pengelolaan dan distribusi data *monitoring Smart Composter*.

Melalui *backend server* yang dikembangkan, data kondisi kompos yang meliputi suhu, kelembapan, level, pH, kadar gas, berat kompos, serta status kematangan kompos dapat disimpan, dikelola, dan didistribusikan kepada aplikasi *client* secara *real-time*. Selain itu, *backend server* juga mendukung penyimpanan riwayat *monitoring* dan pengiriman notifikasi otomatis ketika kondisi kompos telah mencapai status matang.

© 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

© 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 IV
HASIL DAN PEMBAHASAN

4.1 Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk mengidentifikasi kebutuhan yang diperlukan dalam pengembangan *backend server Smart Composter*. Tahap ini bertujuan untuk menentukan layanan, fungsi, serta kebutuhan data yang harus dipenuhi agar *backend server* mampu mengelola autentikasi pengguna, penyimpanan data *monitoring*, sinkronisasi data secara *real-time*, serta distribusi informasi kepada *client*. Hasil analisis kebutuhan menjadi dasar dalam proses perancangan dan implementasi *backend server* sehingga sistem yang dikembangkan dapat memenuhi tujuan penelitian.

4.1.1 Analisis Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang menjelaskan layanan dan fungsi yang harus disediakan oleh *backend server Smart Composter*. *Backend server* dikembangkan menggunakan layanan *Firebase* untuk mendukung autentikasi pengguna, pengelolaan data *monitoring*, sinkronisasi data secara *real-time*, penyimpanan riwayat pengomposan, serta pengiriman notifikasi kepada pengguna. Berdasarkan hasil analisis, *backend server* harus mampu menyediakan layanan registrasi akun dan *login* pengguna menggunakan *Firebase Authentication*. Selain itu, sistem juga harus mendukung proses verifikasi email dan reset *password* untuk membantu pengguna dalam mengelola akun secara aman.

Backend server harus mampu menerima, menyimpan, dan mengelola data *monitoring* yang berasal dari *Smart Composter*. Data yang dikelola meliputi suhu, kelembapan, level, pH, kadar gas, berat kompos, jarak permukaan kompos, kualitas kompos, serta status kematangan kompos. Data tersebut disimpan pada *Firebase* Real*time Database* dan disinkronkan secara *real-time* sehingga dapat diakses oleh aplikasi *client* tanpa memerlukan proses refresh secara manual.

Selain menyimpan data *monitoring* terkini, *backend server* juga harus menyediakan mekanisme penyimpanan riwayat data (history/log) untuk mendukung proses pencatatan dan pelacakan perubahan kondisi kompos dari waktu ke waktu. Riwayat

© 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

data tersebut digunakan sebagai referensi dalam proses *monitoring* dan evaluasi hasil pengomposan.

Untuk mendukung penyampaian informasi kepada pengguna, *backend server* harus mampu mengirimkan notifikasi ketika kondisi tertentu terpenuhi, seperti saat kompos mencapai status matang. Proses pengiriman notifikasi dilakukan menggunakan *Firebase Cloud Messaging* (FCM) yang dipicu secara otomatis oleh *Firebase Cloud Functions* berdasarkan perubahan data pada *Firebase* Real*time Database*. Dengan demikian, pengguna dapat memperoleh informasi kondisi kompos secara cepat tanpa harus melakukan pemantauan secara terus-menerus.

Selain layanan penyimpanan dan distribusi data, *backend server* juga harus mampu menyediakan akses data kepada aplikasi *client* berbasis Android maupun *Website* sehingga pengguna dapat memantau kondisi *Smart Composter* secara *real-time* melalui perangkat yang digunakan.

4.1.2 Analisis Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan kebutuhan yang mendukung operasional *backend server* agar dapat berjalan dengan baik. Kebutuhan ini mencakup perangkat lunak, layanan *cloud*, lingkungan pengembangan, serta karakteristik sistem yang diperlukan untuk mendukung pengelolaan data *monitoring Smart Composter*.

Pada penelitian ini, *backend server* dibangun menggunakan platform *Firebase* sebagai *Backend as a Service* (*BaaS*). Layanan *Firebase* yang digunakan meliputi *Firebase Authentication* untuk autentikasi pengguna, *Firebase* Real*time Database* untuk penyimpanan dan sinkronisasi data *monitoring* secara *real-time*, *Firebase Cloud Messaging* (FCM) untuk pengiriman notifikasi kepada pengguna, *Firebase Cloud Functions* sebagai pemicu otomatis pengiriman notifikasi berdasarkan perubahan data pada *database*, serta *Firebase Hosting* untuk menyediakan layanan berbasis *Website*.

Lingkungan pengembangan yang digunakan meliputi sistem operasi Windows, Visual Studio Code sebagai editor kode, *Firebase Command Line Interface* (CLI) untuk konfigurasi dan *deployment* layanan *Firebase*, *Node*.js untuk pengembangan *Firebase Cloud Functions*, serta Git sebagai sistem pengelolaan versi kode sumber.

© 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

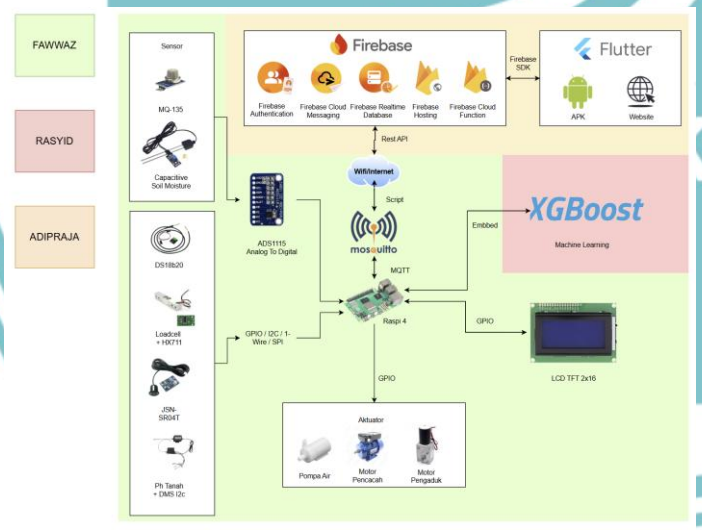
Backend server yang dikembangkan juga harus mampu menyediakan sinkronisasi data secara *real-time*, mendukung akses data dari beberapa *client* secara bersamaan, serta menjaga keamanan autentikasi dan data pengguna melalui mekanisme keamanan yang disediakan oleh *Firebase*. Selain itu, sistem diharapkan memiliki tingkat ketersediaan layanan yang baik sehingga proses *monitoring Smart Composter* dapat berlangsung secara berkelanjutan.

4.2 Perancangan Sistem

Perancangan sistem dilakukan berdasarkan hasil analisis kebutuhan yang telah ditetapkan sebelumnya. Pada penelitian ini, perancangan difokuskan pada *backend server Smart Composter* yang dibangun menggunakan layanan *Firebase*. Perancangan meliputi arsitektur *Client Server*, mekanisme autentikasi pengguna, struktur basis data, sinkronisasi data *monitoring* secara *real-time*, pengiriman notifikasi, serta integrasi antar layanan *Firebase*. Hasil perancangan ini menjadi acuan dalam proses implementasi *backend server*.

4.2.1 Perancangan Arsitektur Sistem

Berikut ini merupakan rancangan dari arsitektur sistem *Smart Composter*.



Gambar 4.1 Arsitektur Sistem

Berdasarkan gambar 4.1, sistem *Smart Composter* dirancang menggunakan arsitektur *Client Server* dengan *Firebase* sebagai backend *server* yang berperan sebagai pusat pengelolaan seluruh layanan sistem. Pada arsitektur ini, *Raspberry Pi* dan aplikasi *monitoring* berbasis *Flutter* berfungsi sebagai *client* yang saling berkomunikasi dengan backend *server* melalui jaringan internet. *Raspberry Pi*

© 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

bertugas mengirimkan data hasil pembacaan sensor ke backend *server*, sedangkan aplikasi *Flutter* mengirimkan permintaan data serta menerima pembaruan informasi secara real-*time* dari *server*. Seluruh proses autentikasi pengguna, penyimpanan data, sinkronisasi data, serta pengiriman notifikasi dikelola secara terpusat oleh backend *server* sehingga setiap *client* dapat mengakses data yang sama secara konsisten tanpa melakukan komunikasi secara langsung antarperangkat. Arsitektur *Client Server* dipilih karena mampu memisahkan proses pengelolaan data dari aplikasi *client*, sehingga sistem menjadi lebih mudah dikembangkan, dipelihara, serta mendukung akses dari berbagai platform secara bersamaan.

Backend *server* pada penelitian ini dibangun menggunakan beberapa layanan *Firebase* yang saling terintegrasi. *Firebase Authentication* digunakan untuk mengelola autentikasi pengguna, meliputi proses registrasi, *login*, verifikasi email, dan reset *password* sehingga hanya pengguna yang telah terverifikasi yang dapat mengakses sistem. *Firebase* Real*time Database* berfungsi sebagai media penyimpanan data *monitoring* kompos sekaligus melakukan sinkronisasi data secara real-*time* kepada seluruh *client* yang terhubung. Proses sinkronisasi tersebut memanfaatkan mekanisme komunikasi berbasis *Website*Socket sehingga setiap perubahan data pada *database* dapat langsung diterima oleh aplikasi tanpa memerlukan proses *polling* secara berkala. Selain itu, *Firebase Hosting* digunakan untuk menyediakan halaman verifikasi email dan reset *password* berbasis *Website* yang dibangun menggunakan HTML, CSS, dan JavaScript serta terintegrasi dengan *Firebase Authentication*.

Untuk mendukung mekanisme notifikasi otomatis, sistem memanfaatkan *Firebase Cloud Functions* dan *Firebase Cloud Messaging* (FCM). *Firebase Cloud Functions* berfungsi menjalankan logika backend secara otomatis ketika terjadi perubahan data pada *Firebase* Real*time Database*. Apabila hasil prediksi menunjukkan bahwa kompos telah mencapai status matang, *Cloud Functions* akan memproses kondisi tersebut dan meneruskan permintaan pengiriman notifikasi kepada *Firebase Cloud Messaging*. Selanjutnya, *Firebase Cloud Messaging* mengirimkan notifikasi ke perangkat Android maupun *Website* milik pengguna yang telah terdaftar sehingga informasi mengenai kondisi kompos dapat diterima secara cepat tanpa pengguna harus membuka aplikasi terlebih dahulu.

© 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

Aplikasi *monitoring* dikembangkan menggunakan *framework Flutter* sehingga dapat dijalankan pada platform Android maupun *Website* dengan basis kode yang sama. Aplikasi ini berfungsi sebagai *client* yang terhubung langsung dengan *Firebase* Real*time Database* untuk menampilkan data *monitoring* secara real-*time*, mengelola riwayat proses pengomposan, menampilkan informasi setiap siklus pengomposan, serta menyediakan fitur pengendalian aktuator sesuai hak akses pengguna. Melalui koneksi real-*time* yang disediakan *Firebase*, setiap perubahan data pada backend *server* dapat langsung ditampilkan pada *dashboard* aplikasi sehingga pengguna selalu memperoleh informasi kondisi *Smart Composter* secara terkini.

Secara keseluruhan, alur kerja sistem diawali dengan pembacaan data sensor oleh *Raspberry Pi* yang meliputi parameter suhu, kelembapan, pH, kadar amonia, berat kompos, serta level kompos. Data sensor tersebut kemudian diproses menggunakan model *Machine Learning* XGBoost untuk menentukan tingkat kematangan kompos. Selanjutnya, hasil pembacaan sensor beserta hasil prediksi dikirimkan ke *Firebase* Real*time Database* melalui jaringan internet dan disimpan pada backend *server*. Data yang telah tersimpan kemudian didistribusikan secara real-*time* kepada aplikasi *Flutter* sehingga dapat ditampilkan pada *dashboard monitoring* maupun riwayat pengomposan. Apabila status kematangan kompos berubah menjadi matang, *Firebase Cloud Functions* akan mendeteksi perubahan tersebut dan secara otomatis mengirimkan notifikasi melalui *Firebase Cloud Messaging* kepada pengguna yang telah terdaftar. Dengan memanfaatkan arsitektur *Client Server* berbasis *Firebase*, seluruh proses autentikasi pengguna, pengelolaan data, sinkronisasi informasi, pengiriman notifikasi, serta layanan *Website* untuk verifikasi email dan reset *password* dapat dilakukan secara terpusat, sehingga sistem *Smart Composter* memiliki kemampuan pengelolaan data yang lebih terintegrasi, andal, dan mudah diakses dari berbagai perangkat.

4.2.2 Perancangan *Database*

Pada penelitian ini, *Firebase* Real*time Database* digunakan sebagai basis data utama untuk menyimpan data *monitoring* serta informasi yang diperlukan oleh sistem *Smart Composter*. Struktur basis data dirancang menggunakan dua *node* utama, yaitu *Current* dan *Log*, yang memiliki fungsi berbeda sesuai dengan

© 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

kebutuhan sistem *monitoring*. *Node Current* digunakan untuk menyimpan data *monitoring* terkini yang diperoleh dari sensor. Setiap kali *Raspberry Pi* mengirimkan data baru, data pada *node* ini akan diperbarui sehingga data sebelumnya akan digantikan oleh data terbaru. Dengan demikian, aplikasi dapat menampilkan kondisi *Smart Composter* secara *real-time* dengan proses pengambilan data yang lebih cepat dan efisien.

Selain *node Current*, sistem juga menggunakan *node Log* untuk menyimpan seluruh riwayat data *monitoring* selama proses pengomposan berlangsung. Berbeda dengan *node Current*, data pada *node Log* tidak ditimpa oleh data baru, melainkan setiap data yang diterima akan disimpan sebagai entri baru sehingga seluruh riwayat *monitoring* tetap tersimpan. Struktur ini memungkinkan aplikasi menampilkan riwayat pengomposan, melakukan analisis terhadap proses pengomposan, serta menyediakan data historis yang dapat diunduh oleh pengguna dalam bentuk laporan.

Dalam perancangan basis data, diterapkan pula pengaturan hak akses menggunakan *Firebase* Real*time Database Security Rules* untuk membatasi akses pengguna terhadap data sesuai dengan tingkat kewenangannya. Pengguna yang telah terautentikasi dapat mengakses data *monitoring*, sedangkan hak akses tulis pada data aktuator dibatasi hanya untuk admin dan petugas bank sampah. Hak akses admin ditentukan berdasarkan *role* `admin`, sedangkan petugas bank sampah memperoleh akses pengendalian aktuator melalui atribut `canAccessRemote` yang diberikan oleh admin. *User* umum tidak memiliki hak akses untuk mengubah data aktuator sehingga hanya dapat menggunakan fitur *monitoring* dan informasi yang tersedia pada sistem. Pengaturan hak akses tersebut dirancang untuk mencegah pengguna yang tidak memiliki kewenangan melakukan pengendalian aktuator melalui fitur *remote*.

Dengan pemisahan fungsi antara *node Current* dan *Log* serta penerapan pengaturan hak akses pada basis data, *Firebase* Real*time Database* mampu mendukung kebutuhan *monitoring* secara *real-time*, menyediakan penyimpanan data historis secara berkelanjutan, serta membatasi akses terhadap fungsi pengendalian aktuator berdasarkan kewenangan masing-masing pengguna.

© 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

4.2.3 Perancangan Antarmuka Android

Perancangan antarmuka aplikasi Android dilakukan untuk menghasilkan tampilan yang mudah digunakan, informatif, dan mampu mendukung interaksi pengguna dengan sistem *Smart Composter* secara efektif. Antarmuka dirancang dengan memperhatikan kemudahan navigasi, konsistensi tampilan, serta penyajian informasi *monitoring* secara jelas sehingga pengguna dapat mengakses seluruh fitur aplikasi dengan mudah. Selain itu, setiap halaman dirancang sesuai dengan kebutuhan fungsional sistem yang telah didefinisikan pada tahap analisis kebutuhan.

Antarmuka aplikasi terdiri atas beberapa halaman utama yang mendukung proses autentikasi pengguna, *monitoring* kondisi kompos, serta pengelolaan data hasil pengomposan. Halaman autentikasi meliputi *login*, register, *forgot password*, dan reset *password* yang digunakan untuk mengelola proses masuk ke dalam sistem. Setelah berhasil melakukan autentikasi, pengguna akan diarahkan ke halaman pemilihan bin kompos untuk memilih unit *Smart Composter* yang akan dimonitor. Selanjutnya, aplikasi menyediakan halaman *dashboard monitoring* yang digunakan untuk menampilkan data sensor secara *real-time*, hasil prediksi tingkat kematangan kompos, status aktuator, serta fitur pengendalian perangkat sesuai hak akses pengguna. Selain itu, tersedia halaman riwayat pengomposan yang berfungsi menampilkan data historis setiap siklus pengomposan beserta informasi pendukung lainnya, seperti waktu proses, hasil *monitoring*, dan kualitas kompos. Perancangan setiap halaman tersebut bertujuan untuk memberikan kemudahan kepada pengguna dalam melakukan *monitoring* dan pengelolaan proses pengomposan melalui aplikasi Android.

4.2.4 Perancangan Antarmuka *Websitesite*

Perancangan antarmuka *Website* pada aplikasi *Smart Composter* dilakukan dengan menyesuaikan tampilan antarmuka aplikasi agar dapat ditampilkan dengan baik pada layar yang memiliki ukuran lebih besar dibandingkan perangkat *mobile*. Penyesuaian yang dilakukan meliputi pengaturan ukuran komponen, tata letak (layout), serta pemanfaatan ruang tampilan yang lebih luas sehingga informasi dapat ditampilkan dengan lebih optimal tanpa mengubah fungsi maupun alur penggunaan aplikasi. Dengan demikian, antarmuka *Website* tetap memberikan

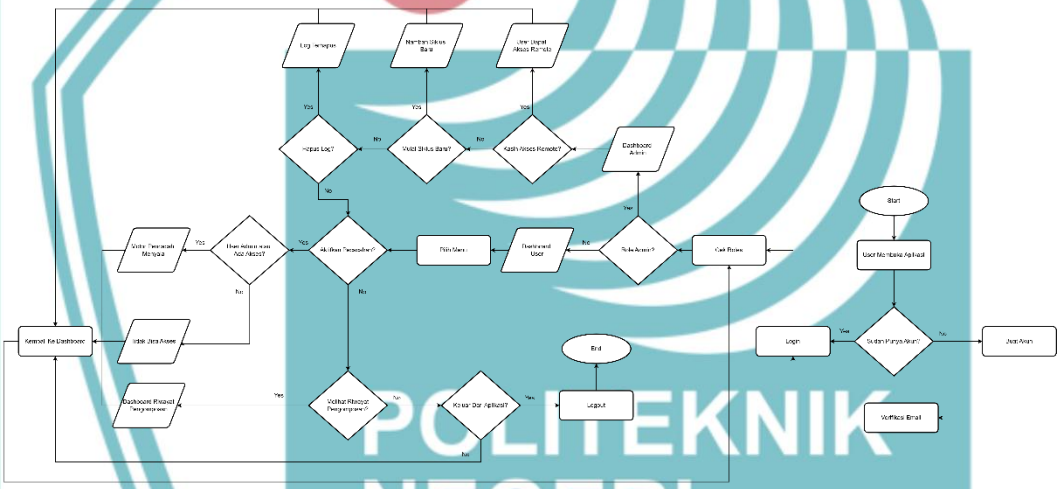
© Hak Cipta milik Politeknik Negeri Jakarta

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

pengalaman penggunaan yang konsisten dengan aplikasi *mobile* namun lebih sesuai untuk diakses melalui browser pada perangkat desktop.

4.2.5 *Flowchart* Aplikasi

Flowchart aplikasi digunakan untuk menggambarkan alur kerja *Smart Composter* dari awal hingga akhir proses penggunaan aplikasi. Diagram ini menunjukkan tahapan yang dilakukan pengguna mulai dari membuka aplikasi, melakukan autentikasi akun, memilih bin kompos, memantau kondisi kompos, mengendalikan mesin pencacah, melihat riwayat pengomposan, hingga keluar dari aplikasi. Dengan adanya *flowchart*, alur proses yang terjadi pada sistem dapat dipahami dengan lebih jelas sehingga memudahkan dalam memahami cara kerja aplikasi yang dikembangkan.



