



**RANCANG BANGUN *SMART DOOR LOCK* BERBASIS
PENGENALAN WAJAH**

Sub Judul:

**IMPLEMENTASI *SMART DOOR LOCK* BERBASIS *IOT*
DENGAN PENDEKATAN *HYBRID EDGE-CLOUD*
*CONTROL***

SKRIPSI

MUHAMMAD HAIKAL FADHILLAH

2207421055

**PROGRAM STUDI TEKNIK MULTIMEDIA DAN JARINGAN
JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER
POLITEKNIK NEGERI JAKARTA**

2026



**IMPLEMENTASI *SMART DOOR LOCK* BERBASIS *IOT*
DENGAN PENDEKATAN *HYBRID EDGE–CLOUD*
*CONTROL***

SKRIPSI

**Diajukan Untuk Melengkapi Syarat-Syarat yang Diperlukan Untuk
Memperoleh Diploma Empat Politeknik**

MUHAMMAD HAIKAL FADHILLAH

2207421055

**PROGRAM STUDI TEKNIK MULTIMEDIA DAN JARINGAN
JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER
POLITEKNIK NEGERI JAKARTA**

2026



Hak Cipta :

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

Nama : Muhammad Haikal Fadhillah
Nim : 2207421055
Jurusan : Teknik Informatika dan Komputer
Program Studi : Teknik Multimedia dan Jaringan
Judul Skripsi : Implementasi *Smart Door Lock* berbasis IoT Dengan Pendekatan *Hybrid Edge-Cloud Control*

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

Apabila dikemudian hari terbukti atau dapat dibuktikan 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, 2 Juli 2026

Yang membuat pernyataan,



Muhammad Haikal Fadhillah

NIM. 2207421055



© 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 : Muhammad Haikal Fadhillah
NIM : 2207421055
Program Studi : Teknik Multimedia dan Jaringan
Judul Skripsi : Implementasi Smart Door Lock Berbasis IoT dengan Pendekatan Hybrid Edge-Cloud Control

Telah diuji oleh tim penguji dalam Sidang Skripsi pada hari Rabu, Tanggal 1, Bulan Juli, Tahun 2026 dan dinyatakan LULUS.

Disahkan oleh

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

()

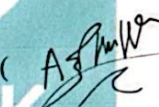
Penguji I : Maria Agustin S.Kom., M.Kom.

()

Penguji II : Susana Dwi Yulianti, S.Kom, M.Kom.

()

Penguji III : Asep Kurniawan, S.Pd., M.Kom.

()

Mengetahui :

Jurusan Teknik Informatika dan Komputer

Ketua

POLITEKNIK
NEGERI
JAKARTA



Dr. Anja Hidayati, S.Kom., M.Kom.

NIP. 197908032003122003



Hak Cipta :

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

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan ke hadirat Allah SWT atas segala rahmat, nikmat, pertolongan, dan kemudahan-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Implementasi *Smart Door Lock* Berbasis *IoT* Dengan Pendekatan *Hybrid Edge-Cloud Control*”. Selama proses penyusunan skripsi ini, penulis mendapatkan banyak dukungan dan bantuan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Allah SWT atas segala kekuatan, petunjuk, dan kemudahan yang diberikan kepada penulis.
2. Kedua orang tua dan keluarga tercinta yang selalu memberikan doa, kasih sayang, dukungan moril maupun materiil, serta senantiasa menjadi motivasi terbesar bagi penulis untuk menyelesaikan pendidikan ini.
3. Bapak Dr. Indra Hermawan, S.Kom., M.Kom. selaku dosen pembimbing yang dengan sabar telah memberikan arahan, bimbingan, masukan berharga, serta kepercayaan dan ruang bagi penulis untuk belajar dan berproses dari awal hingga akhir penyusunan skripsi ini.
4. Dhita Shyva Amara selaku yang senantiasa menemani penulis, memberikan doa, motivasi, keceriaan, serta dukungan yang tak terhingga sepanjang penyusunan skripsi ini.
5. Orang-orang berharga yang telah memberikan nasihat, semangat, dan keyakinan kepada penulis untuk tetap menyelesaikan proses pengerjaan skripsi ini.
6. Seluruh dosen, civitas akademika Politeknik Negeri Jakarta, khususnya Program Studi Teknik Multimedia dan Jaringan, serta teman-teman seperjuangan yang telah memberikan ilmu, pengalaman, bantuan, dan dukungan selama masa perkuliahan hingga penyusunan skripsi ini.

Penulis menyadari bahwa skripsi ini masih memiliki kekurangan. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan. Semoga skripsi ini dapat memberikan manfaat serta kontribusi nyata dalam pengembangan teknologi *Internet of Things*, khususnya pada bidang sistem mekanisme kontrol akses *smart door lock*



© 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

Depok, Juni 2026

Muhammad Haikal Fadhillah





Hak Cipta :

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

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS

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

Nama : Muhammad Haikal Fadhillah
Nim : 2207421055
Jurusan : Teknik Informatika dan Komputer
Program Studi : 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 :

Implementasi Smart Door Lock berbasis IoT Dengan Pendekatan Hybrid Edge-Cloud Control Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksklusif ini Politeknik Negeri Jakarta Berhak menyimpan, mengalihmediakan/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan skripsisaya tanpa meminta izin darisaya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Depok, 27 Juli 2026

Yang Menyatakan,



Muhammad Haikal Fadhillah

NIM. 2207421055



Hak Cipta :

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

ABSTRAK

Penelitian ini merancang dan menganalisis sistem smart door lock berbasis Internet of Things (IoT) menggunakan arsitektur hybrid edge-cloud control untuk mengatasi kelemahan sistem konvensional yang rentan mengalami kegagalan akses dan latensi tinggi saat koneksi internet terputus. Melalui metode Research and Development (R&D), pemrosesan autentikasi wajah dilakukan secara mandiri di tingkat lokal (edge) menggunakan gateway Raspberry Pi 4, sementara manajemen data terpusat dikelola melalui cloud. Integrasi perangkat keras melibatkan ESP32-CAM, node ESP32, RFID, dan keypad, yang didukung oleh transmisi data ESP-NOW serta sinkronisasi cloud menggunakan protokol AMQP melalui broker RabbitMQ. Hasil pengujian menunjukkan tingkat keberhasilan fungsionalitas sebesar seratus persen pada seluruh fitur. Evaluasi kinerja jaringan mencatat stabilitas transmisi citra wajah dengan latensi rendah, serta kemampuan sistem dalam menangani beban seratus koneksi secara bersamaan tanpa adanya data yang hilang (nol persen packet loss). Kesimpulannya, pendekatan hybrid ini terbukti jauh lebih efektif dan andal, memungkinkan eksekusi autentikasi lokal secara cepat tanpa internet (edge autonomy), sekaligus menjamin integritas sinkronisasi log data secara terpusat ketika jaringan kembali terhubung.

Kata Kunci: Smart Door Lock, Internet of Things (IoT), Hybrid Edge-Cloud, Pengenalan Wajah, Quality of Service (QoS).



Hak Cipta :