Gambar 4.2 *Flowchart* Aplikasi

Berdasarkan *flowchart* pada gambar 4.2, proses diawali ketika pengguna membuka aplikasi. Jika pengguna belum memiliki akun, pengguna dapat melakukan registrasi dan verifikasi email terlebih dahulu. Selanjutnya pengguna melakukan *login*, kemudian sistem melakukan pengecekan peran (*role*) pengguna. Apabila pengguna memiliki peran sebagai administrator, sistem akan menampilkan *Dashboard Admin* yang menyediakan fitur pengelolaan pengguna, pembuatan siklus baru, dan penghapusan log. Sebaliknya, apabila pengguna memiliki peran sebagai pengguna (*user*), sistem akan menampilkan *Dashboard User*. Dari *Dashboard User*, pengguna dapat memilih menu untuk memantau kondisi kompos secara *real-time*, melihat riwayat pengomposan, mengaktifkan mesin pencacah apabila memiliki hak akses, atau melakukan *logout* untuk mengakhiri penggunaan aplikasi. Setelah setiap

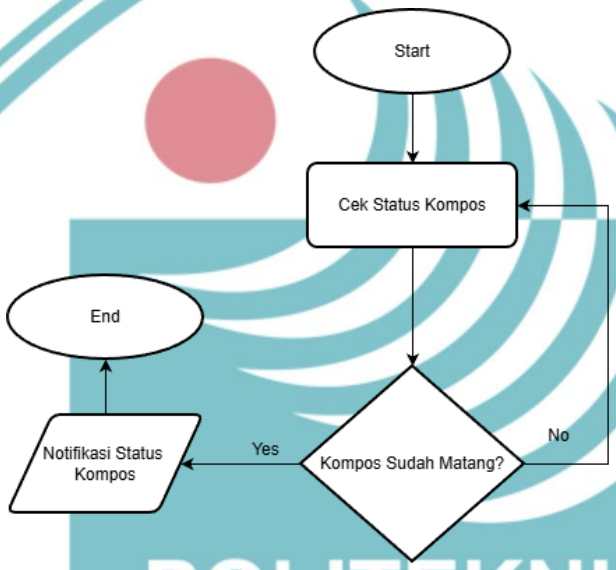
© Hak Cipta milik Politeknik Negeri Jakarta

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

proses selesai dijalankan, pengguna dapat kembali ke *dashboard* untuk melakukan aktivitas lainnya.

4.2.6 *Flowchart* Notifikasi

Flowchart notifikasi kematangan kompos digunakan untuk menggambarkan alur proses pengiriman notifikasi kepada pengguna ketika kompos telah mencapai kondisi matang. Diagram ini menunjukkan tahapan yang dilakukan sistem mulai dari pengecekan status kompos hingga pengiriman notifikasi sebagai informasi kepada pengguna mengenai kondisi kompos yang telah siap digunakan.



Gambar 4.3 *Flowchart* Notifikasi

Berdasarkan *flowchart* pada gambar 4.3, proses diawali dengan sistem melakukan pengecekan status kompos yang tersimpan pada *database*. Selanjutnya sistem melakukan pemeriksaan apakah kompos telah mencapai kondisi matang atau belum. Jika kompos belum matang, sistem akan kembali melakukan pengecekan status kompos secara berkala. Namun, apabila kompos telah matang, sistem akan mengirimkan notifikasi status kompos kepada pengguna melalui aplikasi. Setelah notifikasi berhasil dikirim, proses berakhir dan pengguna dapat menerima informasi bahwa kompos telah siap digunakan.

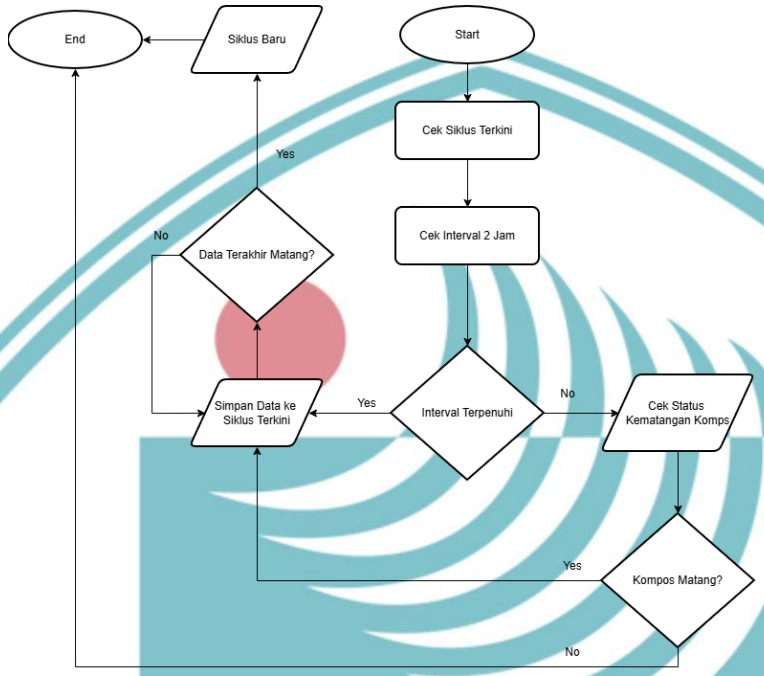
4.2.7 *Flowchart* Siklus Baru

Flowchart manajemen siklus pengomposan digunakan untuk menggambarkan proses pengelolaan data riwayat pengomposan yang tersimpan pada sistem. Proses ini mencakup pengecekan siklus aktif, penyimpanan data sensor secara berkala setiap dua jam, pemeriksaan status kematangan kompos, hingga pembuatan siklus

© 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

pengomposan baru secara otomatis ketika kompos telah mencapai kondisi matang. *Flowchart* ini bertujuan untuk memberikan gambaran mengenai mekanisme pencatatan data riwayat pengomposan berdasarkan siklus yang sedang berlangsung.



Gambar 4.4 *Flowchart* Siklus Pengomposan

Berdasarkan *flowchart* pada gambar 4.4, proses diawali dengan sistem melakukan pengecekan siklus pengomposan terbaru yang sedang aktif. Selanjutnya sistem memeriksa apakah *interval* penyimpanan data selama dua jam telah terpenuhi. Apabila *interval* terpenuhi, data sensor akan disimpan ke dalam siklus pengomposan yang sedang berjalan. Setelah data disimpan, sistem melakukan pemeriksaan terhadap status kematangan kompos. Jika kompos belum matang, data berikutnya akan tetap disimpan pada siklus yang sama hingga proses pengomposan selesai.

Apabila *interval* dua jam belum terpenuhi, sistem tetap melakukan pemeriksaan status kematangan kompos. Jika kompos telah mencapai kondisi matang, sistem akan menyimpan data terakhir ke siklus yang sedang aktif agar kondisi akhir pengomposan tetap tercatat pada riwayat. Setelah itu sistem mengirimkan notifikasi kepada pengguna mengenai status kematangan kompos dan secara otomatis membuat siklus pengomposan baru dengan ID siklus berikutnya. Dengan

© 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

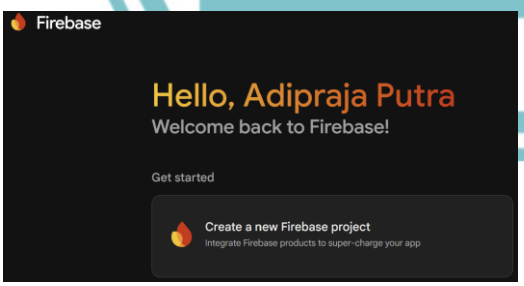
mekanisme ini, data riwayat setiap proses pengomposan dapat tersimpan secara terpisah berdasarkan siklus pengomposan yang berlangsung.

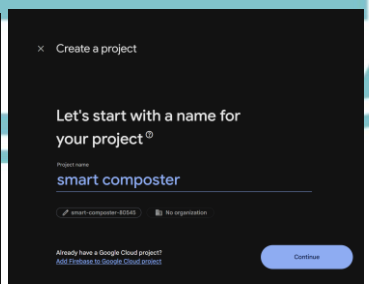
4.3 Implementasi Sistem

Implementasi sistem merupakan tahap penerapan hasil perancangan *backend server* yang telah dilakukan pada tahap sebelumnya. Pada penelitian ini, implementasi difokuskan pada pembangunan *backend server Smart Composter* menggunakan layanan *Firebase* yang terdiri dari *Firebase Authentication*, *Firebase* Real*time Database*, *Firebase Cloud Functions*, *Firebase Cloud Messaging* (FCM), dan *Firebase Hosting*. Implementasi dilakukan untuk mendukung proses autentikasi pengguna, penyimpanan dan sinkronisasi data *monitoring* secara *real-time*, pengelolaan riwayat pengomposan, pengiriman notifikasi, serta penyediaan layanan pendukung yang dibutuhkan oleh aplikasi *monitoring*. Hasil implementasi ini menjadi dasar dalam proses pengujian untuk memastikan seluruh layanan *backend* dapat berfungsi sesuai dengan kebutuhan sistem yang telah ditetapkan.

4.3.1 Implementasi *Firebase*

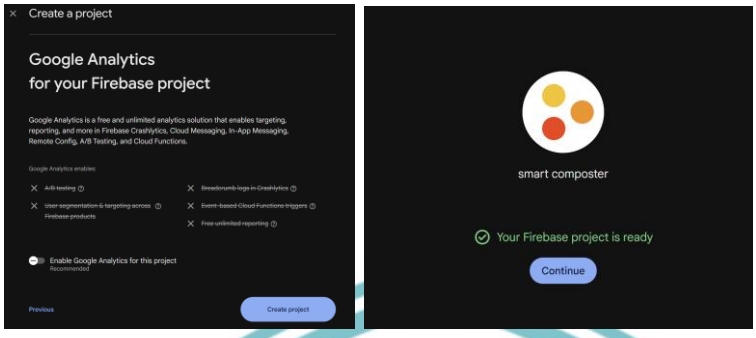
Implementasi *Firebase* diawali dengan pembuatan *project* pada platform *Firebase Console* sebagai pusat layanan *backend* yang digunakan dalam *Smart Composter*. *Firebase* dipilih karena menyediakan berbagai layanan yang terintegrasi, seperti Real*time Database*, *Cloud Functions*, *Authentication*, dan *Firebase Cloud Messaging* yang mendukung kebutuhan penyimpanan data, sinkronisasi data secara *real-time*, serta pengiriman notifikasi kepada pengguna. Pada tahap ini dilakukan konfigurasi awal *project Firebase* yang akan digunakan sebagai lingkungan *backend*.





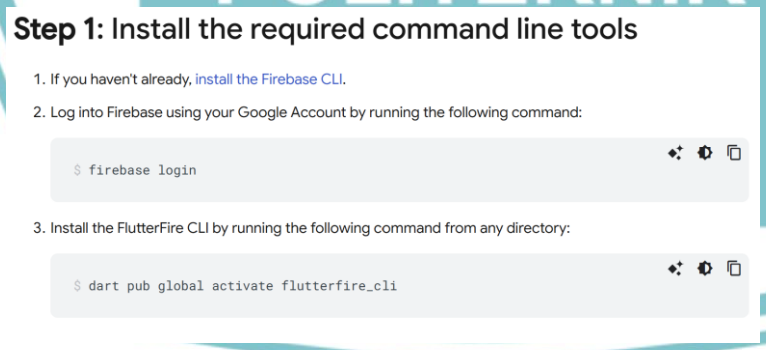
© Hak Cipta milik Politeknik Negeri Jakarta

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 4.5 Implementasi *Firebase*

Berdasarkan gambar 4.5, implementasi *Firebase* diawali dengan pembuatan *project* baru melalui *Firebase Console*. Pada tahap ini pengguna memilih menu Create a new *Firebase project* untuk membuat lingkungan *backend* yang akan digunakan oleh *Smart Composter*. Selanjutnya, pengguna memasukkan nama *project* yang akan digunakan, kemudian melanjutkan proses konfigurasi dengan memilih opsi yang tersedia pada *Firebase Console*. Pada penelitian ini, fitur *Google* Analytics tidak diaktifkan karena tidak digunakan dalam proses pengembangan sistem. Setelah seluruh konfigurasi selesai dilakukan, *Firebase* akan membuat *project* baru dan menampilkan status bahwa *project* telah berhasil dibuat. *Project Firebase* yang telah dibuat kemudian digunakan sebagai pusat layanan *backend* yang mengintegrasikan *Firebase* Real*time Database*, *Firebase Cloud Functions*, dan *Firebase Cloud Messaging* dalam *Smart Composter*.



Gambar 4.6 Konfigurasi *Firebase* pada lingkungan pengembang

Berdasarkan gambar 4.6, setelah *project Firebase* berhasil dibuat, tahap selanjutnya adalah melakukan instalasi *command line* tools yang diperlukan untuk mengelola layanan *Firebase* melalui terminal. Pada penelitian ini digunakan *Firebase Command Line Interface* (CLI) yang berfungsi sebagai alat bantu untuk menghubungkan lingkungan pengembangan dengan *project Firebase* yang telah dibuat. *Firebase* CLI memungkinkan pengguna melakukan berbagai konfigurasi

© Hak Cipta milik Politeknik Negeri Jakarta

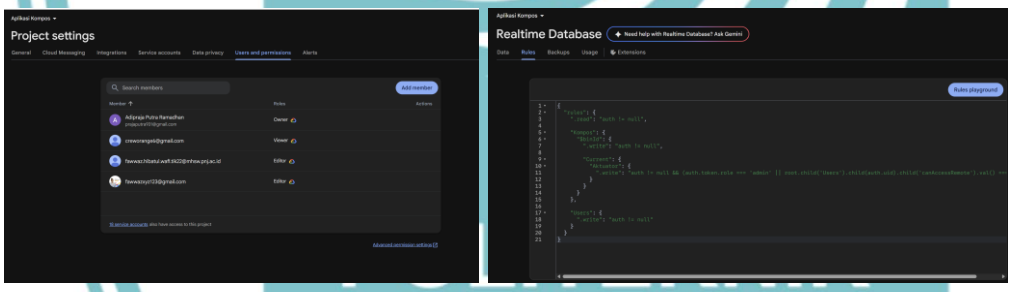
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan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

layanan *Firebase*, seperti *Firebase Hosting*, *Firebase Cloud Functions*, serta proses *deployment* aplikasi dan layanan *backend* ke *server Firebase*.

Proses konfigurasi diawali dengan melakukan autentikasi menggunakan akun *Google* yang terhubung dengan *project Firebase* melalui perintah *Firebase login*. Setelah proses autentikasi berhasil dilakukan, pengguna dapat mengakses dan mengelola seluruh layanan *Firebase* yang tersedia pada *project* melalui terminal. Selain *Firebase* CLI, dilakukan pula instalasi *Flutter*Fire CLI sebagai salah satu utilitas tambahan yang digunakan untuk mendukung integrasi layanan *Firebase* pada tahap pengembangan sistem. Dengan terpasangnya kedua *command line* tools tersebut, proses konfigurasi dan pengelolaan layanan *Firebase* dapat dilakukan secara lebih mudah dan terpusat melalui lingkungan pengembangan yang digunakan.

4.3.2 Implementasi Keamanan *Firebase*

Berikut ini konfigurasi keamanan *Firebase*.



Gambar 4.7 Konfigurasi Keamanan *Firebase*

Berdasarkan gambar 4.7, menunjukkan konfigurasi *Users and Permissions* pada proyek *Firebase Smart Composter* yang memanfaatkan layanan *Google Cloud Identity* and *Access Management* (IAM). Fitur ini digunakan untuk mengatur hak akses setiap anggota tim pengembang terhadap proyek *Firebase* berdasarkan peran (*role*) yang diberikan. Pada penelitian ini, akun pengembang utama diberikan peran sebagai *Owner*, sehingga memiliki hak akses penuh untuk mengelola seluruh layanan *Firebase*, seperti *Firebase Authentication*, *Firebase* Real*time Database*, *Firebase Cloud Messaging*, *Firebase Hosting*, dan *Cloud Functions*. Selain itu, akun anggota tim lainnya diberikan peran sebagai *Editor* dan *Viewer* sesuai dengan kebutuhan pengembangan. Pengguna dengan peran *Editor* diberikan izin untuk melakukan konfigurasi dan pengembangan layanan tertentu, sedangkan pengguna dengan peran *Viewer* hanya dapat melihat konfigurasi proyek tanpa memiliki hak

© 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

untuk melakukan perubahan. Pembagian hak akses tersebut bertujuan untuk menerapkan prinsip *least privilege*, yaitu setiap anggota hanya diberikan hak akses sesuai dengan tugas dan tanggung jawabnya. Dengan demikian, risiko perubahan konfigurasi backend *server* oleh pengguna yang tidak berwenang dapat diminimalkan sehingga keamanan dan pengelolaan proyek *Firebase* menjadi lebih terjamin.

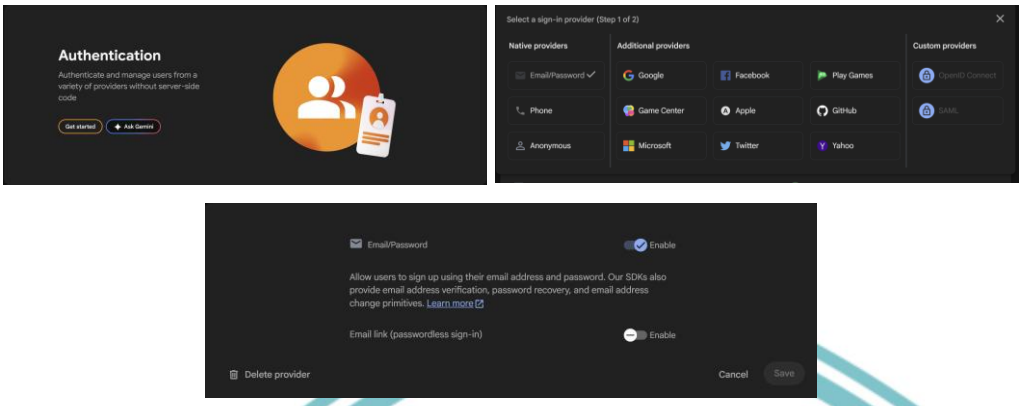
Selain menerapkan pengaturan hak akses, penelitian ini juga mengimplementasikan mekanisme keamanan pada tingkat *database* menggunakan *Firebase* Real*time Database Security Rules*. *Security Rules* merupakan aturan keamanan yang dijalankan langsung pada server *Firebase* untuk mengendalikan setiap permintaan akses terhadap *database*. Pada penelitian ini, akses terhadap data *Smart Composter* dibatasi hanya untuk pengguna yang telah berhasil melakukan autentikasi menggunakan *Firebase Authentication*. Selain itu, pembatasan hak akses secara khusus diterapkan pada data aktuator yang digunakan dalam fitur *remote*. Hak akses tulis (*write*) pada data aktuator hanya diberikan kepada admin yang memiliki *role* admin dan petugas bank sampah yang telah memperoleh hak akses *remote* melalui atribut can*AccessRemote*. Secara *default*, *user* umum tidak memiliki hak akses untuk mengubah data aktuator sehingga tidak dapat mengendalikan perangkat melalui fitur *remote*. Admin memiliki kewenangan untuk mengelola hak akses pengguna dengan mengubah nilai can*AccessRemote*, sehingga *user* umum yang telah diberikan hak akses dapat berperan sebagai petugas bank sampah dan menggunakan fitur *remote*. Pada data pengguna, setiap pengguna hanya dapat membaca data akun miliknya sendiri, sedangkan admin dapat membaca data pengguna untuk keperluan pengelolaan hak akses. Penerapan *Security Rules* tersebut bertujuan untuk mencegah akses dan pengendalian aktuator oleh pengguna yang tidak memiliki kewenangan. Dengan kombinasi *Firebase Authentication*, *role* pengguna, dan *Firebase* Real*time Database Security Rules*, *backend server* mampu menerapkan mekanisme autentikasi dan kontrol akses berdasarkan tingkat kewenangan pengguna sehingga keamanan data serta pengendalian aktuator *Smart Composter* dapat terjaga.

4.3.3 Implementasi *Firebase Authentication*

Berikut ini tampilan pengaturan dari *Firebase Authentication*

© Hak Cipta milik Politeknik Negeri Jakarta

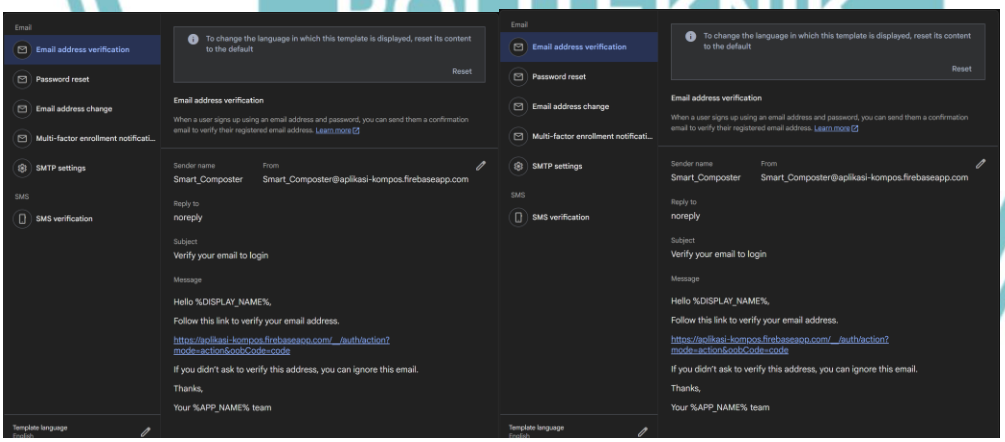
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan 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 4.8 Settingan *Firebase Authentication*

Berdasarkan gambar 4.8, settingan *Firebase Authentication* digunakan untuk mengelola proses autentikasi pengguna pada aplikasi *Smart Composter*. Implementasi fitur ini memungkinkan pengguna melakukan registrasi akun menggunakan alamat email dan kata sandi. Setelah akun berhasil dibuat, pengguna dapat melakukan *login* untuk mengakses fitur-fitur yang tersedia pada aplikasi.