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

DAFTAR ISI

SURAT PERNYATAAN BEBAS PLAGIARISME	ii
LEMBAR PENGESAHAN	iii
KATA PENGANTAR	iv
SURAT PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS	vi
ABSTRAK	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	x
DAFTAR TABEL	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan dan Manfaat	3
1.4.1 Tujuan	4
1.4.2 Manfaat	4
1.5 Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA	6
2.1 Kajian Pustaka	6
2.1.1 <i>Internet of Things (IoT)</i>	6
2.1.2 ESP32	7
2.1.3 Raspberry Pi	7
2.1.4 ESP32CAM	8
2.1.5 Edge Computing	8
2.1.6 Cloud Computing	9
2.1.7 Arsitektur Hybrid Edge-Cloud Control	9
2.2 Penelitian Terdahulu	11
BAB III METODE PENELITIAN	15
3.1 Rancangan Penelitian	15
3.2 Tahapan Penelitian	16
3.3 Objek Penelitian	17
BAB IV HASIL DAN PEMBAHASAN	19
4.1 Analisis Kebutuhan	19
4.1.1 Kebutuhan Perangkat Keras	19
4.1.2 Kebutuhan Perangkat Lunak	21
4.2 Perancangan Sistem	21
4.2.1 Blok Diagram	22
4.2.2 Arsitektur Sistem	23
4.2.3 Perangkat Keras	25
4.2.4 Sistem Sinkronisasi	28
4.2.5 Proses Autentikasi	29
4.3 Implementasi Sistem	31
4.3.1 Implementasi Sistem Perangkat Lunak	31
4.3.2 Implementasi Sistem Perangkat Keras	48



© 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.....	52
4.4.1	Deskripsi Pengujian	52
4.4.2	Prosedur Pengujian	57
4.4.3	Data Hasil Pengujian.....	58
4.5	Analisis Data	73
BAB V PENUTUP.....		78
5.1	Kesimpulan.....	78
5.2	Saran	79
DAFTAR PUSTAKA		80
DAFTAR RIWAYAT HIDUP.....		82
LAMPIRAN.....		83





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, 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 Konsep <i>IoT</i>	7
Gambar 2.2 Arsitektur <i>hybrid cloud-edge</i>	10
Gambar 4.1 Blok Diagram Sistem	23
Gambar 4.2 Arsitektur Sistem.....	24
Gambar 4.3 Rangkaian elektronik pada <i>gateway</i>	26
Gambar 4.4 Detail rangkaian node.....	27
Gambar 4.5 Proses Sinkronisasi.....	28
Gambar 4.6 Alur Autentikasi Pengguna	30
Gambar 4.7 Mengaktifkan aplikasi <i>gateway</i>	32
Gambar 4.8 Halaman login pada aplikasi <i>gateway</i>	32
Gambar 4.9 Pengaturan API IDE dan API KEY	33
Gambar 4.10 Proses menyimpan API ID dan API KEY	33
Gambar 4.11 Halaman login pada <i>gateway</i>	34
Gambar 4.12 Halaman home aplikasi <i>gateway</i>	34
Gambar 4.13 Authentication daemon	35
Gambar 4.14 Menu sinkronisasi pada aplikasi <i>gateway</i>	35
Gambar 4.15 Menu room	36
Gambar 4.16 Menu “Room” ketika berhasil membuat child node	36
Gambar 4.17 Detail node list setelah pembuatan ruangan.....	37
Gambar 4.18 Proses menabahkan akses wajah pada suatu ruangan	37
Gambar 4.19 Daftar wajah yang memiliki akses ke ruangan.....	38
Gambar 4.20 Menu Enrollment	39
Gambar 4.21 Proses enrolment	40
Gambar 4.22 Proses enrolment berhasil.....	40
Gambar 4.23 Pengecekan layanan sinkronisasi	41
Gambar 4.24 Proses sinkronisasi manual.....	42
Gambar 4.25 proses sinkronisasi berhasil dilakukan	42
Gambar 4.26 Menu ESP Mesh Network.....	43
Gambar 4.27 Detail Network	43
Gambar 4.28 Menu setting.....	44
Gambar 4.29 Proses setup node	45
Gambar 4.30 Halaman konfigurasi node	46
Gambar 4.31 Halaman update firmware	46
Gambar 4.32 Proses bergabung ke jaringan ESP <i>mesh</i>	48
Gambar 4.33 Tampilan <i>gateway</i>	49
Gambar 4.34 Perangkat keras <i>bridge</i>	50
Gambar 4.35 Perangkat keras <i>node</i>	51
Gambar 4.36 Proses pengukuran latency one trip transmission	54
Gambar 4.37 Alur pengukuran <i>full trip data latency</i>	55
Gambar 4.38 Skema pengujian delay	56
Gambar 4.39 Hasil pengujian <i>one trip</i>	63
Gambar 4.40 Hasil pengujian <i>full trip</i>	65
Gambar 4.41 Metode pengujian latency	66



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, 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 4.42 Grafik delay sinkronisasi secara bersamaan.....	68
Gambar 4.43 Proses pengujian <i>Synchronization</i> data integrity	70
Gambar 4.44 Grafik perbandingan delay <i>synchronization</i> data integrity	71
Gambar 4.45 Grafik analisis <i>latency</i> dan <i>jitter</i> pengujian full-trip ESP-NOW	73
Gambar 4.46 Grafik analisis packet loss pengujian full-trip ESP-NOW	74
Gambar 4.47 Grafik analisis performa delay sinkronisasi secara bersamaan	75





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, 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 Terdahulu	11
Tabel 4.1 Spesifikasi Kebutuhan Perangkat Keras	20
Tabel 4.2 Spesifikasi Kebutuhan Perangkat Lunak	21
Tabel 4.3 Tabel prosedur pengujian.....	57
Tabel 4.4 Hasil Pengujian Fungsionalitas	59
Tabel 4.5 Hasil pengujian <i>one trip</i>	62
Tabel 4.6 Hasil pengujian <i>full trip</i>	64
Tabel 4.7 Hasil pengujian <i>Synchronization Delay</i>	67
Tabel 4.8 Hasil pengujian <i>Synchronization Data Integrity</i>	70





Hak Cipta :

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

Perkembangan *Internet of Things* (IoT) dalam beberapa tahun terakhir menunjukkan pertumbuhan yang sangat pesat. Riset industri menunjukkan peningkatan eksponensial dalam adopsi perangkat pintar secara global di berbagai sektor, terutama otomasi industri, perkantoran, dan sistem keamanan fisik terintegrasi. Penerapan IoT pada sistem keamanan menjadi semakin relevan seiring meningkatnya kebutuhan akan pengelolaan akses yang lebih modern, dinamis, dan terintegrasi secara *real-time* (Zein et al., 2024).

Salah satu implementasi IoT yang paling krusial di bidang keamanan fisik adalah sistem kunci pintu pintar (*smart door lock*), yang dikembangkan untuk mengatasi berbagai keterbatasan bawaan dari sistem penguncian mekanis konvensional. Sistem penguncian mekanis konvensional memiliki berbagai kelemahan nyata, seperti risiko kehilangan kunci fisik, kemudahan duplikasi kunci tanpa izin, keterbatasan dalam membagikan hak akses secara dinamis bagi banyak pengguna, serta ketiadaan pencatatan log aktivitas keluar masuk secara otomatis (Murjitama et al., 2024).

Beberapa penelitian terdahulu telah berupaya mengembangkan sistem pintu pintar, namun masih menyisakan sejumlah kelemahan. Sistem yang memproses autentikasi wajah secara lokal di tingkat *edge* tanpa pembagian komputasi yang jelas dengan *cloud* sering kali terkendala dalam pencatatan log terpusat dan penyimpanan jangka panjang (Sopandi & Hendrawati, 2023; Syarifullah et al., 2023). Sebaliknya, sistem yang memindahkan seluruh proses pengenalan wajah ke server *cloud* sangat rentan terhadap kegagalan autentikasi saat jaringan terputus serta memiliki latensi respons yang tinggi (Putri et al., 2025). Di sisi lain, integrasi layanan awan komersial berskala besar sering kali meningkatkan kompleksitas arsitektur sistem (Kundang & Kadek, 2021), sementara sistem lokal berbasis jaringan mesh yang ada saat ini masih terbatas pada autentikasi kartu RFID tanpa menyertakan teknologi biometrik seperti pengenalan wajah (Fachrudin, 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

Meskipun sistem *smart door lock* berbasis IoT telah banyak dikembangkan, sebagian besar solusi saat ini masih sangat bergantung pada arsitektur *cloud computing* secara penuh. Ketergantungan penuh pada *cloud* ini menimbulkan masalah keandalan yang besar ketika koneksi internet mengalami gangguan atau tidak stabil. Pada kondisi tersebut, sistem tidak dapat memproses autentikasi atau bahkan mengalami kegagalan fungsi. Selain itu, pengiriman data berukuran besar seperti citra wajah dari pintu ke server *cloud* di luar jaringan lokal sering kali terkendala latensi yang tinggi dan kerentanan terhadap kehilangan paket data (Barus et al., 2024).