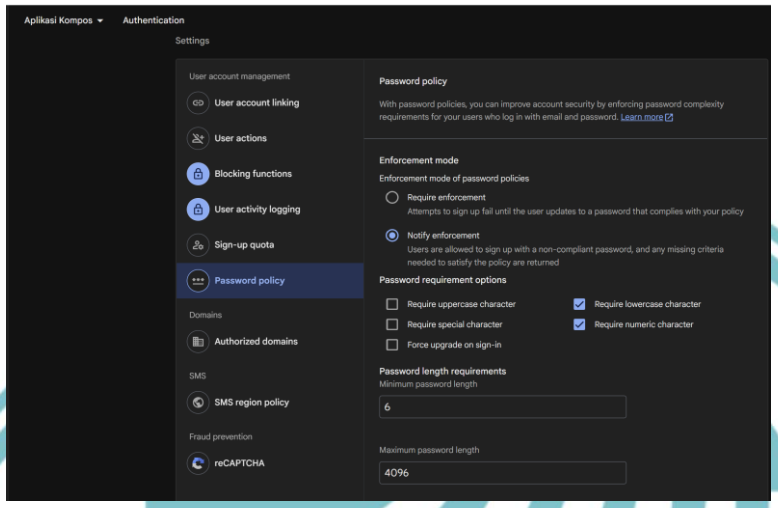
Selain proses *login* dan registrasi, sistem juga menyediakan fitur perubahan *password* yang bertujuan untuk meningkatkan keamanan akun pengguna. Seluruh data autentikasi dikelola oleh *Firebase Authentication* sehingga proses validasi pengguna dapat dilakukan secara aman.



Gambar 4.9 Penyesuaian *Template Authentication*

Berdasarkan gambar 4.9, melakukan penyesuaian pada *template* email bawaan yang disediakan oleh *Firebase*. Terdapat tiga *template* yang dimodifikasi, yaitu email verification dan reset *password* (pengaturan ulang kata sandi). Penyesuaian yang dilakukan pada ketiga *template* tersebut adalah mengubah nama pengirim menjadi nama pengembang, sehingga email yang diterima oleh pengguna tidak lagi

menampilkan nama *default* dari *Firebase*, melainkan menampilkan identitas yang lebih personal dan relevan dengan aplikasi yang dikembangkan. Hal ini bertujuan agar komunikasi melalui email terasa lebih pro*f*esional.



Gambar 4.10 Kebijakan *Password*

Berdasarkan gambar 4.10, untuk meningkatkan keamanan akun pengguna, sistem menerapkan kebijakan *password* pada *Firebase Authentication* dengan panjang minimum 6 karakter serta mewajibkan penggunaan huruf kecil dan angka. Kebijakan ini bertujuan untuk mengurangi risiko penggunaan *password* yang lemah dan mudah ditebak, sehingga dapat membantu melindungi akun pengguna dari akses yang tidak sah. Selain itu, pengaturan *notify enforcement* digunakan agar pengguna tetap dapat melakukan registrasi sambil memperoleh informasi mengenai kriteria *password* yang disarankan untuk meningkatkan keamanan akun.

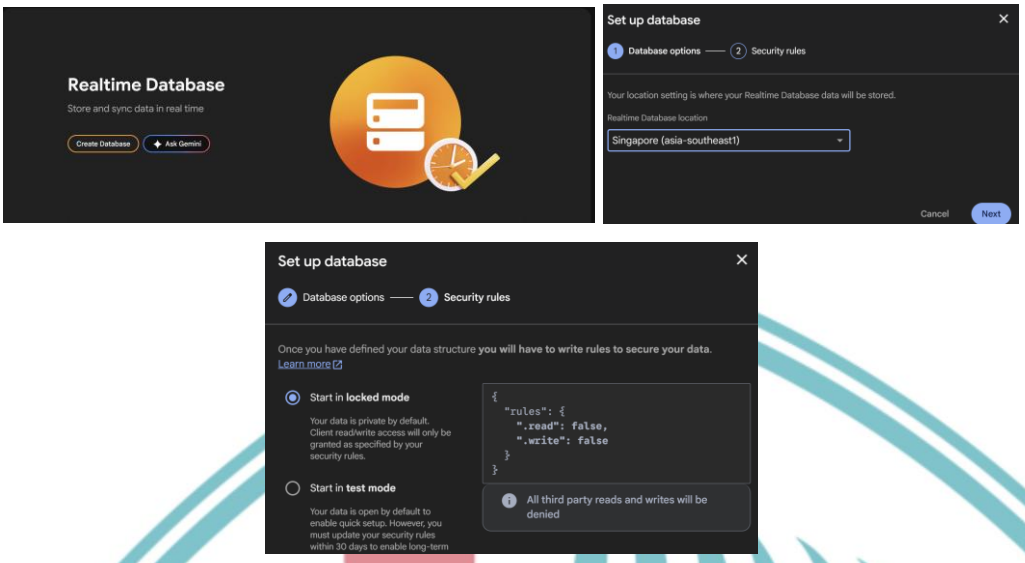
4.3.4 Implementasi *Firebase* Real*time Database*

Firebase Real*time Database* diimplementasikan sebagai media penyimpanan dan sinkronisasi data antara perangkat sensor, aplikasi *mobile*, dan *dashboard Website*. Berikut ini konfigurasi dari Real*time Database*.

© 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

© 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 4.11 Konfigurasi *Firebase* Real*time Database*

Berdasarkan gambar 4.11, konfigurasi *Firebase* Real*time Database* dilakukan dengan memilih lokasi *server* di wilayah Singapore (asia-southeast1) sebagai tempat penyimpanan data. Pemilihan lokasi ini bertujuan untuk mengurangi latensi komunikasi antara perangkat IoT, *server Firebase*, dan aplikasi *monitoring* karena letaknya relatif dekat dengan Indonesia. Selain itu, *database* dikonfigurasi menggunakan *Locked Mode*, sehingga seluruh akses baca (*read*) dan tulis (*write*) ditolak secara *default* sampai aturan keamanan (*Security Rules*) ditentukan. Konfigurasi ini diterapkan untuk meningkatkan keamanan data dan mencegah akses tidak sah terhadap informasi yang tersimpan pada *Smart Composter*.

4.3.4.1 Struktur Real*time Database*

Struktur *Firebase* Real*time Database* dirancang untuk mendukung proses *monitoring* dan pencatatan data pengomposan secara *real-time*. *Database* ini terdiri atas dua *node* utama, yaitu *node* Kompos dan *node* Token. *Node* Kompos digunakan untuk menyimpan data sensor, status aktuator, hasil prediksi kematangan kompos, serta riwayat proses pengomposan pada setiap wadah kompos. Sementara itu, *node* Token digunakan untuk menyimpan token perangkat pengguna yang digunakan sebagai tujuan pengiriman notifikasi melalui *Firebase Cloud Messaging* (FCM). Dengan struktur tersebut, sistem mampu melakukan pemantauan kondisi kompos secara berkelanjutan sekaligus mengirimkan notifikasi kepada pengguna ketika kompos sudah matang pada proses pengomposan.

Gambar 4.12 Struktur Real*time Database Smart Composter*

Berdasarkan gambar 4.12, struktur *Firebase* Real*time Database* pada sistem *monitoring* kematangan kompos terdiri atas tiga *node* utama, yaitu *node* Kompos, *User*, dan Token. *Node* Kompos digunakan untuk menyimpan seluruh data yang berkaitan dengan proses pengomposan, *node User* digunakan untuk menyimpan data pengguna beserta hak akses terhadap fitur sistem, sedangkan *node* Token digunakan untuk menyimpan token perangkat pengguna yang digunakan dalam proses pengiriman notifikasi. Struktur *database* ini dirancang agar mampu mendukung proses pemantauan kondisi kompos secara *real-time*, penyimpanan data historis, pengelolaan hak akses pengguna, serta penyampaian informasi kematangan kompos kepada pengguna melalui layanan notifikasi.

Pada *node* Kompos terdapat beberapa *node* turunan yang merepresentasikan setiap wadah kompos, yaitu Kompos1, Kompos2, Kompos3, dan Kompos4. Masing-masing *node* kompos memiliki dua bagian utama, yaitu *Current* dan Log. *Node Current* digunakan untuk menyimpan kondisi terkini dari proses pengomposan yang diperbarui secara berkala oleh *Raspberry Pi*. Data yang tersimpan pada *node* ini meliputi status aktuator, nilai sensor, status kematangan kompos, kualitas kompos, nilai prediksi Germination Index (GI), serta *timestamp* data tersimpan. Informasi tersebut digunakan untuk menampilkan kondisi aktual kompos pada aplikasi *monitoring* sehingga pengguna dapat memantau proses pengomposan secara langsung.

Di dalam *node Current*, terdapat *node* Aktuator yang berfungsi untuk menyimpan status perangkat yang digunakan dalam proses pengomposan, yaitu kipas, mesin

© 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

© 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

pencacah, mesin pengaduk, dan pompa air. Selain itu, terdapat *node* Sensor yang digunakan untuk menyimpan hasil pembacaan sensor berupa suhu, kelembapan, pH, kadar amonia, berat kompos, dan level. *Node* Status_Kematangan digunakan untuk menyimpan hasil klasifikasi tingkat kematangan kompos, sedangkan *node* Kualitas_Kompos digunakan untuk menyimpan hasil penilaian kualitas kompos yang dihasilkan sistem. Selanjutnya, *node* GI_Prediksi menyimpan nilai hasil prediksi Germination Index yang digunakan sebagai indikator tambahan dalam menentukan kualitas dan kematangan kompos. Adapun *node Timestamp* digunakan untuk menyimpan waktu terakhir data diperbarui pada *database*.

Selain menyimpan data terkini, setiap *node* kompos juga memiliki *node* Log yang berfungsi sebagai penyimpanan data historis proses pengomposan. Data pada *node* ini disimpan dalam bentuk *record* dengan identitas unik. Setiap *record* berisi informasi hasil pembacaan sensor, status kematangan kompos, kualitas kompos, serta informasi siklus pengomposan. Data log disimpan secara berkala setiap dua jam sekali sebagai riwayat perkembangan kondisi kompos. Apabila sistem mendeteksi bahwa kompos telah mencapai kondisi matang sebelum *interval* pencatatan berikutnya, maka data log akan langsung disimpan tanpa menunggu periode dua jam berakhir. Selain itu, setiap data log juga menyimpan *Timestamp*_Awal dan *Timestamp*_Akhir yang menunjukkan rentang waktu pencatatan data pada periode tersebut. Informasi ini digunakan untuk mempermudah proses pelacakan dan analisis perkembangan kondisi kompos pada setiap siklus pengomposan.

Node Token digunakan untuk menyimpan token perangkat pengguna yang diperoleh dari layanan *Firebase Cloud Messaging* (FCM). Token tersebut merepresentasikan perangkat pengguna yang sedang mengakses aplikasi *monitoring* berbasis *Website*. Data token digunakan sebagai alamat tujuan pengiriman notifikasi sehingga sistem dapat mengirimkan pemberitahuan secara otomatis kepada pengguna ketika terjadi perubahan kondisi tertentu, seperti perubahan status kematangan kompos atau informasi penting lainnya yang memerlukan perhatian pengguna.

Node User digunakan untuk menyimpan data pengguna yang telah terdaftar pada sistem beserta informasi hak aksesnya. Setiap pengguna memiliki identitas unik

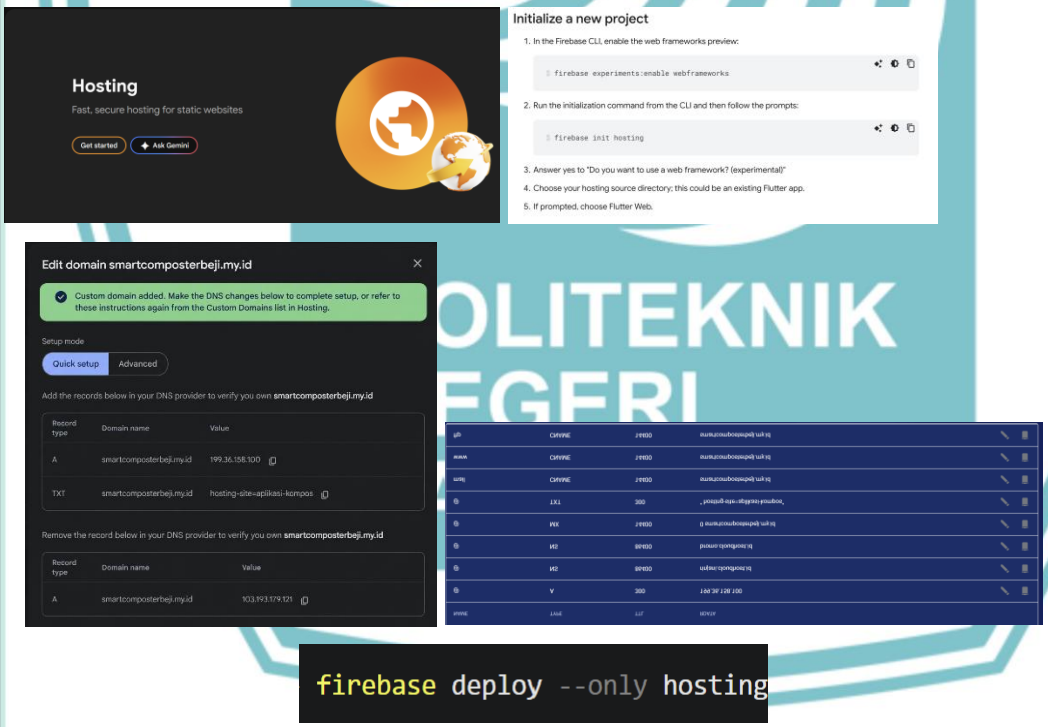
© Hak Cipta milik Politeknik Negeri Jakarta

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

yang diperoleh dari *Firebase Authentication* dan menyimpan informasi seperti nama, alamat email, serta status hak akses terhadap fitur tertentu. Pada penelitian ini, *node User* dimanfaatkan untuk mengatur hak akses fitur *remote* mesin pencacah. Administrator dapat memberikan atau mencabut izin kepada pengguna untuk mengoperasikan mesin pencacah dari aplikasi. Apabila pengguna telah diberikan hak akses oleh administrator, maka fitur *remote* akan dapat digunakan. Sebaliknya, apabila hak akses belum diberikan, pengguna tidak dapat mengoperasikan mesin pencacah melalui aplikasi sehingga keamanan dan pengendalian perangkat tetap terjaga.

4.3.5 Implementasi *Firebase Hosting*

Firebase Hosting digunakan untuk melakukan *deployment* aplikasi *Smart Composter* berbasis *Website* serta halaman yang terintegrasi dengan *Firebase Authentication*.



Gambar 4.13 Konfigurasi *Firebase Hosting*

Berdasarkan gambar 4.13, setelah konfigurasi *Firebase* CLI selesai dilakukan, tahap berikutnya adalah implementasi *Firebase Hosting* yang digunakan sebagai layanan *hosting* untuk aplikasi *Website Smart Composter*. Proses konfigurasi diawali dengan mengaktifkan fitur *Firebase Hosting* melalui *Firebase* CLI dan melakukan inisialisasi *hosting* pada *project Firebase* yang telah dibuat. Pada tahap

© 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

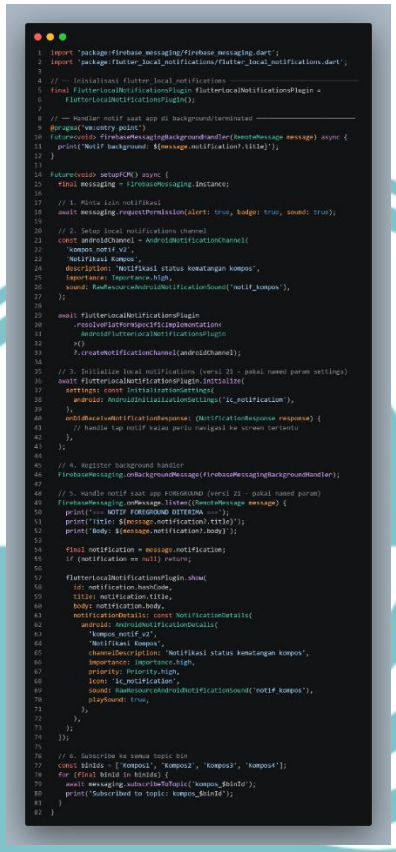
ini ditentukan direktori sumber aplikasi *Website* yang akan di-deploy ke *Firebase Hosting*. Setelah proses inisialisasi selesai, aplikasi *Website* dapat dipublikasikan menggunakan layanan *hosting* yang disediakan oleh *Firebase* sehingga dapat diakses melalui jaringan internet. Proses publikasi aplikasi dilakukan menggunakan perintah *deployment* melalui *Firebase* CLI, yaitu *Firebase* deploy --only *hosting*, yang berfungsi untuk mengunggah hasil build aplikasi *Website* ke *server Firebase Hosting*.

Untuk mempermudah akses pengguna, *Firebase Hosting* kemudian dikonfigurasi menggunakan custom domain dengan alamat smartcomposterbeji.my.id. Konfigurasi custom domain dilakukan dengan menambahkan record DNS yang diberikan oleh *Firebase* ke penyedia layanan domain yang digunakan. Proses ini meliputi penambahan record A dan TXT sebagai mekanisme verifikasi kepemilikan domain serta pengalihan alamat domain menuju layanan *Firebase Hosting*. Setelah konfigurasi DNS berhasil diverifikasi, aplikasi *Website monitoring* dapat diakses melalui alamat domain yang telah ditentukan sehingga pengguna tidak perlu menggunakan alamat domain bawaan *Firebase Hosting*. Dengan demikian, aplikasi *Smart Composter* dapat diakses secara langsung melalui domain yang lebih mudah diingat dan lebih representatif terhadap sistem yang dikembangkan.

4.3.6 Implementasi *Firebase Cloud Messaging*

Firebase Cloud Messaging (FCM) diimplementasikan sebagai layanan pengiriman notifikasi pada *Smart Composter*. Layanan ini digunakan untuk menyampaikan informasi secara *real-time* kepada pengguna ketika terjadi perubahan kondisi pada proses pengomposan yang terdeteksi oleh sistem. Pada penelitian ini, FCM diintegrasikan dengan aplikasi berbasis *Flutter* sehingga pengguna dapat menerima notifikasi meskipun aplikasi tidak sedang digunakan secara aktif. Proses pengiriman notifikasi dilakukan melalui *Firebase Cloud Functions* yang akan dipicu oleh perubahan data pada *Firebase* Real*time Database*, kemudian

meneruskan pesan notifikasi ke perangkat pengguna yang telah terdaftar melalui token FCM.



```dart
import 'package:firebase_messaging/firebase_messaging.dart';
import 'package:flutter_local_notifications/flutter_local_notifications.dart';

// -- Inisialisasi flutter_local_notifications ----------
final FlutterLocalNotificationsPlugin flutterLocalNotificationsPlugin =
    FlutterLocalNotificationsPlugin();

// -- Handler notif saat app di background/terminated ---------
@pragma('vm:entry-point')
Future<void> firebaseMessagingBackgroundHandler(RemoteMessage message) async {
  print('Notif background: ${message.notification?.title}');
}

Future<void> setupFCM() async {
  final messaging = FirebaseMessaging.instance;

  // 1. Minta izin notifikasi
  await messaging.requestPermission(alert: true, badge: true, sound: true);

  // 2. Setup local notifications channel
  const androidChannel = AndroidNotificationChannel(
    'kompos_notif_v2',
    'Notifikasi Kompos',
    description: 'Notifikasi status kematangan kompos',
    importance: Importance.high,
    sound: RawResourceAndroidNotificationSound('notif_kompos'),
  );

  await flutterLocalNotificationsPlugin
      .resolvePlatformSpecificImplementation<
        AndroidFlutterLocalNotificationsPlugin
      >()
      ?.createNotificationChannel(androidChannel);

  // 3. Initialize local notifications (versi 21 - pakai named param settings)
  await flutterLocalNotificationsPlugin.initialize(
    settings: const InitializationSettings(
      android: AndroidInitializationSettings('ic_notification'),
    ),
    onDidReceiveNotificationResponse: (NotificationResponse response) {
      // handle tap notif kalau perlu navigasi ke screen tertentu
    },
  );

  // 4. Register background handler
  FirebaseMessaging.onBackgroundMessage(firebaseMessagingBackgroundHandler);

  // 5. Handle notif saat app FOREGROUND (versi 21 - pakai named param)
  FirebaseMessaging.onMessage.listen((RemoteMessage message) {
    print('=== NOTIF FOREGROUND DITERIMA ===');
    print('title: ${message.notification?.title}');
    print('body: ${message.notification?.body}');

    final notification = message.notification;
    if (notification == null) return;

    flutterLocalNotificationsPlugin.show(
      id: notification.hashCode,
      title: notification.title,
      body: notification.body,
      notificationDetails: const NotificationDetails(
        android: AndroidNotificationDetails(
          'kompos_notif_v2',
          'Notifikasi Kompos',
          channelDescription: 'Notifikasi status kematangan kompos',
          importance: Importance.high,
          priority: Priority.high,
          icon: 'ic_notification',
          sound: RawResourceAndroidNotificationSound('notif_kompos'),
          playSound: true,
        ),
      ),
    );
  });

  // 6. Subscribe ke semua topic bin
  const binIds = ['Kompos1', 'Kompos2', 'Kompos3', 'Kompos4'];
  for (final binId in binIds) {
    await messaging.subscribeToTopic('kompos_$binId');
    print('Subscribed to topic: kompos_$binId');
  }
}
```

Gambar 4.14 Setup *Firebase Cloud Messaging*

Berdasarkan gambar 4.14, pada sisi aplikasi *Flutter*, implementasi *Firebase Cloud Messaging* diawali dengan menambahkan dependensi *Firebase Messaging* serta melakukan konfigurasi *Firebase* pada aplikasi. Setelah aplikasi berhasil terhubung dengan *Firebase*, sistem akan meminta izin kepada pengguna untuk menerima notifikasi. Selanjutnya, aplikasi dikonfigurasi untuk berlangganan pada topik notifikasi yang telah ditentukan sesuai dengan wadah kompos yang dimonitor. Mekanisme ini memungkinkan perangkat Android menerima notifikasi yang dikirimkan oleh *Firebase Cloud Functions* melalui *Firebase Cloud Messaging* tanpa memerlukan penyimpanan token perangkat pada *Firebase* Real*time Database*. Selain itu, aplikasi juga dikonfigurasi untuk menangani notifikasi ketika berada pada kondisi foreground, background, maupun terminated sehingga informasi yang dikirimkan oleh sistem dapat diterima dan ditampilkan kepada pengguna pada berbagai kondisi penggunaan aplikasi.

© Hak Cipta milik Politeknik Negeri Jakarta

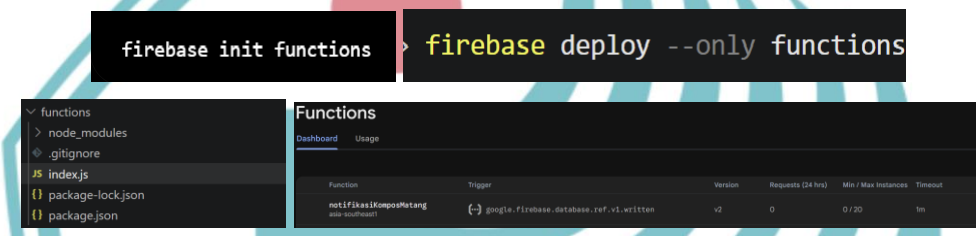
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
   a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
   b. Pengutipan tidak merugikan 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

© 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

4.3.7 Implementasi *Firebase Cloud Functions*

Firebase Cloud Functions diimplementasikan sebagai layanan *backend* yang berfungsi untuk menjalankan proses otomatis berdasarkan perubahan data pada *Firebase* Real*time Database*. Pada penelitian ini, *Cloud Functions* digunakan untuk memantau perubahan kondisi kompos *Firebase* Real*time Database*. Ketika status kompos matang, fungsi akan dijalankan secara otomatis tanpa memerlukan intervensi pengguna. Selain itu, *Cloud Functions* juga berperan dalam proses pengiriman notifikasi kepada pengguna melalui *Firebase Cloud Messaging* (FCM) sehingga notifikasi kematangan kompos dapat diterima secara *real-time* oleh pengguna aplikasi.



Gambar 4.15 Konfigurasi *Cloud* Function *Firebase*

Pada gambar 4.15, implementasi *Firebase Cloud Functions* diawali dengan melakukan inisialisasi layanan *Cloud Functions* menggunakan *Firebase* CLI melalui perintah *Firebase* init *functions*. Proses inisialisasi menghasilkan folder *functions* yang berisi *file* konfigurasi dan *source* code *Cloud Functions*. Setelah konfigurasi selesai dilakukan, *Cloud Functions* digunakan untuk memantau perubahan data pada *Firebase* Real*time Database* dan menjalankan proses otomatis, seperti pengiriman notifikasi kepada pengguna melalui *Firebase Cloud Messaging*. Fungsi yang telah dibuat kemudian diunggah ke *server Firebase* menggunakan perintah *deployment* sehingga dapat berjalan secara otomatis ketika kondisi yang ditentukan terpenuhi.

4.3.8 Implementasi Antarmuka Android

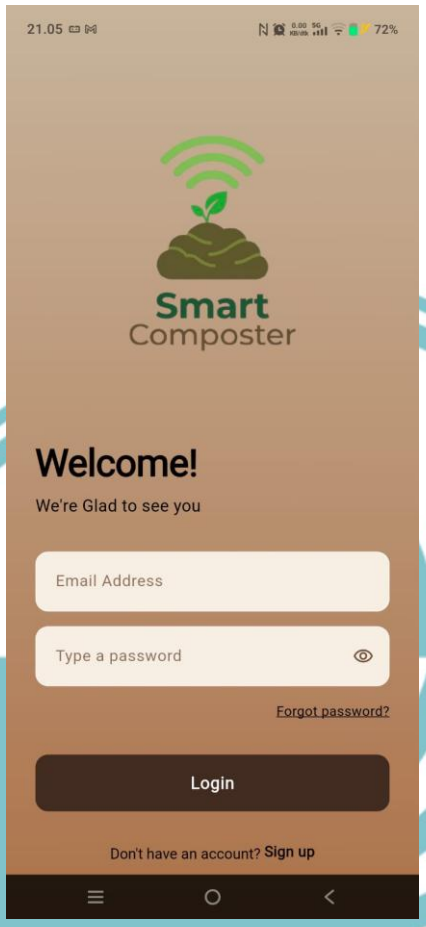
Implementasi antarmuka Android dilakukan untuk menyediakan media interaksi antara pengguna dan *backend server Smart Composter*. Aplikasi dibangun menggunakan *Flutter* dan terintegrasi dengan layanan *Firebase*.

4.3.8.1 Halaman *Login*

Berikut ini halaman *login* pada tampilan android.

© 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 4.16 Halaman *Login*

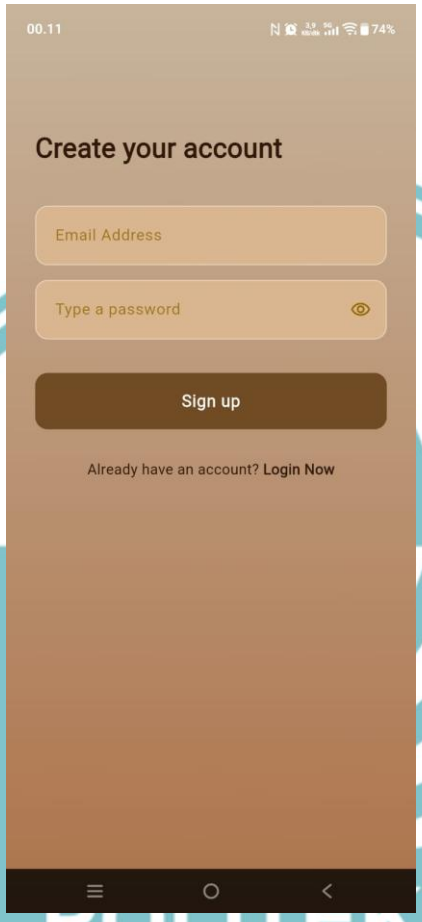
Berdasarkan gambar 4.16, halaman *Login* dikembangkan sebagai gerbang utama autentikasi pengguna sebelum mengakses fitur pada aplikasi *Smart Composter*. Pada halaman ini pengguna dapat memasukkan alamat email dan *password* yang telah terdaftar untuk melakukan proses *login* menggunakan *Firebase Authentication*. Selain itu, tersedia fitur *Forgot Password* untuk membantu pengguna melakukan reset *password* apabila lupa kata sandi, serta tautan Sign Up yang digunakan untuk mengarahkan pengguna ke halaman registrasi akun baru.

© 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

4.3.8.2 Halaman Register

Berikut ini halaman register pada tampilan android.



Gambar 4.17 Halaman Register

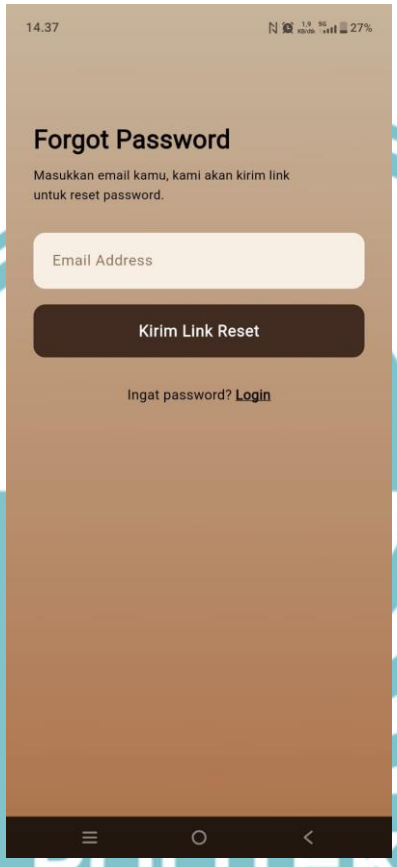
Berdasarkan gambar 4.17, halaman Registrasi dikembangkan untuk memungkinkan pengguna membuat akun baru pada aplikasi *Smart Composter*. Pada halaman ini pengguna diminta memasukkan alamat email dan *password* yang akan digunakan sebagai kredensial *login*. Proses pendaftaran dilakukan menggunakan *Firebase Authentication* sehingga data akun dapat dikelola secara aman. Selain itu, tersedia tautan menuju halaman *login* bagi pengguna yang telah memiliki akun dan ingin masuk ke dalam aplikasi.

© 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

4.3.8.3 Halaman *Forgot Password*

Berikut ini halaman *forgot password* pada tampilan android.



Gambar 4.18 Halaman *Forgot Password*

Berdasarkan gambar 4.18, halaman *Forgot Password* dikembangkan untuk membantu pengguna mengatur ulang *password* akun apabila lupa kata sandi yang digunakan untuk *login*. Pada halaman ini pengguna diminta memasukkan alamat email yang telah terdaftar pada sistem. Setelah email dikirim, *Firebase Authentication* akan mengirimkan tautan reset *password* ke alamat email tersebut sehingga pengguna dapat membuat *password* baru dan kembali mengakses akun pada aplikasi *Smart Composter*.

© 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

4.3.8.4 Halaman Reset *Password*

Berikut ini halaman reset *password* pada tampilan android.

Gambar 4.19 Halaman Reset *Password*

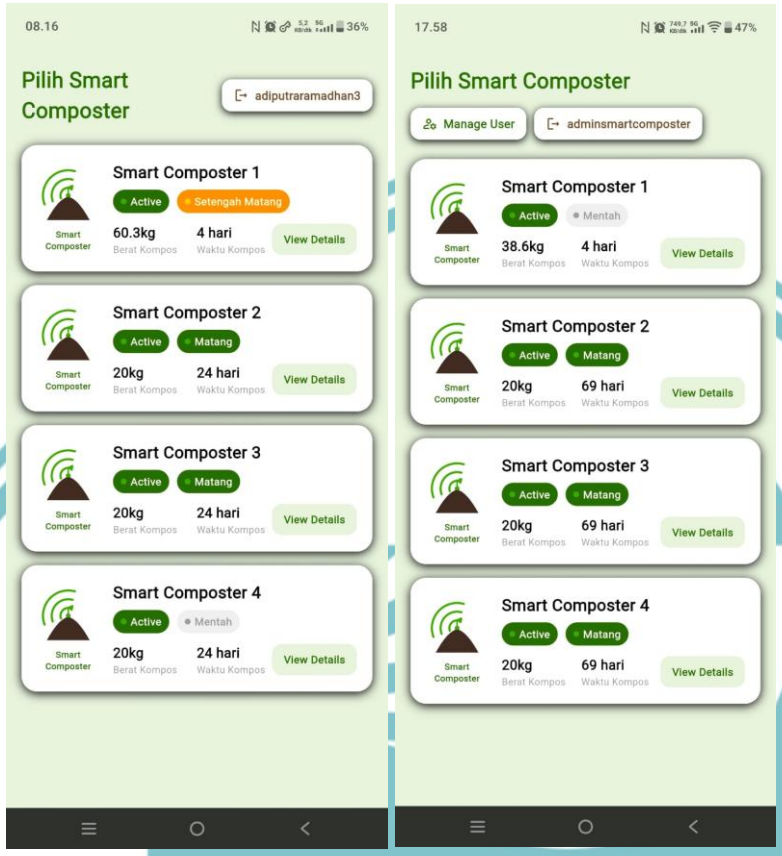
Berdasarkan gambar 4.19, tampilkan halaman reset *password* yang digunakan pengguna untuk mengatur ulang kata sandi akun. Pada halaman ini, pengguna diminta memasukkan *password* baru dan mengonfirmasi kembali *password* tersebut. Sistem akan melakukan validasi untuk memastikan *password* memenuhi ketentuan yang telah ditetapkan sebelum perubahan *password* disimpan ke sistem. Setelah proses berhasil, pengguna dapat menggunakan *password* baru untuk masuk ke aplikasi.

© 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

4.3.8.5 Halaman Pemilihan Bin Kompos

Berikut ini halaman pemilihan bin kompos pada tampilan android.



Gambar 4.20 Halaman Pemilihan Bin Kompos

Berdasarkan Gambar 4.20, halaman pemilihan bin kompos merupakan halaman yang digunakan untuk menampilkan daftar *Smart Composter* yang terhubung dengan sistem. Pada halaman ini ditampilkan informasi ringkas setiap *Smart Composter*, meliputi status perangkat, tingkat kematangan kompos (Mentah, Setengah Matang, atau Matang), berat kompos, serta lama waktu proses pengomposan. Data yang ditampilkan diperoleh dari *Firebase* Real*time Database* sehingga perubahan kondisi kompos dapat langsung terlihat pada aplikasi. Selain itu, tersedia tombol *View Details* yang digunakan untuk mengakses halaman *monitoring* dan melihat kondisi kompos secara lebih rinci pada *Smart Composter* yang dipilih. Khusus pada akun admin, halaman ini juga menyediakan fitur *Manage User* yang digunakan untuk mengelola hak akses pengguna. Melalui fitur tersebut, admin dapat memberikan hak akses kepada *user* umum untuk menjadi petugas bank sampah sehingga memperoleh akses untuk mengendalikan aktuator melalui fitur

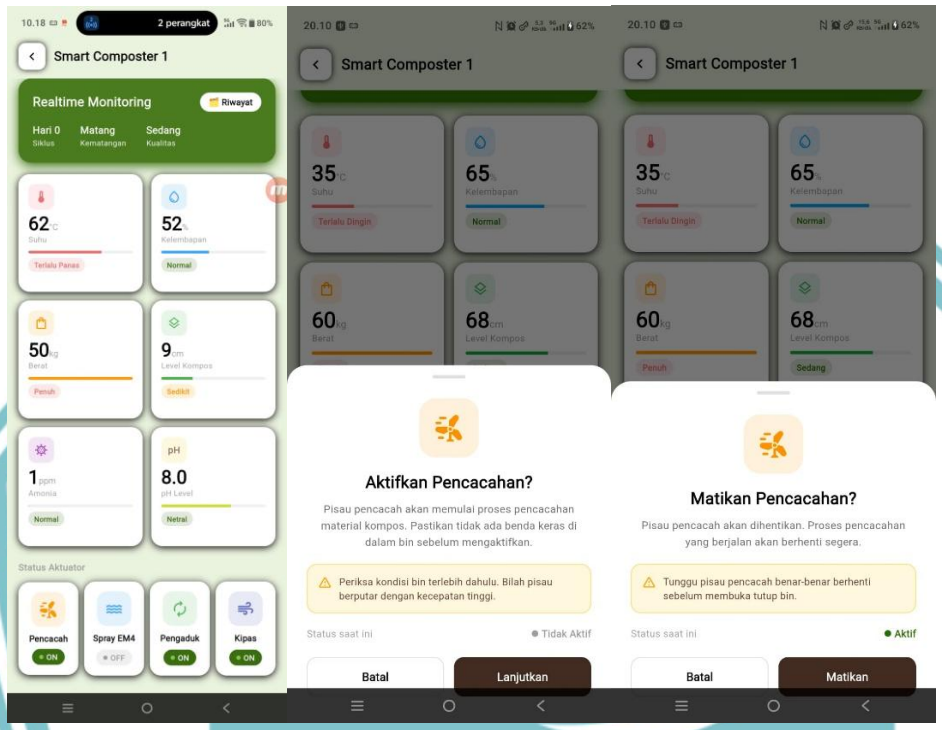
© Hak Cipta milik Politeknik Negeri Jakarta

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

remote. Fitur *Manage User* hanya dapat diakses oleh admin dan tidak ditampilkan pada akun *user* umum maupun petugas bank sampah.

4.3.8.6 Halaman *Dashboard Monitoring*

Berikut ini halaman *dashboard monitoring* pada tampilan android.



Gambar 4.21 Halaman *Dashboard Monitoring*

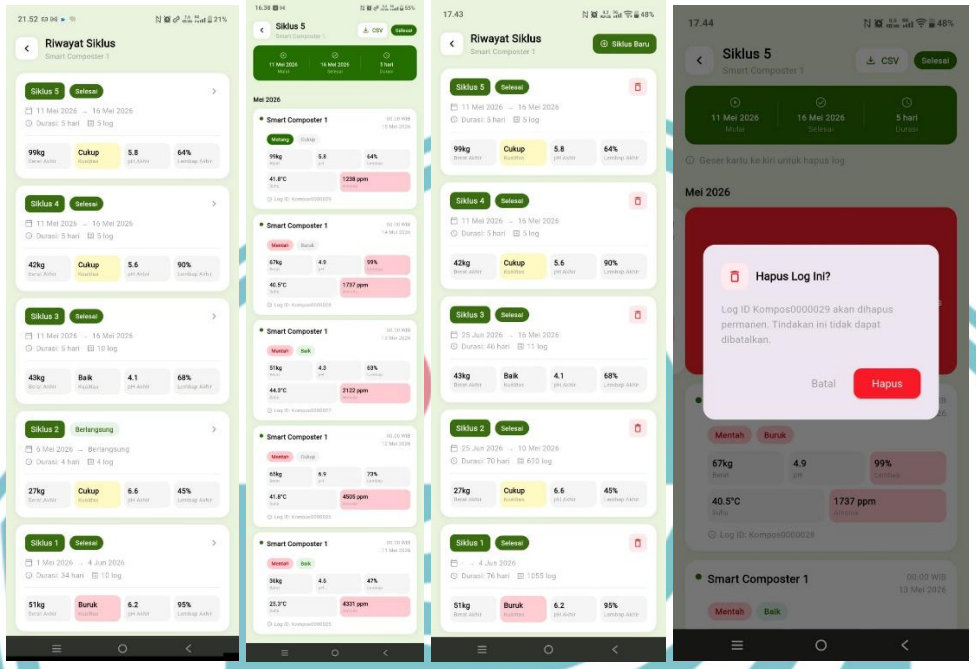
Berdasarkan Gambar 4.21, halaman *dashboard monitoring* dikembangkan untuk menampilkan kondisi *Smart Composter* berdasarkan data yang diperoleh dari sensor. Halaman ini menyajikan berbagai parameter *monitoring* seperti suhu, kelembapan, berat kompos, level, kadar amonia, dan pH, serta informasi mengenai siklus, tingkat kematangan, dan kualitas kompos. Selain fungsi *monitoring*, halaman ini juga menyediakan fitur pengendalian aktuator seperti pencacah, yang dilengkapi dengan dialog konfirmasi untuk memastikan setiap tindakan dilakukan secara aman dan terkontrol serta status spray EM4, pengaduk dan kipas.