Sebagai solusi untuk mengatasi kendala tersebut, konsep *edge computing* diterapkan untuk memindahkan pemrosesan data ke dekat sumber data (lokal) guna meminimalkan latensi dan meningkatkan kecepatan respons. Namun, pemrosesan di sisi *edge* saja memiliki keterbatasan dalam kapasitas penyimpanan data jangka panjang serta kemudahan manajemen sistem secara terpusat. Oleh karena itu, arsitektur *hybrid edge–cloud control* menawarkan solusi dengan mengombinasikan keunggulan pemrosesan lokal pada sisi *edge* dengan fleksibilitas pengelolaan dan monitoring data secara terpusat pada sisi *cloud*.

Di lingkungan Politeknik Negeri Jakarta, yang memiliki aktivitas akademis tinggi serta fasilitas berupa laboratorium dan ruang kelas yang tersebar di banyak gedung bertingkat, diperlukan suatu sistem kontrol akses pintu yang andal, responsif, dan mudah dikelola secara terpusat. Berdasarkan kondisi tersebut, penelitian ini bertujuan untuk merancang dan membangun sistem *smart door lock* berbasis IoT dengan pendekatan *hybrid edge–cloud control*. Sistem ini memanfaatkan ESP32 sebagai *node*, Raspberry Pi sebagai *gateway* tingkat *edge*, metode autentikasi berbasis pengenalan wajah, RFID, dan PIN, serta menggunakan media komunikasi nirkabel *ESP Mesh Network* dan *ESP NOW* untuk menjangkau area gedung secara luas dan meminimalkan area tanpa sinyal (*dead zone*). Sistem ini diharapkan mampu meningkatkan keandalan operasional, keamanan fisik, dan efisiensi pengelolaan akses di lingkungan kampus.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka permasalahan yang dapat



Hak Cipta :

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

dirumuskan dalam penelitian ini adalah sebagai berikut:

1. Bagaimana merancang sistem *smart door lock* berbasis *Internet of Things (IoT)* dengan pendekatan *hybrid edge–cloud control*?
2. Bagaimana mengimplementasikan sistem *smart door lock* berbasis *Internet of Things (IoT)* dengan pendekatan *hybrid edge–cloud control*?
3. Bagaimana kinerja sistem *smart door lock* berbasis *IoT* dengan pendekatan *hybrid edge–cloud control* dalam memberikan respon akses pintu pada kondisi *gateway* sedang *online* maupun *offline*?

1.3 Batasan Masalah

Agar penelitian ini lebih terarah dan fokus sesuai dengan tujuan yang ingin dicapai, maka batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Pemrosesan data pada sisi *edge* dibatasi pada fungsi kontrol akses dan pengambilan keputusan pembukaan pintu secara lokal, sedangkan pemrosesan pada sisi *cloud* dibatasi pada penyimpanan data, *monitoring*, dan pengelolaan akses.
2. Pengujian sistem dibatasi pada pengujian fungsionalitas, parameter *Quality of Service (QoS)*, dan kinerja sistem pada saat perangkat *gateway* dalam kondisi *online* maupun *offline*.
3. Perangkat keras yang digunakan dibatasi pada ESP32 (sebagai kontroler node dan bridge), ESP32-CAM (pengambil citra wajah), Raspberry Pi 4 (sebagai *gateway edge*), modul pembaca RFID (PN532/RC522), keypad 4x4, serta solenoid door lock. Perangkat lunak yang digunakan dibatasi pada firmware C/C++ pada node/bridge, pemrosesan lokal berbasis Python (OpenCV, MobileFaceNet, SQLite) pada Raspberry Pi, dan layanan cloud menggunakan broker RabbitMQ (AMQP).