© 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

4.3.8.7 Halaman Riwayat Kompos

Berikut ini halaman riwayat pada tampilan android.



Gambar 4.22 Halaman Riwayat Kompos

Berdasarkan Gambar 4.22, halaman riwayat kompos digunakan untuk menampilkan riwayat proses pengomposan yang tersimpan pada setiap *Smart Composter*. Halaman ini terdiri dari daftar siklus pengomposan yang menampilkan ringkasan informasi seperti tanggal pengomposan, durasi, jumlah log, kualitas kompos, berat akhir, pH akhir, dan kelembapan akhir, serta halaman detail siklus yang menampilkan seluruh data log *monitoring* yang dicatat secara berkala setiap 2 jam sekali maupun saat kompos telah mencapai status matang. Pada halaman ini, hak akses pengguna dibedakan berdasarkan tingkat kewenangannya. *User* umum dan petugas bank sampah dapat melihat riwayat siklus dan detail data log pengomposan, sedangkan admin memiliki hak akses tambahan untuk membuat siklus pengomposan baru, menghapus siklus, serta menghapus data log yang tersimpan. Sebelum proses penghapusan dilakukan, sistem menampilkan dialog konfirmasi untuk mencegah penghapusan data secara tidak disengaja. Selain itu, aplikasi menyediakan fitur *export* log ke dalam format CSV (*Comma-Separated Values*) sehingga data *monitoring* pada suatu siklus pengomposan dapat diunduh dan digunakan untuk analisis lebih lanjut. *File* CSV tersebut dapat diolah

menggunakan perangkat lunak seperti Micros*oft* Excel atau aplikasi pengolah data lainnya untuk keperluan evaluasi, visualisasi, maupun dokumentasi hasil pengomposan. Dengan adanya halaman ini, pengguna dapat meninjau kembali perkembangan dan hasil proses pengomposan, sementara admin dapat melakukan pengelolaan data siklus dan log pengomposan sesuai dengan hak akses yang dimiliki.

4.3.8.8 Halaman Kelola Pengguna

Berikut ini merupakan halaman kelola pengguna yang dapat diakses oleh admin.

Gambar 4.23 Halaman Kelola Pengguna

Berdasarkan Gambar 4.23, halaman kelola pengguna digunakan oleh admin untuk mengelola hak akses pengguna yang telah terdaftar pada sistem. Melalui halaman ini, admin dapat meningkatkan hak akses *user* umum menjadi petugas bank sampah. Pengguna yang telah memiliki hak akses sebagai petugas bank sampah dapat mengakses fitur *remote* untuk mengendalikan mesin pencacah, sedangkan *user* umum hanya dapat mengakses fitur *monitoring* dan informasi yang tersedia pada aplikasi.

4.3.9 Implementasi Antarmuka *Website*

Aplikasi *monitoring* juga diimplementasikan dalam bentuk *Website* menggunakan *Flutter Website*. Tampilan *Website* dirancang dengan fungsi yang sama seperti aplikasi Android, namun dilakukan penyesuaian ukuran dan tata letak antarmuka agar dapat ditampilkan secara optimal pada layar desktop.

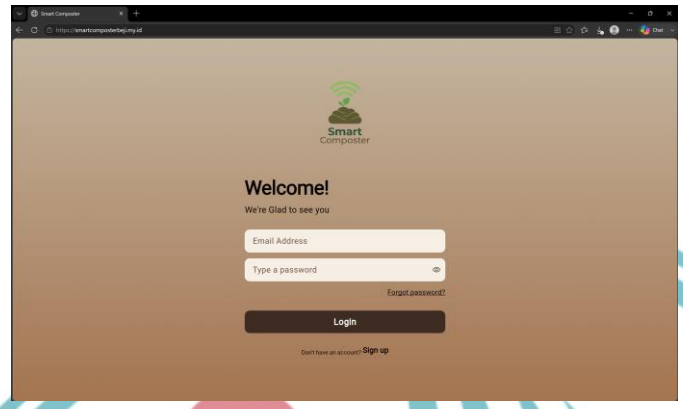
© Hak Cipta milik Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu 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

© 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

4.3.9.1 Halaman *Login*

Berikut ini halaman *login* pada tampilan *Websitesite*.

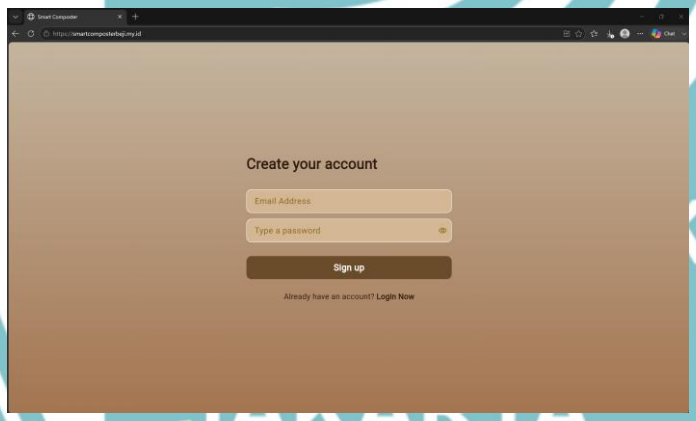


Gambar 4.24 Halaman *Login Websitesite*

Berdasarkan gambar 4.24, ukuran halaman *login* disesuaikan dengan ukuran *device* desktop.

4.3.9.2 Halaman Register

Berikut ini halaman register pada tampilan *Websitesite*.



Gambar 4.25 Halaman *Register Websitesite*

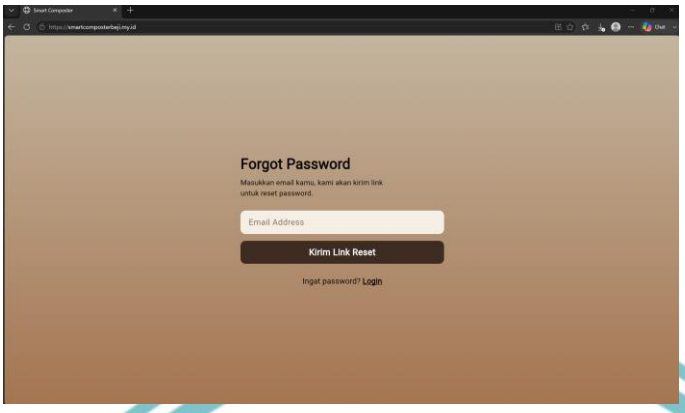
Berdasarkan gambar 4.25, ukuran halaman *register* disesuaikan dengan ukuran *device* desktop

4.3.9.3 Halaman *Forgot Password*

Berikut ini halaman *forgot password* pada tampilan *Websitesite*.

© 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

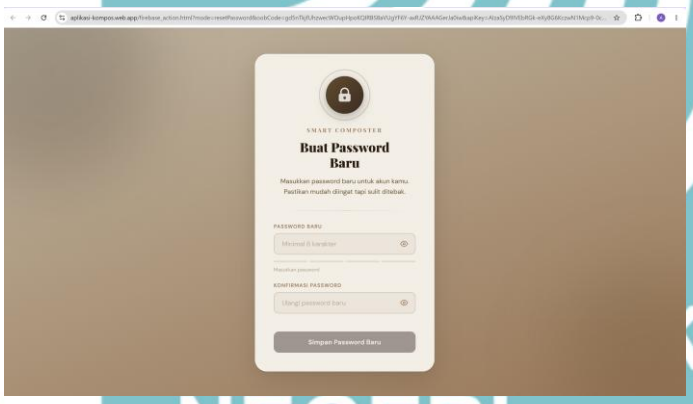


Gambar 4.26 Halaman *Forgot Password Websitesite*

Berdasarkan gambar 4.26, ukuran halaman *forgot password* disesuaikan dengan ukuran *device* desktop

4.3.9.4 Halaman Reset *Password*

Berikut ini halaman Reset *Password* pada tampilan *Websitesite*.



Gambar 4.27 Halaman *Reset Password Websitesite*

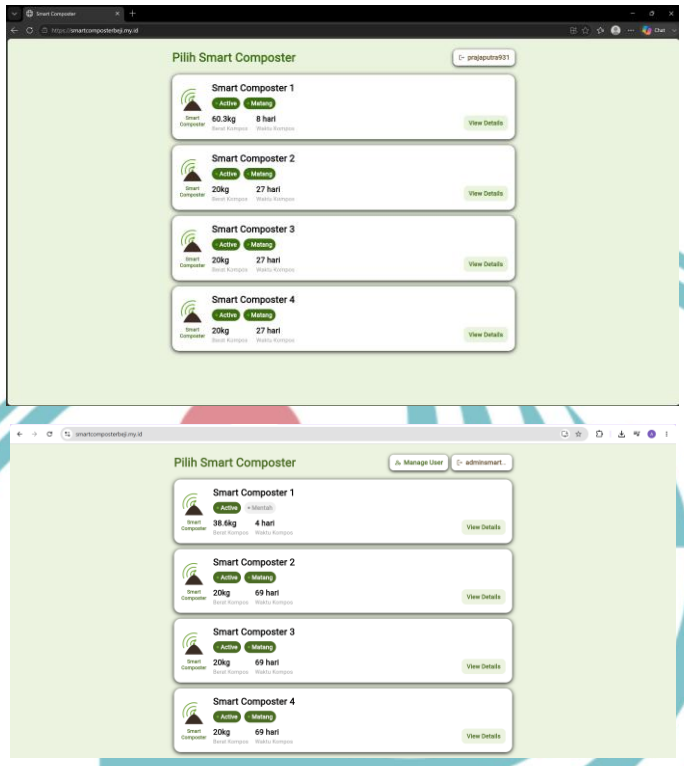
Berdasarkan gambar 4.27, ukuran halaman *reset password* disesuaikan dengan ukuran *device* desktop

© 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

4.3.9.5 Halaman Pemilihan Bin Kompos

Berikut ini halaman pemilihan bin kompos pada tampilan *Websitesite*.

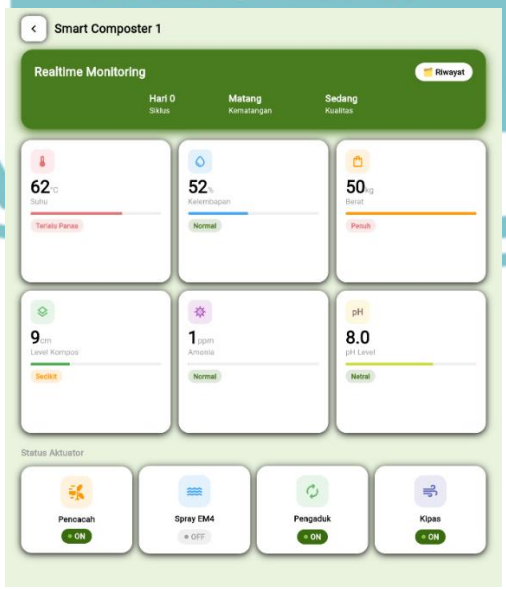


Gambar 4.28 Halaman Pemilihan *Bin* Kompos *Websitesite*

Berdasarkan gambar 4.28, ukuran halaman pemilihan bin kompos disesuaikan dengan ukuran *device* desktop

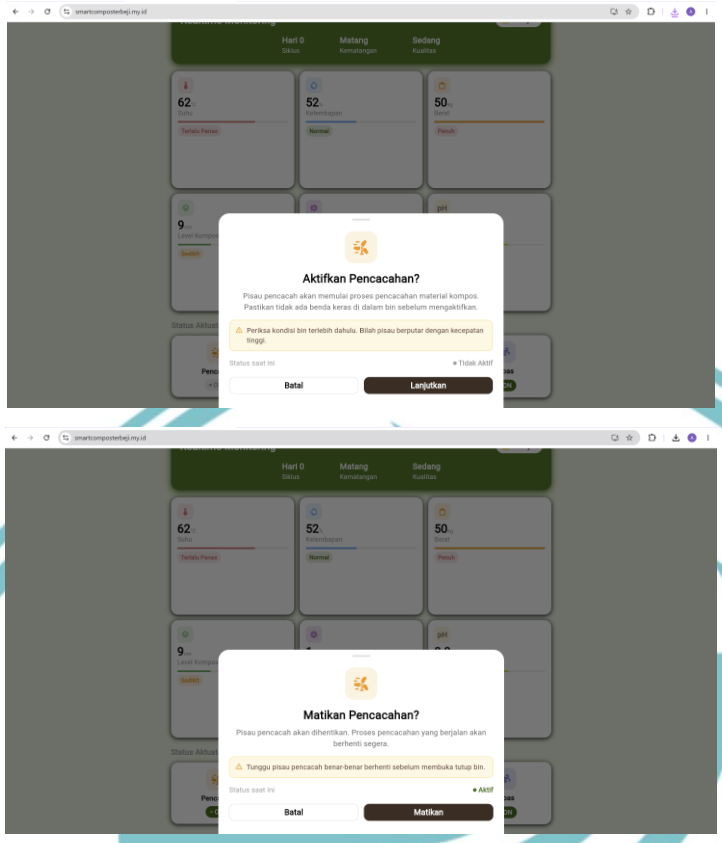
4.3.9.6 Halaman *Dashboard Monitoring*

Berikut ini halaman *dashboard monitoring* pada tampilan *Websitesite*.



© 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

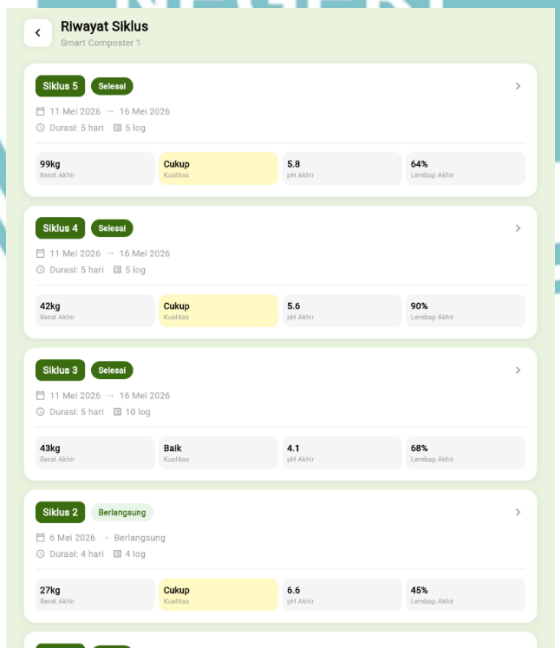


Gambar 4.29 Halaman *Dashboard Monitoring Websitesite*

Berdasarkan gambar 4.29, ukuran halaman *dashboard monitoring* disesuaikan dengan ukuran *device* desktop

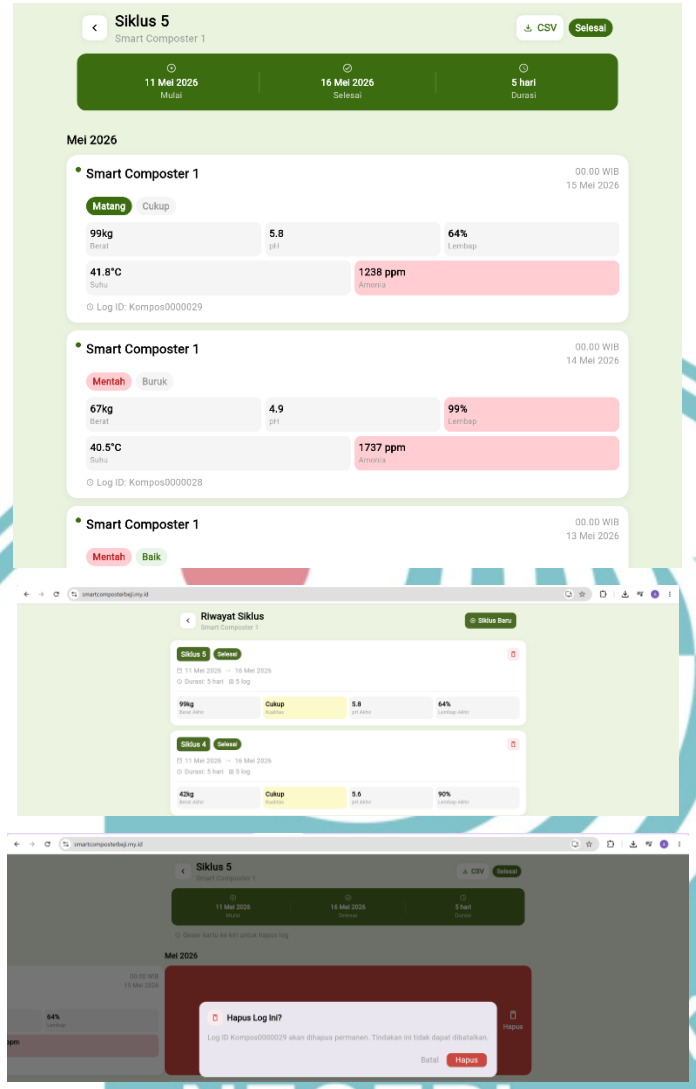
4.3.9.7 Halaman Riwayat Kompos

Berikut ini halaman riwayat kompos pada tampilan *Websitesite*.



© 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

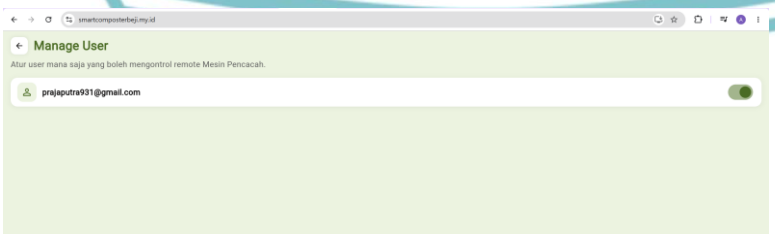


Gambar 4.30 Halaman Riwayat Kompos *Websitesite*

Berdasarkan gambar 4.30, ukuran halaman riwayat disesuaikan dengan ukuran *device* desktop

4.3.9.8 Halaman Kelola Pengguna

Berikut ini halaman Kelola Pengguna pada tampilan *Websitesite*.



Gambar 4.31 Halaman Kelola Pengguna *Websitesite*

Berdasarkan gambar 4.31, ukuran halaman Kelola Pengguna disesuaikan dengan ukuran *device* desktop

© 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

4.4 Pengujian

Pengujian dilakukan untuk memastikan bahwa aplikasi *Smart Composter* yang telah dikembangkan dapat berjalan sesuai dengan kebutuhan pengguna. Pengujian yang dilakukan pada penelitian ini terdiri dari pengujian fungsional menggunakan metode *Black box Testing*, pengujian kualitas perangkat lunak menggunakan standar ISO/IEC 25010, pengujian *Quality of Service* (QoS) menggunakan standar TIPHON, serta pengujian *reliability server*. Pengujian *Black box* digunakan untuk memastikan seluruh fitur aplikasi berjalan sesuai dengan fungsinya, sedangkan pengujian ISO/IEC 25010 digunakan untuk mengetahui tingkat kualitas aplikasi berdasarkan penilaian pengguna.

Pengujian QoS dilakukan untuk mengetahui kualitas komunikasi data antara *Firebase* Real*time Database* dan aplikasi *Smart Composter*. Pengujian ini bertujuan untuk mengukur performa pertukaran data yang terjadi selama proses *monitoring* secara *real-time* dengan menggunakan parameter *delay*, *Jitter*, *throughput* dan *packet loss*. Melalui pengujian ini dapat diketahui tingkat kecepatan pengiriman data, kapasitas transfer data, serta keandalan komunikasi yang terjadi antara *server* dan aplikasi. Hasil pengujian QoS digunakan untuk memastikan bahwa sistem mampu menyampaikan informasi kondisi kompos secara cepat, stabil, dan tanpa kehilangan data selama proses *monitoring* berlangsung.

Selain itu, pengujian *reliability server* dilakukan untuk mengevaluasi kemampuan layanan *backend server* dalam mempertahankan operasional sistem secara konsisten selama proses *monitoring* berlangsung. Pengujian ini dilakukan dengan mengamati keberhasilan layanan *Firebase* dalam menerima, menyimpan, dan menyinkronkan data sensor yang dikirimkan oleh *Raspberry Pi* ke *Firebase* Real*time Database* serta mendistribusikannya kepada aplikasi *monitoring*. Parameter yang digunakan dalam pengujian *reliability* meliputi *availability*, *success rate*, *failure rate*, dan integritas data. Hasil pengujian ini digunakan untuk memastikan bahwa *backend server* mampu menyediakan layanan secara stabil dan andal sehingga proses *Smart Composter* dapat berjalan tanpa gangguan yang signifikan.

© 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

4.4.1 Deskripsi Pengujian

Pada penelitian ini dilakukan empat jenis pengujian yang meliputi pengujian fungsionalitas menggunakan metode *Black box Testing*, pengujian kualitas perangkat lunak berdasarkan standar ISO/IEC 25010, pengujian *Quality of Service* (QoS) menggunakan standar TIPHON, dan pengujian *reliability server*. Pengujian tersebut dilakukan untuk memastikan bahwa *Smart Composter* yang dikembangkan mampu beroperasi sesuai dengan kebutuhan pengguna serta memiliki kualitas layanan yang baik dari sisi aplikasi maupun *backend server*.

Pengujian *Black box Testing* digunakan untuk memverifikasi fungsi-fungsi yang terdapat pada aplikasi tanpa memperhatikan struktur kode program. Pengujian ini berfokus pada kesesuaian antara masukan yang diberikan dengan keluaran yang dihasilkan oleh sistem. Selanjutnya, pengujian ISO/IEC 25010 dilakukan untuk mengevaluasi kualitas perangkat lunak dari aspek *usability* berdasarkan penilaian pengguna setelah menggunakan aplikasi *Smart Composter*.

Selain itu, pengujian QoS dilakukan untuk mengukur kualitas komunikasi data antara *Firebase* Real*time Database* dan aplikasi *Smart Composter*. Parameter yang digunakan meliputi *delay*, *jitter*, *throughput* dan *packet loss* yang dianalisis menggunakan perangkat lunak *Wireshark*. Pengujian ini bertujuan untuk mengetahui tingkat kecepatan, kapasitas transfer data, dan keandalan komunikasi selama proses *monitoring* berlangsung.

Pengujian *reliability server* dilakukan untuk mengevaluasi kemampuan *backend server* dalam menyediakan layanan secara konsisten selama proses *monitoring* berlangsung. Pengujian ini difokuskan pada kinerja layanan *Firebase* dalam menerima, menyimpan, dan mendistribusikan data secara *real-time*. Parameter yang digunakan meliputi *availability*, *success rate*, *failure rate*, dan integritas data. Hasil pengujian *reliability* digunakan untuk mengetahui tingkat keandalan *backend server* dalam mendukung proses *Smart Composter* secara berkelanjutan.

4.4.2 Prosedur Pengujian

4.4.2.1 Prosedur *Black box Testing*

Langkah-langkah pengujian *Black box* adalah sebagai berikut:

1. Menentukan fitur yang akan diuji.
2. Menentukan skenario pengujian untuk setiap fitur.

© 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. Memberikan input sesuai skenario pengujian.
4. Mengamati hasil keluaran sistem.
5. Membandingkan hasil aktual dengan hasil yang diharapkan.
6. Menentukan status pengujian berhasil atau gagal.

Skenario pengujian *Black box Testing*

No	Fitur	Skenario Pengujian
1	*Login*	Memasukan email dan *password* yang benar
2	*Login*	Memasukan email atau *password* yang salah
3	Registrasi	Mengisi email dan *password* sesuai ketentuan
4	Registrasi	Mengisi email dan *password* sesuai ketentuan
5	Verifikasi Email	Klik *link* verifikasi dan berhasil
6	Verifikasi Email	Tidak klik *link* verifikasi atau *link* kadaluarsa
7	Reset *Password*	Mengisi sesuai aturan
8	Reset *Password*	Tidak mengisi sesuai aturan
9	Pemilihan Bin Kompos	Memilih bin komposter 1
10	Pemilihan Bin Kompos	Memilih bin komposter 2
11	Pemilihan Bin Kompos	Memilih bin komposter 3
12	Pemilihan Bin Kompos	Memilih bin komposter 4
13	*Monitoring*	Data sensor pada *Firebase* berubah
14	*Remote*	Mengaktifkan/mematikan mesin pencacah saat belum dapat akses
15	*Remote*	Mengaktifkan/mematikan mesin pencacah saat sudah dapat akses
16	Siklus Baru	Membuat siklus baru setelah siklus sebelumnya berstatus matang

© 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

No	Fitur	Skenario Pengujian
17	Siklus Baru	Menekan tombol "Mulai Siklus Baru" saat siklus sebelumnya selesai
18	Log Kompos	Menyimpan data kompos setiap 2 jam sekali
19	Log Kompos	Menyimpan data kompos saat sudah matang
20	Notifikasi	Mengirimkan notifikasi saat kompos matang
21	Download Log	Pengguna memilih salah satu siklus pengomposan kemudian menekan tombol Export CSV
22	*Logout*	Klik *Logout*

4.4.2.2 Prosedur Pengujian ISO/IEC 25010

Pengujian ISO/IEC 25010 dilakukan melalui tahapan sebagai berikut:

1. Menentukan karakteristik kualitas yang akan diuji.
2. Menyusun instrumen kuesioner berdasarkan karakteristik ISO/IEC 25010.
3. Memberikan aplikasi kepada responden untuk digunakan.
4. Membagikan kuesioner kepada responden.
5. Mengumpulkan hasil penilaian.
6. Menghitung skor dan persentase kualitas aplikasi.

Pengujian *Software Quality Assurance* (SQA) pada penelitian ini dilakukan menggunakan instrumen kuesioner yang mengacu pada karakteristik *usability* dalam standar ISO/IEC 25010. Instrumen tersebut disusun dalam bentuk beberapa pernyataan yang bertujuan untuk memperoleh penilaian responden terhadap kualitas aplikasi yang telah dikembangkan. Daftar pernyataan yang digunakan pada pengujian SQA disajikan pada Tabel 4.1.

Tabel 4.1 Pertanyaan SQA

No	Pertanyaan
1	Tampilan aplikasi mudah dipahami
2	Navigasi aplikasi mudah digunakan
3	Informasi yang ditampilkan mudah dibaca
4	Pengguna dapat mempelajari cara penggunaan aplikasi dengan cepat
5	Fitur-fitur aplikasi mudah ditemukan
6	Aplikasi membantu pengguna dalam memantau kematangan kompos

© 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	Aplikasi memberikan pengalaman penggunaan yang nyaman
8	Secara keseluruhan aplikasi mudah digunakan

Berdasarkan Tabel 4.1, kuesioner SQA terdiri atas beberapa pernyataan yang digunakan untuk mengevaluasi kualitas aplikasi *Smart Composter* dari aspek *usability*. Setiap pernyataan diberikan kepada responden untuk memperoleh penilaian terhadap kemudahan penggunaan aplikasi, kemudahan dalam mempelajari fitur, tampilan antarmuka, serta tingkat kepuasan pengguna selama menggunakan aplikasi. Hasil penilaian dari kuesioner tersebut kemudian diolah menggunakan skala Likert untuk memperoleh nilai *usability* aplikasi.

4.4.2.3 Prosedur Pengujian QoS

Pengujian *Quality of Service* (QoS) dilakukan untuk mengevaluasi kualitas komunikasi data dan respons sistem pada aplikasi *monitoring* kompos berbasis Internet *of Things* (IoT). Pengujian dilakukan pada tiga skenario utama, yaitu pengiriman data sensor, pengiriman notifikasi, dan pengendalian aktuator jarak jauh. Pengujian pengiriman data sensor dilakukan sebanyak 30 kali dengan cara mengubah nilai sensor pada *Firebase* Real*time Database* dan mengamati proses sinkronisasi data pada aplikasi *monitoring*. Parameter yang diukur meliputi *delay*, dan *packet loss* menggunakan perangkat lunak *Wireshark*.

Hasil dari skenario pengujian tersebut digunakan untuk mengevaluasi performa sistem dalam mendukung komunikasi data secara *real-time*.

4.4.2.4 Prosedur Pengujian *Reliability Server*

Berikut ini tabel prosedur pengujian *reliability server*

Tabel 4.2 Prosedur Pengujian *Reliability Server*

No	Parameter	Skenario Pengujian
1	*Availability*	Mengamati ketersediaan layanan *Firebase* Real*time Database* selama proses *monitoring* berlangsung.
2	*Success Rate*	Mengirim data sensor ke *Firebase* Real*time Database* secara berulang dan menghitung jumlah data yang berhasil tersimpan.
3	*Failure rate*	Menghitung jumlah data yang gagal tersimpan selama proses pengiriman data ke *Firebase* Real*time Database*.
4	Integrasi Data	Membandingkan nilai data sensor yang dikirim oleh perangkat dengan data yang tersimpan pada *Firebase* Real*time Database*.

© 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

Berdasarkan tabel 4.2, pengujian *reliability* dilakukan selama 1 jam, 2 jam dan 3 jam dengan mengikuti *interval* pengiriman data yang diterapkan pada sistem, yaitu setiap 30 detik. Dengan *interval* tersebut diperoleh 120 data, 240 data dan 360 data pengamatan yang digunakan untuk mengevaluasi *availability*, *success rate* dan *failure rate*, dan integritas data *backend server*.

4.4.3 Data Hasil Pengujian

Subbab ini menyajikan data hasil pengujian yang diperoleh dari pengujian *Black box Testing*, ISO/IEC 25010, *Quality of Service* (QoS), dan *reliability server*. Data tersebut digunakan untuk menganalisis kinerja *Smart Composter* yang telah dikembangkan.

Hasil Pengujian Fungsionalitas *Black box*

Berikut ini merupakan hasil dari pengujian *Black box*.

Tabel 4.3 Hasil Pengujian *Black box*

No	Fitur	Skenario Pengujian	Hasil yang Diharapkan	Status
1	*Login*	Memasukan email dan *password* yang benar	Pengguna masuk ke batch *screen*	Sukses
2	*Login*	Memasukan email atau *password* yang salah	Pengguna tidak dapat masuk ke batch *screen* dan terdapat notifikasi salah email atau *password*	Sukses
3	Registrasi	Mengisi email dan *password* sesuai ketentuan	Pengguna berhasil dibuat	Sukses
4	Registrasi	Mengisi email dan *password* sesuai ketentuan	Akun gagal dibuat	Sukses
5	Verifikasi Email	Klik *link* verifikasi dan berhasil	Pengguna bisa masuk ke aplikasi	Sukses
6	Verifikasi Email	Tidak klik *link* verifikasi atau *link* kadaluarsa	Pengguna tidak bisa masuk ke aplikasi	Sukses
7	Reset *Password*	Mengisi sesuai aturan	*Password* berubah	Sukses

No	Fitur	Skenario Pengujian	Hasil yang Diharapkan	Status
8	Reset *Password*	Tidak mengisi sesuai aturan	Muncul notif *password* tidak sesuai dengan aturan	Sukses
9	Pemilihan Bin Kompos	Memilih bin komposter 1	Menampilkan *Monitoring Screen* bin 1	Sukses
10	Pemilihan Bin Kompos	Memilih bin komposter 2	Menampilkan *Monitoring Screen* bin 2	Sukses
11	Pemilihan Bin Kompos	Memilih bin komposter 3	Menampilkan *Monitoring Screen* bin 3	Sukses
12	Pemilihan Bin Kompos	Memilih bin komposter 4	Menampilkan *Monitoring Screen* bin 4	Sukses
13	*Monitoring*	Data sensor pada *Firebase* berubah	Aplikasi menerima dan menampilkan data sensor terbaru sesuai dengan nilai yang tersimpan pada *Firebase*.	Sukses
14	*Remote*	Mengaktifkan/mematikan mesin pencacah saat belum dapat akses	Pengguna tidak dapat mengaktifkan/mematikan mesin pencacah	Sukses
15	*Remote*	Mengaktifkan/mematikan mesin pencacah saat sudah dapat akses	Pengguna dapat mengaktifkan/mematikan mesin pencacah	Sukses
16	Siklus Baru	Membuat siklus baru setelah siklus sebelumnya berstatus matang	Sistem berhasil membuat siklus baru	Sukses

© 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

© Hak Cipta milik Politeknik Negeri Jakarta

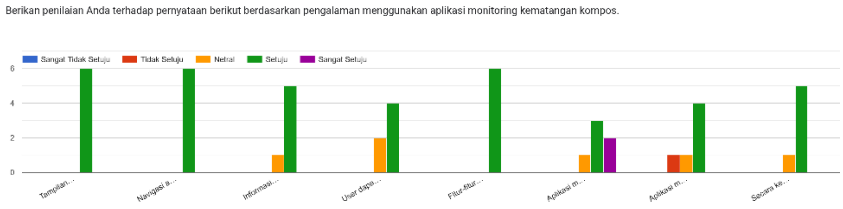
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan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

No	Fitur	Skenario Pengujian	Hasil yang Diharapkan	Status
17	Siklus Baru	Menekan tombol "Mulai Siklus Baru" saat siklus sebelumnya selesai	Siklus baru berhasil ditambahkan ke sistem	Sukses
18	Log Kompos	Menyimpan data kompos setiap 2 jam sekali	Data tersimpan di Riwayat *Screen*	Sukses
19	Log Kompos	Menyimpan data kompos saat sudah matang	Data tersimpan di Riwayat *Screen*	Sukses
20	Notifikasi	Mengirimkan notifikasi saat kompos matang	Muncul notifikasi di Pengguna	Sukses
21	*Download Log*	Pengguna memilih salah satu siklus pengomposan kemudian klik CSV	Sistem menghasilkan dan mengunduh *file* CSV yang berisi seluruh data log *monitoring* pada siklus yang dipilih	Sukses
22	*Logout*	Klik *Logout*	Kembali ke halaman *login*	Sukses

Berdasarkan tabel 4.3, hasil pengujian *Black box Testing* pada Tabel 4.2 yang telah dilakukan terhadap 22 skenario pengujian, tingkat keberhasilan pengujian *Black box* sebesar 100%, seluruh fitur pada aplikasi *Smart Composter* berhasil berjalan sesuai dengan kebutuhan yang telah ditentukan. Pengujian mencakup fitur autentikasi pengguna, pengelolaan akun, pemilihan bin kompos, *monitoring* data sensor, pengendalian mesin pencacah, pengelolaan siklus kompos, pencatatan riwayat kompos, pengiriman notifikasi, hingga proses *logout*. Seluruh skenario pengujian memperoleh status sukses, yang menunjukkan bahwa sistem mampu memberikan keluaran sesuai dengan hasil yang diharapkan pada setiap kondisi pengujian. Dengan demikian, dapat disimpulkan bahwa fungsi-fungsi yang terdapat pada aplikasi telah berjalan dengan baik dan memenuhi kebutuhan fungsional sistem.

4.4.3.1 Hasil Pengujian SQA Standar ISO/IEC 25010

Berikut ini merupakan hasil dari pengujian SQA



Gambar 4.32 Grafik Hasil Pengujian SQA

Perhitungan Nilai

- Jumlah responden = 6 orang
- Jumlah pertanyaan = 8
- Skala maksimum = 5

Maka:

Skor maksimal:

$$6 \times 8 \times 5 = 240$$

skor diperoleh:

$$\frac{186}{240} \times 100\% = 77{,}5\%$$

Berdasarkan hasil pengujian *usability* yang dilakukan terhadap 6 responden, diperoleh total skor sebesar 186 dari skor maksimum 240. Hasil perhitungan menunjukkan nilai *usability* sebesar 77,5%. Nilai tersebut menunjukkan bahwa aplikasi *Smart Composter* memiliki tingkat kemudahan penggunaan yang baik. Pengguna dapat memahami fungsi sistem, mengakses fitur yang tersedia, serta memperoleh informasi yang dibutuhkan melalui aplikasi dengan cukup mudah.

4.4.3.2 Hasil Pengujian QoS

Hasil pengujian *Quality of Service* (QoS) digunakan untuk menilai performa komunikasi data pada sistem yang telah dikembangkan. Pengujian ini dilakukan dengan mengukur beberapa parameter QoS, yaitu *delay*, *jitter*, *throughput* dan *packet loss*. Melalui pengujian tersebut, dapat diketahui kualitas layanan jaringan dalam mendukung proses pertukaran data antara perangkat dan *backend server* selama sistem beroperasi.

© 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

© 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 4.4 Hasil Pengujian *Delay*

No	*Delay* (ms)	Rata-rata *Delay* (ms)	Kategori
1	43.06		
2	46.29		
3	42.58		
4	43.26		
5	49.56		
6	46.22		
7	55.16		
8	49.37		
9	50.79		
10	42.04		
11	53.38		
12	47.20		
13	54.90		
14	53.60		
15	52.34	48,27	Sangat Bagus
16	41.50		
17	52.07		
18	51.99		
19	48.47		
20	43.86		
21	39.98		
22	46.28		
23	52.29		
24	54.39		
25	50.19		
26	44.26		
27	46.82		
28	48.85		
29	51.08		
30	46.28		

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

Berdasarkan tabel 4.4, hasil pengujian *delay* yang dilakukan sebanyak 30 kali, diperoleh nilai *delay* berkisar antara 39,98 ms hingga 55,16 ms, dengan rata-rata *delay* sebesar 48,27 ms. Nilai tersebut menunjukkan bahwa waktu yang dibutuhkan paket data untuk dikirim dari sistem *Smart Composter* menuju aplikasi *monitoring* relatif rendah sehingga proses komunikasi berlangsung dengan cepat. Berdasarkan standar TIPHON, rata-rata *delay* sebesar 48,27 ms termasuk dalam kategori Sangat Bagus, yang mengindikasikan bahwa sistem memiliki latensi yang sangat rendah dan mampu mendukung pengiriman data secara *real-time*. Dengan demikian, komunikasi data antara perangkat *Smart Composter*, *Firebase* Real*time Database*, dan aplikasi *monitoring* berjalan dengan responsif, sehingga informasi kondisi kompos dapat diterima pengguna tanpa mengalami keterlambatan yang berarti.

Tabel 4.5 Hasil Pengujian *Jitter*

No	*Jitter* (ms)	Rata-rata *Jitter* (ms)	Kategori
1	-	4,81	Bagus
2	3.23		
3	3.71		
4	0.68		
5	6.30		
6	3.34		
7	8.94		
8	5.79		
9	1.42		
10	8.75		
11	11.34		
12	6.18		
13	7.70		
14	1.30		
15	1.26		
16	10.84		
17	10.57		
18	0.08		

© 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

No	*Jitter* (ms)	Rata-rata *Jitter* (ms)	Kategori
19	3.52	4,81	Bagus
20	4.61		
21	3.88		
22	6.30		
23	6.01		
24	2.10		
25	4.20		
26	5.93		
27	2.56		
28	2.03		
29	2.23		
30	4.80		

Berdasarkan tabel 4.5, hasil pengujian *jitter* yang dilakukan sebanyak 30 kali, diperoleh nilai *jitter* berkisar antara 0,08 ms hingga 11,34 ms, dengan rata-rata *jitter* sebesar 4,81 ms. Nilai tersebut menunjukkan bahwa variasi waktu kedatangan paket data selama proses komunikasi relatif kecil sehingga pengiriman data berlangsung secara stabil dan konsisten. Berdasarkan standar TIPHON, rata-rata *jitter* sebesar 4,81 ms termasuk dalam kategori Bagus, yang mengindikasikan bahwa fluktuasi *delay* antar paket tidak signifikan dan tidak memengaruhi kinerja sistem secara berarti. Dengan demikian, komunikasi data antara perangkat *Smart Composter*, *Firebase* Real*time Database*, dan aplikasi *monitoring* memiliki tingkat kestabilan yang baik, sehingga data sensor dapat diterima secara berurutan dan andal untuk mendukung proses pemantauan kompos secara real-*time*.

Tabel 4.6 Hasil Pengujian *Throughput*

No	Jumlah Data (B*yte*)	Waktu Pengiriman (s)	*Throughput* (Kbps)	Kategori
1	75757	31,678	19	Sangat Buruk
2	78343	31,869	19	Sangat Buruk
3	87095	31,792	21	Sangat Buruk
4	84373	31,648	21	Sangat Buruk

© 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

No	Jumlah Data (B*yte*)	Waktu Pengiriman (s)	*Throughput* (Kbps)	Kategori
5	86999	31,696	21	Sangat Buruk
6	88655	31,922	22	Sangat Buruk
7	84149	31,765	21	Sangat Buruk
8	85595	31,699	21	Sangat Buruk
9	83897	31,767	21	Sangat Buruk
10	118150	36,200	26	Sangat Buruk
11	103619	32,946	25	Sangat Buruk
12	84353	31,676	21	Sangat Buruk
13	90419	31,837	22	Sangat Buruk
14	87131	31,667	22	Sangat Buruk
15	89825	31,982	22	Sangat Buruk
16	83567	31,545	21	Sangat Buruk
17	88861	31,591	22	Sangat Buruk
18	90245	31,647	22	Sangat Buruk
19	91469	32,549	22	Sangat Buruk
20	88859	31,858	22	Sangat Buruk
21	93353	31,857	23	Sangat Buruk
22	93143	32,266	23	Sangat Buruk
23	90233	31,652	22	Sangat Buruk
24	87023	31,711	21	Sangat Buruk
25	90365	31,844	22	Sangat Buruk
26	88181	32,169	21	Sangat Buruk
27	91463	33,366	21	Sangat Buruk
28	87233	31,641	22	Sangat Buruk
29	84005	31,504	21	Sangat Buruk
30	92494	31,591	23	Sangat Buruk

Berdasarkan tabel 4.6, hasil pengujian *throughput* yang dilakukan sebanyak 30 kali, diperoleh nilai *throughput* berkisar antara 19 Kbps hingga 26 Kbps, dengan rata-rata *throughput* sebesar 21,7 Kbps. Nilai *throughput* yang relatif konsisten menunjukkan bahwa proses pengiriman data dari perangkat *Smart Composter* menuju *Firebase* Real*time Database* dan aplikasi *monitoring* berlangsung secara

© 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

stabil selama pengujian. Berdasarkan standar TIPHON, rata-rata *throughput* tersebut termasuk dalam kategori Sangat Buruk karena nilai *throughput* yang dihasilkan masih berada pada rentang yang rendah. Namun, hasil ini dipengaruhi oleh karakteristik *Smart Composter* yang hanya mengirimkan data sensor berukuran kecil secara periodik (setiap 30 detik), sehingga kebutuhan bandwidth memang rendah dan tidak dirancang untuk transfer data berkapasitas besar. Oleh karena itu, meskipun nilai *throughput* tergolong rendah menurut standar TIPHON, kondisi tersebut tidak memengaruhi kinerja sistem karena seluruh data sensor tetap berhasil dikirim dan diterima dengan baik untuk mendukung proses *monitoring* kompos secara real-*time*.