1.4 Tujuan dan Manfaat

Penetapan tujuan dan manfaat penelitian ini berfungsi sebagai acuan arah pengembangan serta tolok ukur keberhasilan dari sistem *smart door lock* yang dibangun. Dengan mendefinisikan target pencapaian secara jelas, penelitian ini diharapkan mampu memberikan kontribusi nyata baik dalam aspek akademis



Hak Cipta :

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

sebagai referensi ilmiah maupun dalam aspek praktis berupa peningkatan efisiensi dan keamanan kontrol akses pintu di lingkungan kampus.

1.4.1 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut:

1. Merancang sistem *smart door lock* berbasis *Internet of Things* (IoT) dengan pendekatan *hybrid edge-cloud control*.
2. Mengimplementasikan sistem *smart door lock* berbasis *Internet of Things* (IoT) dengan pendekatan *hybrid edge-cloud control*.
3. Menganalisis kinerja sistem *smart door lock* berbasis IoT dengan pendekatan *hybrid edge-cloud control* dalam memberikan respon akses pintu pada kondisi *gateway* sedang *online* maupun *offline*.

1.4.2 Manfaat

Manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Memberikan solusi sistem penguncian pintu yang lebih aman, responsif, dan andal dengan menjadikan pengenalan wajah melalui kamera sebagai autentikasi utama, menyempurnakan sistem sebelumnya yang hanya bergantung pada akses token fisik (RFID) dan PIN.
2. Memudahkan pengelolaan dan monitoring akses pintu secara terpusat melalui sistem berbasis *IoT*.
3. Memberikan bukti empiris dan evaluasi mengenai keandalan arsitektur *hybrid edge-cloud control* dalam mengatasi kendala jaringan, sehingga membuktikan bahwa sistem keamanan berbasis IoT tetap dapat beroperasi maksimal meskipun terjadi pemutusan koneksi (*offline*).

1.5 Sistematika Penulisan

Sistematika penulisan proposal skripsi ini disusun secara sistematis untuk memberikan gambaran mengenai alur dan isi penelitian yang dilakukan, dengan susunan sebagai berikut:

1. BAB I PENDAHULUAN

Bab ini berisi latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian, serta sistematika penulisan.



© 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. BAB II TINJAUAN PUSTAKA

Bab ini membahas teori dan konsep yang mendukung penelitian, meliputi *Internet of Things (IoT)*, *smart door lock*, *edge computing*, *cloud computing*, arsitektur *hybrid edge-cloud*, serta penelitian terdahulu yang relevan.

3. BAB III METODE PENELITIAN

Bab III berisikan penjelasan mengenai metodologi pelaksanaan penelitian yang digunakan sebagai acuan dasar dalam membangun sistem. Bagian ini memuat sub-bab Rancangan Penelitian yang menjelaskan penggunaan pendekatan *Research and Development (R&D)* untuk menguji efektivitas sistem *smart door lock hybrid edge-cloud*. Selain itu, bab ini memaparkan Tahapan Penelitian secara sistematis mulai dari studi literatur hingga evaluasi dan dokumentasi sistem, serta mendefinisikan Objek Penelitian terkait performa dan keandalan perangkat pengunci pintu cerdas yang diimplementasikan.

4. BAB IV HASIL DAN PEMBAHASAN

Bab IV akan berisikan pembahasan mengenai parameter pengujian kuantitatif (seperti pengujian transmisi data ESP-NOW, response time autentikasi, delay sinkronisasi, dan integritas data), hasil pengujian logika fungsional perangkat, serta evaluasi menyeluruh terhadap performa sistem hybrid yang telah diimplementasikan.