Tabel 4.7 Hasil Pengujian *Packet loss*

No	*Packet* Dikirim	Paket A-B	*Packet* B-A	*Packet* Diterima	*Packet loss* (%)	Kategori
1	247	130	117	247	0	Sangat Baik
2	245	132	113	245	0	Sangat Baik
3	306	159	147	306	0	Sangat Baik
4	306	160	146	306	0	Sangat Baik
5	305	162	143	305	0	Sangat Baik
6	309	162	147	309	0	Sangat Baik
7	303	160	143	303	0	Sangat Baik
8	304	158	146	304	0	Sangat Baik
9	299	159	140	299	0	Sangat Baik
10	353	183	170	353	0	Sangat Baik
11	329	173	156	329	0	Sangat Baik
12	306	158	148	306	0	Sangat Baik
13	315	162	153	315	0	Sangat Baik
14	307	160	147	307	0	Sangat Baik
15	306	163	143	306	0	Sangat Baik
16	294	157	137	294	0	Sangat Baik
17	312	163	149	312	0	Sangat Baik
18	312	163	149	312	0	Sangat Baik
19	310	164	146	310	0	Sangat Baik
20	312	161	151	312	0	Sangat Baik

© 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

No	*Packet* Dikirim	Paket A-B	*Packet* B-A	*Packet* Diterima	*Packet loss* (%)	Kategori
21	318	165	153	318	0	Sangat Baik
22	315	163	152	315	0	Sangat Baik
23	312	162	150	312	0	Sangat Baik
24	305	160	145	305	0	Sangat Baik
25	314	164	150	314	0	Sangat Baik
26	302	162	140	302	0	Sangat Baik
27	310	165	145	310	0	Sangat Baik
28	308	162	146	308	0	Sangat Baik
29	301	158	143	301	0	Sangat Baik
30	325	166	159	325	0	Sangat Baik

Berdasarkan tabel 4.7, hasil pengujian *packet loss* yang dilakukan sebanyak 30 kali, diperoleh jumlah paket yang diterima selalu sama dengan jumlah paket yang dikirim sehingga menghasilkan *packet loss* sebesar 0% pada seluruh pengujian. Berdasarkan standar TIPHON, nilai *packet loss* sebesar 0% termasuk dalam kategori Sangat Baik, yang menunjukkan bahwa tidak terdapat paket data yang hilang selama proses komunikasi antara perangkat *Smart Composter*, *Firebase* Real*time Database*, dan aplikasi *monitoring*. Hasil ini mengindikasikan bahwa mekanisme pengiriman data berjalan dengan sangat andal serta mampu menjaga integritas data yang dikirimkan. Dengan demikian, sistem memiliki tingkat keandalan komunikasi yang tinggi sehingga seluruh informasi kondisi kompos dapat diterima secara utuh dan akurat untuk mendukung proses *monitoring* secara real-*time*.

4.4.3.3 Hasil Pengujian *Reliability Server*

Hasil pengujian *reliability* digunakan untuk menilai tingkat keandalan sistem dalam menjalankan proses dan layanan yang tersedia. Pengujian ini dilakukan untuk mengetahui kemampuan sistem dalam beroperasi secara stabil, meminimalkan terjadinya kegagalan, serta menjaga konsistensi kinerja selama digunakan. Adapun hasil pengujian *reliability* yang diperoleh ditunjukkan pada Tabel 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

Tabel 4.8 Hasil Pengujian *Reliability Server*

No	Jumlah Data (data)	*Availability* (%)	*Success Rate* (%)	*Failure rate* (%)	Integritas Data (%)
1	120	100	100	0	100
2	240	100	100	0	100
3	360	100	100	0	100

Berdasarkan Tabel 4.8, peningkatan durasi pengujian dari 1 jam sampai 3 jam tidak memengaruhi kinerja *backend server*. Seluruh data berhasil diterima dan disimpan tanpa kehilangan data maupun kegagalan proses penyimpanan. Hal ini menunjukkan bahwa *Firebase* Real*time Database* memiliki kemampuan yang baik dalam menangani pengiriman data secara kontinu dalam jangka waktu yang lebih panjang.

4.4.4 Analisis Data / Evaluasi Pengujian

Berdasarkan hasil pengujian yang telah dilakukan, *Smart Composter* yang dikembangkan mampu menjalankan fungsi *Smart Composter*, pengendalian aktuator, serta pengelolaan data secara baik dan sesuai dengan kebutuhan sistem.

Hasil pengujian *Black box Testing* menunjukkan bahwa seluruh 22 skenario pengujian memperoleh status sukses. Pengujian meliputi fitur autentikasi pengguna, registrasi akun, verifikasi email, reset *password*, pemilihan bin kompos, *monitoring* data sensor, *remote* mesin pencacah, pengelolaan siklus kompos, penyimpanan log kompos, notifikasi, dan *logout*. Seluruh fitur menghasilkan keluaran yang sesuai dengan hasil yang diharapkan sehingga dapat disimpulkan bahwa kebutuhan fungsional sistem telah berhasil diimplementasikan dengan baik dan sistem dapat digunakan sesuai tujuan yang dirancang.

Berdasarkan pengujian *Software Quality Assurance* (SQA) menggunakan standar ISO/IEC 25010, diperoleh nilai *usability* sebesar 77,5%. Nilai tersebut menunjukkan bahwa aplikasi berada pada kategori baik dan dapat diterima oleh pengguna. Hasil ini menunjukkan bahwa pengguna mampu memahami fungsi sistem, mengakses fitur-fitur yang tersedia, serta memperoleh informasi yang dibutuhkan dengan cukup mudah. Dengan demikian, aplikasi yang dikembangkan telah memenuhi aspek kemudahan penggunaan dan memberikan pengalaman pengguna yang baik.

© 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

Hasil pengujian *Quality of Service* (QoS) menunjukkan bahwa komunikasi data antara perangkat IoT, *Firebase* Real*time Database*, dan aplikasi *monitoring* berjalan dengan baik. Pada pengujian perubahan data sensor diperoleh rata-rata *delay* sebesar 48,27 ms dengan kategori sangat baik, *jitter* sebesar 4,81 ms dengan kategori baik, dan *packet loss* sebesar 0% dengan kategori sangat baik. Hasil tersebut menunjukkan bahwa data sensor dapat dikirim dan diterima secara cepat serta stabil tanpa kehilangan paket data.

Selanjutnya, hasil pengujian *reliability server* menunjukkan bahwa *backend server* yang dibangun menggunakan *Firebase* Real*time Database* memiliki tingkat keandalan yang sangat baik. Pengujian *reliability* dilakukan dalam tiga skenario, yaitu selama 1 jam, 2 jam, dan 3 jam. Pada ketiga skenario tersebut diperoleh *availability* sebesar 100%, *success rate* sebesar 100%, *failure rate* sebesar 0%, dan integritas data sebesar 100%. Hasil ini menunjukkan bahwa *Firebase* Real*time Database* mampu mempertahankan keandalan layanan meskipun jumlah data yang dikirim meningkat seiring bertambahnya durasi pengujian. Nilai integritas data sebesar 100% juga menunjukkan bahwa seluruh data yang dikirim oleh perangkat berhasil tersimpan dengan lengkap dan sesuai tanpa mengalami perubahan maupun kehilangan data. Hasil ini membuktikan bahwa *backend server* mampu mendukung proses *Smart Composter* secara stabil, konsisten, dan *real-time* selama sistem beroperasi.

Secara keseluruhan, hasil pengujian *Black box Testing*, ISO/IEC 25010, *Quality of Service* (QoS), dan *reliability server* menunjukkan bahwa sistem *Smart Composter* yang dikembangkan telah memenuhi kebutuhan fungsional, memiliki tingkat kemudahan penggunaan yang baik, mampu menyediakan komunikasi data yang stabil, serta didukung oleh *backend server* yang andal. Dengan demikian, sistem dinilai layak untuk digunakan sebagai solusi *Smart Composter*.

© 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 V
PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat diperoleh beberapa kesimpulan sebagai berikut:

1. *Backend server Smart Composter* berbasis *Firebase* berhasil dirancang dan diimplementasikan menggunakan arsitektur *Client Server* dengan memanfaatkan layanan *Firebase Authentication*, *Firebase* Real*time Database*, *Firebase Cloud Messaging* (FCM), *Firebase Cloud Functions*, dan *Firebase Hosting*.
2. Integrasi layanan *Firebase* berhasil mendukung proses autentikasi pengguna, penyimpanan data *monitoring*, sinkronisasi data secara *real-time*, pengelolaan riwayat pengomposan, layanan *monitoring* berbasis *Website*, serta pengiriman notifikasi otomatis kepada pengguna.
3. Hasil pengujian menunjukkan bahwa *backend server* yang dikembangkan mampu beroperasi dengan baik. Tingkat keberhasilan pengujian *Black box* sebesar 100%, nilai *usability* mencapai 77,5% dengan kategori baik, nilai *packet loss* sebesar 0%, serta *reliability server* menunjukkan *availability* sebesar 100%, *success rate* sebesar 100%, *failure rate* sebesar 0%, dan integritas data sebesar 100%. Hasil tersebut menunjukkan bahwa *backend server Firebase* mampu mendukung proses penyimpanan, sinkronisasi, dan distribusi data *Smart Composter* secara andal.

5.2 Saran

1. Melakukan pengujian *backend server* pada skenario multi-*node* dengan melibatkan beberapa unit *Smart Composter* untuk mengevaluasi performa sinkronisasi data, penggunaan sumber daya, dan skalabilitas sistem.
2. Menambahkan fitur analisis data historis dan visualisasi tren pengomposan untuk mendukung proses evaluasi dan pengambilan keputusan berbasis data.
3. Melakukan pengujian performa *backend server* pada skala yang lebih besar dengan jumlah pengguna dan volume data yang lebih banyak untuk mengevaluasi kemampuan skalabilitas 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR PUSTAKA

Kementerian Lingkungan Hidup dan Kehutanan Republik Indonesia (2025), *Sistem Informasi Pengelolaan Sampah Nasional. Available at: https://sampahnasional.kemenlh.go.id/#data- section*

Pillai, S.M., 2023. *Teknologi pengomposan pantas untuk pengurusan sisa pertanian (Rapid composting technology for agricultural waste management)*. Buletin Teknologi MARDI, 35, pp.85–90. Available at: http://ebuletin.mardi.gov.my/buletin/35/Shashikala.pdf

Prayoga, M.W. and Yudistira, B.G.K., 2025. Sistem *Monitoring* dan Pengelolaan Sampah Berbasis IoT dan *Cloud* Menggunakan *Google Firebase* pada Tempat Sampah Pintar. *Jurnal Informatika dan Teknik Elektro Terapan (JITET)*, 13(3), pp.1133–1141. Available at: https://doi.org/10.23960/jitet.v13i3.7175

Alsayaydeh, J.A.J., Bacarra, R., Abd Gani, S.F., Mamchenko, S. and Herawan, S.G., 2025. *FoodSharePro: An Integrated Mobile Platform for Sustainable Food Donation and Decentralized Composting*. International Journal *of* Advanced Computer Science and Applications (IJACSA), 16(7), pp.106–122. Available at: https://doi.org/10.14569/IJACSA.2025.0160711

Setiawan, P.R., Ramadhan, R.A. and Labellapansa, A. (2022) 'Pelatihan Pemrograman *Flutter*', *Jurnal Pengabdian Masyarakat dan Penerapan Ilmu Pengetahuan (JPMPI)*, 3 (1), pp. 22–27. Available at: https://doi.org/10.25299/jpmpip.2022.10699

Ahmad, F. and Sutabri, T. (2024) 'Implementasi Metode *Task Based Language Learning* untuk Meningkatkan Kompetensi Bahasa Inggris Berbasis Android', *Router: Jurnal Teknik Informatika dan Terapan*, 2(4), pp. 13–27. Available at: https://doi.org/10.62951/router.v2i4.269

Saraf, P.R., Jadhao, S.M., Wanjari, S.J., Kolwate, S.G. and Patil, A.D. (2022) 'A Review on *Firebase* (*Backend* as a *Service*) for *Mobile* Application *Development*', *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, 10(1), pp. 967–971. Available at: https://doi.org/10.22214/ijraset.2022.39958

© 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

Chougale, P., Yadav, V. and Gaikwad, A. (2021) '*Firebase* – Overview and Usage', *International Research Journal of Modernization in Engineering Technology and Science*, 3(12), pp. 1178– 1183. Available at: https://www.researchgate.net/publication/362539877_Firebase_Overview_And_Usage

Martin, R.S. and Dewanto, Y. (2023) 'Prototipe kunci pintu otomatis menggunakan sensor kamera berbasis Raspberry', *Jurnal Teknologi Industri*, 12(1), pp. 21–29. Available at: https://journal.universitassuryadarma.ac.id/index.php/jti/article/view/1044

Wahyudi, D., Mujiono, M., & Fu'ad, M. N. (2024). *Analisis Quality of Service (QoS) pada Jaringan Nirkabel di Akademi Komunitas Negeri Putra Sang Fajar Blitar Berdasarkan Standar Parameter TIPHON: Studi Kasus Program Studi Administrasi Server dan Jaringan Komputer*. JAMI: Jurnal Ahli Muda Indonesia, 5(2), 164–171. Available at: https://doi.org/10.46510/jami.v5i2.343

Aswaldi, H. (2025) 'Penerapan teknologi internet *of things* (IoT) untuk *monitoring* kualitas udara dalam ruangan', *Journal of Computer Science and Information Technology*, 1(2), pp. 39–45. Available at: https://doi.org/10.70716/jocsit.v1i2.255

Gusdevi, H., Hadhiwibowo, A., Agustina, N., Fatah, A. and Naseer, M. (2023) 'Timbangan berbasis IoT untuk pemantauan dan pengelolaan sampah organik pada smart waste *management* di Desa Manyingsal', *NARATIF: Jurnal Ilmiah Nasional Riset Aplikasi dan Teknik Informatika*, 5(2), pp. 162–170. Available at: https://naratif.utb-univ.ac.id/index.php/naratif/article/view/270/105

Santosa, R., Sari, P.A. and Sasongko, A.T. (2023) 'Sistem *monitoring* suhu dan kelembaban berbasis IoT (Internet *of* Thing) pada gudang penyimpanan PT Sakafarma Laboratories', *Jurnal Teknologi dan Sistem Informasi Bisnis*, 5(4), pp. 391–400. Available at: https://pdfs.semanticscholar.org/4289/46b30653a94be99328952818252d

© 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

8dd96c0d.pdf

Waruwu, M. (2024) 'Metode Penelitian dan Pengembangan (R&D): Konsep, Jenis, Tahapan dan Kelebihan', *Jurnal Ilmiah Profesi Pendidikan*, 9(2), pp. 1220–1230. Available at: https://doi.org/10.29303/jipp.v9i2.2141

Aziz, F.A., Subandri, M.A. and Armada, 2025. *Penerapan REST API pada Aplikasi Antarmuka Alat Pemantauan Tambak Udang*. Journal *of* Applied Computer Science and Technology (JACOST), 6(2), pp.82–91. Available at: https://doi.org/10.52158/19qkf837

Rezeki, T.P. and Nasution, M.I.P., 2025. *Integrasi Data Base Berbasis Cloud untuk Skalabilitas Bisnis: Studi Kasus Netflix*. Jurnal Sains Student *Research*, 3(3), pp.328–340. Available at: https://ejurnal.kampusakademik.co.id/index.php/jssr/article/view/4743/4166

Syaputra, A. And Ratnasari, R., 2025. *Cloud-Based Healthcare: Inovasi Sistem Informasi Mobile dengan Firebase sebagai Cloud Database*. Jurnal Rekayasa Perangkat Lunak, 4(1). Available at: https://journal.aisyahuniversity.ac.id/index.php/J-Rapa

Nyabuto, G.M., Mony, V. and Mbugua, S. (2024) 'Architectural Review *of Client Server* Models', *International Journal of Scientific Research & Engineering Trends*, 10(1), pp. 139–143. Available at:

Pratama, R., Siambaton, M.Z. and Haramaini, T., 2022. *Implementasi Internet of Things pada Sistem Kendali Suhu dan Kelembaban Rak Server Berbasis Mikrokontroler*. Sudo: Jurnal Teknik Informatika, pp.145–153. Available at: https://doi.org/10.56211/sudo.v1i4.129

Nynäs, M. 2024. *Website Development with Firebase: Online Yatzy Game*. Thesis. Centria University *of* Applied Sciences, Information Technology. https://urn.fi/URN:NBN:fi:amk-2024051512112

Amanuel, A., 2022. *Supabase vs Firebase: Evaluation of Performance and Development of Progressive Website Apps*. Bachelor's Thesis, Information Technology. Metropolia University *of* Applied Sciences, Helsinki, Finland.

© 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

https://www.theseus.fi/bitstream/handle/10024/771009/Ayezabu_Amanuel.pdf?sequence=2&isAllowed=y

Budiman, A.S. and Parhusip, J., 2026. *Analisis kualitas dan penerapan software quality assurance menggunakan black box Testing pada aplikasi Website Explore Palangka Raya*. JATI (Jurnal Mahasiswa Teknik Informatika), 10(1), pp.65-69. https://www.ejournal.itn.ac.id/jati/article/view/16610/8800

Bayramova, T.A. and Malikova, N.C., 2024. *Developing a conceptual model for improving the software system reliability*. Problems *of* Information Society, 15(1), pp.42–56. Available at: http://doi.org/10.25045/jpis.v15.i1.05

Google. (2026). *Understand Firebase Realtime Database Security Rules*. *Firebase* Documentation. https://*Firebase.google*.com/docs/*database/security*

Google. (2026). *Firebase IAM permissions*. *Firebase* Documentation. https://*Firebase.google*.com/docs/*projects*/iam/permissions

Diharja, S.P., Arimbawa, I.W.A. and Zubaidi, A., 2025. *Rancang Bangun Sistem Monitoring Proses Pembuatan Kompos dari Sampah Rumah Tangga Berbasis Internet of Things (IoT)*. Jurnal Teknologi Informasi, Komputer dan Aplikasinya (JTIKA), 7(1), pp.1–12. Available at https://jtika.if.unram.ac.id/index.php/JTIKA/article/view/388/177

Diannita, R., Muslih, M., Yahya, Y.K., Ma'ruf, A., Arrosi, M. and Fatayati, U., 2025. *Kompos Digital: Revolusi Pengelolaan Sampah Berbasis Teknologi*. Prosiding Seminar Nasional Pengabdian kepada Masyarakat (SNPPM 2025), pp.1–13. Available at https://journal.unj.ac.id/unj/index.php/snppm/article/view/61035/21591

Aprizal, M. and Wagyana, A., 2025. *Rancang Bangun Sistem Komposter Otomatis Berbasis IoT dengan Pemantauan Suhu, Kelembapan, dan pH secara Real Time*. SPEKTRAL: Journal *of* Communications, Antennas and Propagation, 6(2), pp.345–351. Available at https://jurnal.pnj.ac.id/index.php/spektral/article/view/7877/3773

Bachri, S.F.R., 2022. *Implementasi Protocol LoRaWAN pada Wireless Sensor Network untuk Sistem Kompos Pintar dengan Modul Komunikasi LoRa*.

© 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

Skripsi. Universitas Islam Negeri Syarif Hidayatullah Jakarta. Available at https://repository.uinjkt.ac.id/dspace/handle/123456789/65285

Gurla Venkata Kameswari, S., Basavaraju, A., Kumar, C.S. and Bapat, J., 2025. *Compost Monitoring System for Kitchen Waste Management: Development, Deployment and Analysis*. IoT, 6, p.64. available at https://doi.org/10.3390/iot6040064

© Hak Cipta ... ri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, 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 RIWAYAT HIDUP

Saya Adipraja Putra, lahir di Depok pada 30 Oktober 2004. Saat ini saya berdomisili di Depok, Jawa Barat. Saya menempuh pendidikan dasar di SD Negeri Pondokcina 5 pada tahun 2010-2016, kemudian melanjutkan pendidikan di SMP Negeri 8 pada tahun 2016–2019 dan SMA Muhammadiyah 2 Beji Depok, pada tahun 2019–2022. Setelah lulus SMA, saya melanjutkan pendidikan pada Program Studi Teknik Mul*time*dia dan Jaringan sejak tahun 2022.

© 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

LAMPIRAN

Lampiran 1. Surat Izin Observasi

KEMENTERIAN PENDIDIKAN TINGGI, SAINS, DAN TEKNOLOGI
POLITEKNIK NEGERI JAKARTA
Jalan Prof. Dr. G. A.Siwabessy, Kampus UI, Depok 16425
Telepon (021) 7270036, Hunting, Fax (021) 7270034
Laman: http://www.pnj.ac.id Posel: humas@pnj.ac.id

Nomor : 51/DST/PL3.A.9/B/KM.07.00/2026 26 Juni 2026
Perihal : Permohonan Izin Observasi

Kepada Yth.
Kepala Bank Sampah Rumah Harum
Jl. Merdeka No.1, RT.05/RW.01, Abadijaya, Kec. Sukmajaya, Kota Depok, Jawa Barat 16411

Dengan hormat,
Sehubungan dengan mata kuliah Skripsi yang dilaksanakan pada semester 8 (delapan) Program Studi Teknik Multimedia dan Jaringan Jurusan Teknik Informatika dan Komputer Politeknik Negeri Jakarta. Dengan ini kami mohon kesediaan Bapak/Ibu agar dapat mengizinkan mahasiswa kami untuk melakukan observasi di Bank Sampah Rumah Harum dengan judul penelitian "RANCANG BANGUN SMART COMPOSTER BERBASIS IOT DAN MACHINE LEARNING DENGAN SISTEM MONITORING BERBASIS FIREBASE DAN FLUTTER ". Tugas mata kuliah ini bertujuan untuk menambah wawasan terkait dengan aplikasi teori yang sudah dipelajari di Kampus dengan kondisi lapangan sebagai wadah pembelajaran dan penambah informasi mengenai mata kuliah tersebut. Adapun berikut adalah nama mahasiswa kami:

No.	Nama dan Nim	Semester/ Program Studi	Nomor Telepon dan Email	Tujuan Observasi
1	Adipraja Putra Ramadhan 2207421022	8 / Teknik Multimedia dan Jaringan	0852-1556-9215 adipraja.putra.ramadhan.tik22@mhsw.pnj.ac.id	Melaksanakan observasi dan wawancara untuk memperoleh data mengenai pengelolaan sampah organik, kebutuhan pengguna, dan alur operasional yang akan digunakan sebagai studi kasus dalam penelitian dan pengembangan Smart Composter
2	Fawwaz Hibatul Wafi 2207421025		0896-0877-9892 fawwaz.hibatul.wafi.tik22@mhsw.pnj.ac.id	
3	Rasyid Fathoni Syahputra 2207421027		0897-6583-287 rasyid.fathoni.syahputra.tik22@mhsw.pnj.ac.id	

Demikian surat ini kami buat, atas perhatian dan kerjasama Bapak/Ibu kami ucapkan terima kasih.

Ketua Jurusan,

Dr. Anita Hidayati, S.Kom., M.Kom.
NIP. 197908032003122003

Tembusan :
1. Direktur;
2. Wakil Direktur Bidang Akademik;
3. Kepala Bagian Keuangan dan Umum
4. Kasubbag. Umum 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

Lampiran 2. Wawancara

No	Pertanyaan	Jawaban
2	Apakah Bapak/Ibu merasa terbantu jika kondisi kompos dapat dipantau melalui aplikasi di HP?	Iya, akan lebih praktis karena kami bisa melihat kondisi kompos tanpa harus selalu mengecek secara langsung ke lokasi.
3	Seberapa penting menurut Bapak/Ibu adanya riwayat atau laporan hasil pengomposan?	Cukup penting, karena riwayat tersebut dapat digunakan untuk evaluasi dan mengetahui apakah proses pengomposan berjalan dengan baik atau tidak.
3	Apakah Bapak/Ibu membutuhkan pemberitahuan otomatis ketika kompos sudah matang atau terjadi masalah pada proses pengomposan?	Iya, pemberitahuan seperti itu akan membantu petugas agar bisa segera mengambil tindakan tanpa harus memeriksa secara terus-menerus.
4	Menurut Bapak/Ibu, apakah penyimpanan data secara digital dapat mempermudah pengelolaan kompos dibandingkan pencatatan manual?	Iya, karena data lebih rapi, mudah dicari kembali, dan tidak mudah hilang seperti pencatatan secara manual.
5	Menurut Bapak/Ibu, apakah penyimpanan data secara digital dapat mempermudah pengelolaan kompos dibandingkan pencatatan manual?	Iya, karena data lebih rapi, mudah dicari kembali, dan tidak mudah hilang seperti pencatatan secara manual.

© 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

Lampiran 3. Dokumentasi Wawancara

Lampiran 4. Kecepatan internet saat pengujian QoS

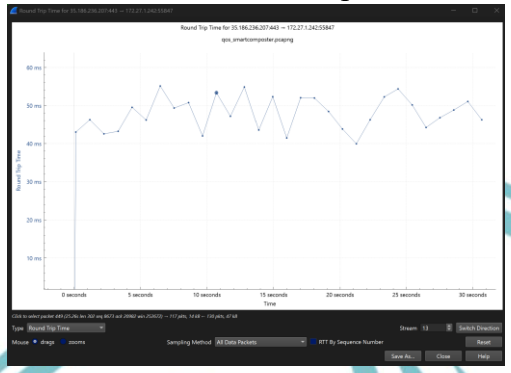
Lampiran 5. Script pengujian QoS

```python
import firebase_admin
from firebase_admin import credentials
from firebase_admin import db

import time
import random

cred = credentials.Certificate("firebase_key.json")

firebase_admin.initialize_app(
    cred,
    {
        "databaseURL":
        "https://aplikasi-kompos-default-rtdb.asia-southeast1.firebasedatabase.app/"
    }
)

current_ref = db.reference("Kompos/Kompos1/Current")

JUMLAH_PAKET = 30

for seq in range(1, JUMLAH_PAKET + 1):

    current_ref.update({
        "Sensor": {
            "Suhu": round(random.uniform(25, 40), 1),
        },
    })

    print(f"Packet {seq} sent")

    time.sleep(1)

print("Selesai")
```

© 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

Lampiran 6. Grafik *Delay* pada Wireshark



Lampiran 7. Data Mentah Pengujian *Delay* Notifikasi

```
3.11
1.88
1.92
2.16
2,64
1.46
1.67
2.29
1.65
1.52
4.33
2.13
2.02
1.55
1.79
1.54
1.63
1.40
1.41
1.87
1.94
1.79
1.59
```

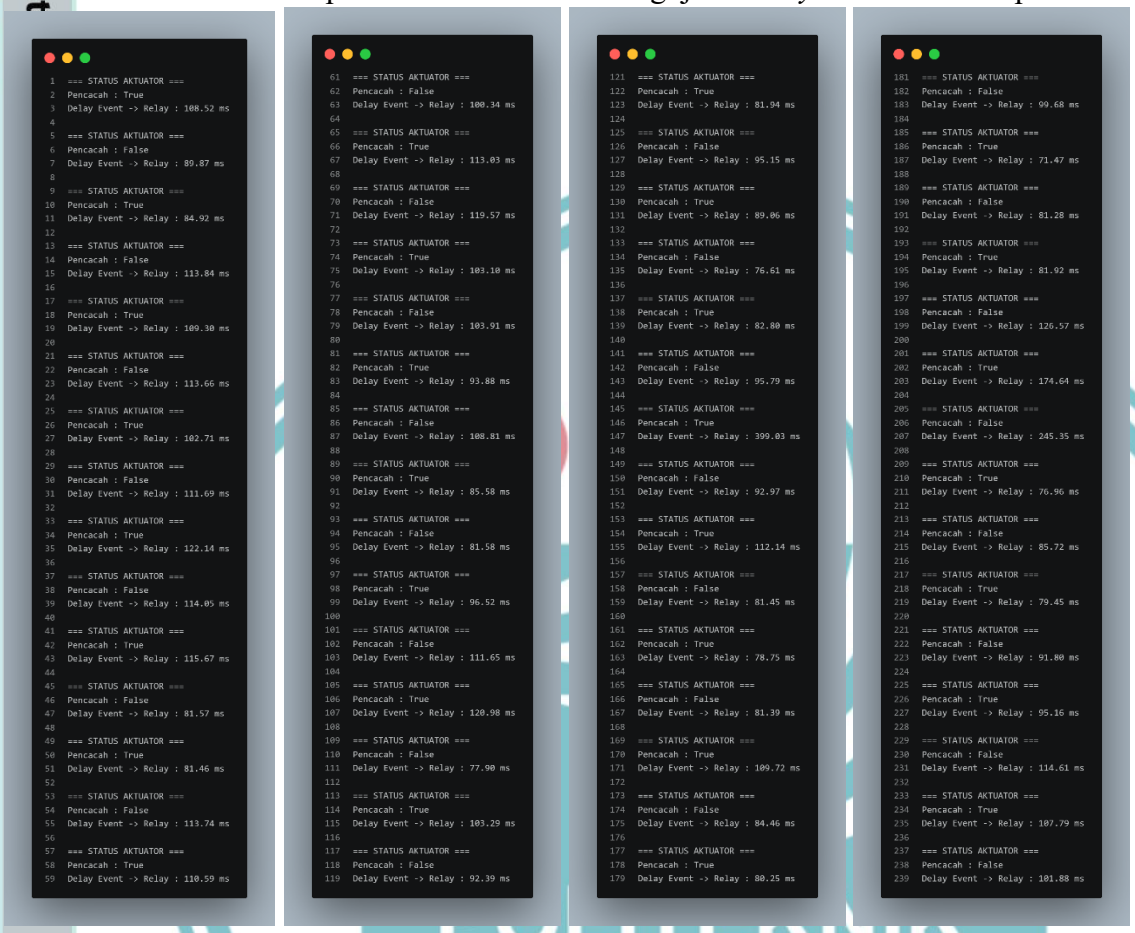
Lampiran 8. Script Pengujian *Remote* Mesin pencacah

© Hak Cipta

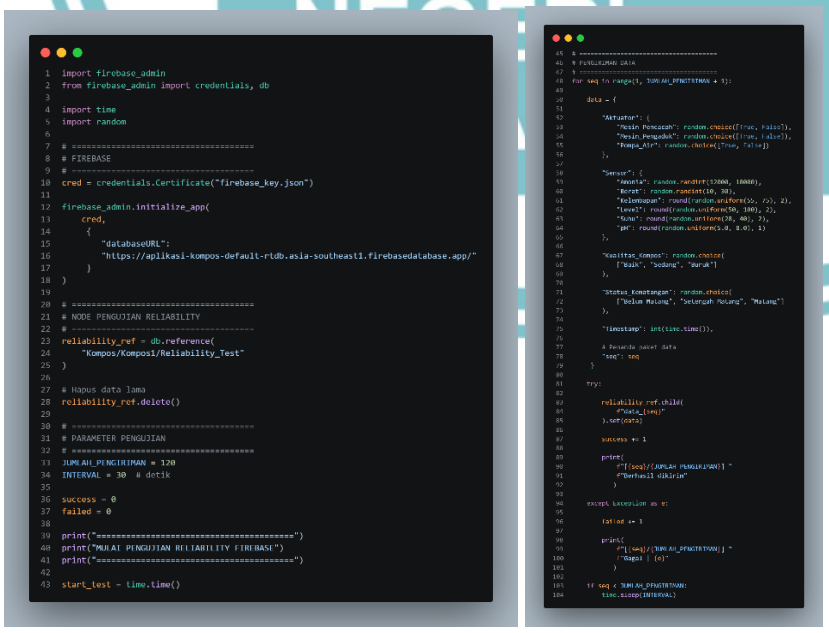
Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
   a. Pengutipan hanya untuk kepentingan pendidikan, 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

Lampiran 9. Data mentah Pengujian *Delay Remote* Mesin pencacah



Lampiran 10. Script Pengujian *Reliability Server*



© Hak Cipta milik Politeknik Negeri Jakarta

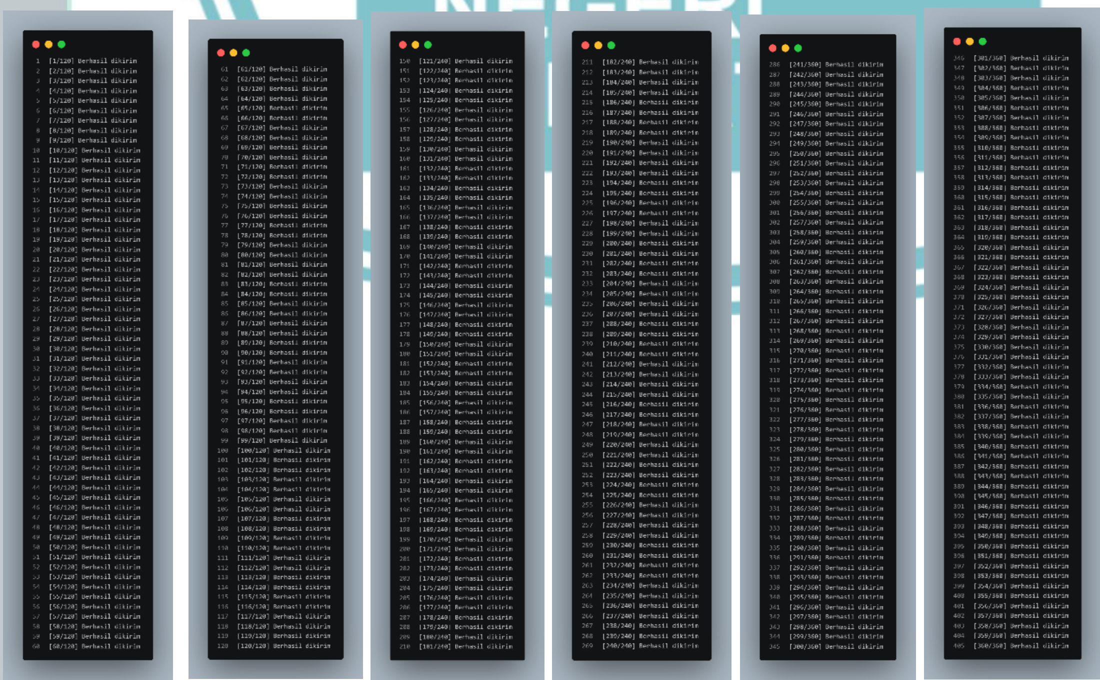
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan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

(Lanjutan)

```python
# ====================================
# SUCCESS & FAILURE RATE
# ------------------------------------
total = success + failed

success_rate = (
    success / total * 100
) if total > 0 else 0

failure_rate = (
    failed / total * 100
) if total > 0 else 0

# ====================================
# CEK DATA DI FIREBASE
# ------------------------------------
firebase_data = reliability_ref.get()

jumlah_data_tersimpan = (
    len(firebase_data)
    if firebase_data
    else 0
)

# ------------------------------------
# CEK INTEGRITAS DATA
# ------------------------------------
integrity_ok = 0

if firebase_data:

    for item in firebase_data.values():

        if (
            "Aktuator" in item
            and "Sensor" in item
            and "Kualitas_Kompos" in item
            and "Status_Kematangan" in item
            and "Timestamp" in item
            and "seq" in item
        ):

            sensor = item["Sensor"]

            if (
                "Amonia" in sensor
                and "Berat" in sensor
                and "Kelembapan" in sensor
                and "Level" in sensor
                and "Suhu" in sensor
                and "pH" in sensor
            ):
                integrity_ok += 1

integrity_rate = (
    integrity_ok /
    JUMLAH_PENGIRIMAN
    * 100
)

# ------------------------------------
# AVAILABILITY
# ------------------------------------
availability = (
    jumlah_data_tersimpan /
    JUMLAH_PENGIRIMAN
    * 100
)

# ------------------------------------
# DURASI PENGUJIAN
# ------------------------------------
end_test = time.time()

durasi_detik = (
    end_test - start_test
)

durasi_jam = (
    durasi_detik / 3600
)
```

```python
# ====================================
# HASIL AKHIR
# ====================================
print("\n====================================")
print("HASIL PENGUJIAN RELIABILITY")
print("====================================")

print(
    f"Durasi Pengujian      : "
    f"{durasi_jam:.2f} jam"
)

print(
    f"Jumlah Pengiriman     : "
    f"{JUMLAH_PENGIRIMAN}"
)

print(
    f"Data Berhasil         : "
    f"{success}"
)

print(
    f"Data Gagal            : "
    f"{failed}"
)

print(
    f"Data Tersimpan        : "
    f"{jumlah_data_tersimpan}"
)

print(
    f"Availability          : "
    f"{availability:.2f}%"
)

print(
    f"Success Rate          : "
    f"{success_rate:.2f}%"
)

print(
    f"Failure Rate          : "
    f"{failure_rate:.2f}%"
)

print(
    f"Integritas Data       : "
    f"{integrity_rate:.2f}%"
)

print("====================================")
```

Lampiran 11. Data Mentah Pengujian *Reliability*



(Lanjutan)

```
HASIL PENGUJIAN RELIABILITY
Durasi Pengujian     : 0.99 jam
Jumlah Pengiriman    : 120
Data Berhasil        : 120
Data Gagal           : 0
Data Tersimpan       : 120
Availability         : 100.00%
Success Rate         : 100.00%
Failure Rate         : 0.00%
Integritas Data      : 100.00%
```

```
HASIL PENGUJIAN RELIABILITY
Durasi Pengujian     : 2.01 jam
Jumlah Pengiriman    : 240
Data Berhasil        : 240
Data Gagal           : 0
Data Tersimpan       : 240
Availability         : 100.00%
Success Rate         : 100.00%
Failure Rate         : 0.00%
Integritas Data      : 100.00%
```

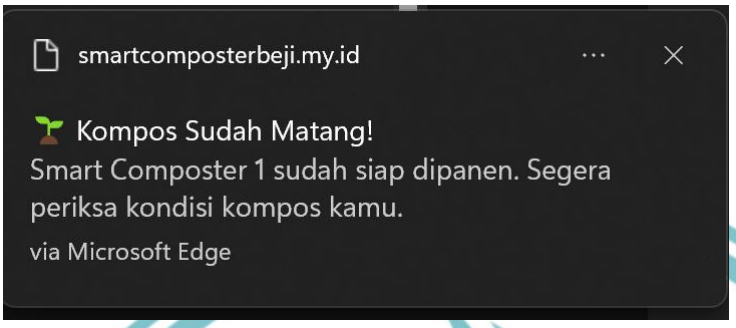
```
HASIL PENGUJIAN RELIABILITY
Durasi Pengujian     : 3.00 jam
Jumlah Pengiriman    : 360
Data Berhasil        : 360
Data Gagal           : 0
Data Tersimpan       : 360
Availability         : 100.00%
Success Rate         : 100.00%
Failure Rate         : 0.00%
Integritas Data      : 100.00%
```

© Hak Cipta milik Politeknik Negeri Jakarta

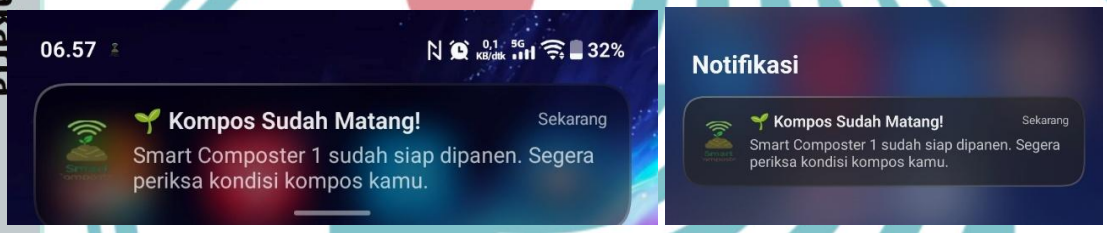
Hak Cipta :

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

Lampiran 12 Tampilan Notifikasi *Website*



Lampiran 13 Tampilan Notifikasi Android



© 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