5. BAB V PENUTUP

Bab V berisikan penjelasan mengenai hasil akhir dari penelitian berupa kesimpulan dan saran untuk penelitian berikutnya

Hak Cipta :

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

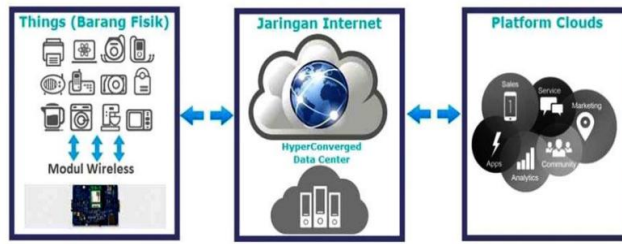
TINJAUAN PUSTAKA

2.1 Kajian Pustaka

Kajian pustaka pada sub-bab ini memaparkan landasan teori, konsep dasar, serta komponen-komponen teknologi yang digunakan sebagai acuan dalam merancang dan mengimplementasikan sistem *smart door lock* berbasis *Internet of Things* (IoT) dengan pendekatan *hybrid edge-cloud control*. Penjelasan teoritis dimulai dari konsep dasar IoT sebagai infrastruktur konektivitas utama, dilanjutkan dengan spesifikasi dan peran dari perangkat keras pendukung seperti mikrokontroler ESP32, modul kamera ESP32-CAM, dan *single-board computer* Raspberry Pi yang bertindak sebagai *gateway* lokal. Selanjutnya, dibahas pula paradigma komputasi modern seperti *edge computing* dan *cloud computing*, serta bagaimana integrasi keduanya dalam arsitektur *hybrid edge-cloud control* dapat mengoptimalkan latensi pemrosesan dan efisiensi manajemen data. Landasan teori ini sangat krusial untuk memastikan bahwa sistem yang dibangun memiliki pondasi akademis dan teknis yang kuat serta dapat berfungsi secara andal.

2.1.1 *Internet of Things* (IoT)

Internet of Things (IoT) merupakan paradigma teknologi yang memungkinkan berbagai perangkat fisik untuk saling terhubung melalui jaringan internet sehingga mampu melakukan pengumpulan, pertukaran, dan pemrosesan data secara otomatis (Ahdan & Susanto, 2021). *Internet of Things* (IoT) mengintegrasikan berbagai perangkat dan sistem melalui jaringan komunikasi untuk memungkinkan pertukaran data dan pengendalian perangkat secara otomatis (Saragih & Kurniawan, 2025). Dalam bidang keamanan, IoT dimanfaatkan untuk mendukung proses pemantauan dan pengendalian akses secara terintegrasi. Penerapan IoT pada sistem penguncian pintu memberikan pengembangan dari sistem konvensional dengan menghadirkan mekanisme akses berbasis digital serta kemudahan dalam pengelolaan sistem secara terpusat (Tawakal & Ramdhani, 2021).



Gambar 2.1 Konsep IoT

Secara umum, konsep kerja dari *Internet of Things* (IoT) melibatkan interaksi dinamis antara beberapa elemen utama, sebagaimana diilustrasikan pada Gambar 2.1. Elemen-elemen tersebut terdiri atas *things* (barang fisik seperti mikrokontroler, sensor, kamera, dan aktuator) yang dilengkapi dengan modul komunikasi nirkabel untuk mengumpulkan data dari lingkungan sekitar. Data tersebut kemudian ditransmisikan melalui jaringan internet menuju *platform clouds* untuk disimpan, diproses, dan dianalisis secara terpusat. Melalui platform awan tersebut, administrator atau pengguna dapat melakukan pemantauan kondisi secara *real-time* sekaligus mengirimkan kembali instruksi kontrol secara otomatis kepada perangkat keras fisik di lapangan.

2.1.2 ESP32

Mikrokontroler ESP32 merupakan sistem tertanam dengan dukungan komunikasi nirkabel WiFi dan Bluetooth yang memiliki kemampuan pemrosesan memadai serta konsumsi daya relatif rendah (Subito et al., 2023). ESP32 bisa digunakan untuk rangkaian pengganti pada Arduino, ESP32 memiliki kemampuan untuk mendukung terkoneksi ke WI-FI secara langsung (Nizam et al., 2022). ESP32 berperan sebagai perangkat *edge* yang mengendalikan aktuator pengunci pintu, menerima perintah dari proses pemrosesan, serta mengirimkan status operasional ke perangkat lain (Setyawan et al., 2023). Pemanfaatan ESP32 sebagai perangkat *edge* memungkinkan mekanisme kontrol akses tetap berjalan secara lokal meskipun terjadi gangguan pada koneksi internet, sehingga keandalan sistem dapat terjaga.

2.1.3 Raspberry Pi

Sistem *IoT* yang menerapkan arsitektur multi network atau menggunakan gateway membutuhkan perangkat komputasi yang memiliki biaya rendah serta konsumsi



Hak Cipta :

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



Hak Cipta :

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

daya yang efisien, seperti Raspberry Pi (Simadiputra & Surantha, 2021). Raspberry Pi banyak dimanfaatkan dalam pengembangan sistem *IoT* karena mudah dioperasikan dan dikendalikan, serta memiliki harga modul yang relatif lebih terjangkau dibandingkan dengan perangkat sejenis seperti Intel Edison, Mediatek LinkIt One, dan NVIDIA Jetson Nano (Mudaliar & Sivakumar, 2020).

Pada penelitian ini digunakan Raspberry Pi seri 4B yang dilengkapi dengan *System on Chip* (SoC) Broadcom BCM2711 berarsitektur Quad-Core Cortex-A72 (ARM v8) 64-bit dengan kecepatan 1,8 GHz. Perangkat ini didukung oleh memori LPDDR4-3200 SDRAM sebesar 4 GB, serta kapabilitas konektivitas nirkabel IEEE 802.11ac 2,4 GHz dan 5,0 GHz, Bluetooth 5.0, dan Gigabit Ethernet. Raspberry Pi 4B beroperasi dengan catu daya 5 V arus searah (DC) dan arus minimal sebesar 3 A melalui konektor USB-C, sehingga memiliki kapasitas komputasi dan transfer data yang tangguh untuk diterapkan sebagai *gateway* dan perangkat pemrosesan dalam sistem *IoT* (Raspberry Pi Foundation, n.d.).

2.1.4 ESP32CAM

ESP32-CAM merupakan modul mikrokontroler yang mengintegrasikan chip ESP32 dengan kamera digital dalam satu papan, serta dilengkapi kemampuan komunikasi nirkabel seperti Wi-Fi dan Bluetooth, sehingga banyak digunakan dalam pengembangan sistem *Internet of Things (IoT)* (Isrofi et al., 2021). Modul ini mendukung proses akuisisi citra secara langsung dan pengiriman data melalui jaringan, menjadikannya sesuai untuk aplikasi sistem keamanan. Dalam sistem *smart door lock*, ESP32-CAM berperan sebagai perangkat *edge* yang menangkap gambar saat terjadi aktivitas tertentu dan mengirimkan data tersebut ke sistem monitoring, sehingga mendukung fungsi pengawasan dan pencatatan aktivitas akses secara *real-time*.

2.1.5 Edge Computing

Edge computing merupakan pendekatan komputasi yang mengalihkan proses pemrosesan data dari pusat data terpusat atau layanan *cloud* ke perangkat yang berada di dekat sumber data, seperti sensor dan aktuator (Permadi et al., 2024). Pendekatan ini bertujuan untuk meningkatkan efisiensi sistem dengan cara



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

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

menurunkan latensi, mempercepat waktu respons, serta mengurangi ketergantungan terhadap komunikasi jaringan yang terus menerus(Li et al., 2022). Dengan pemrosesan yang dilakukan secara lokal, sistem dapat merespons peristiwa secara langsung tanpa harus menunggu hasil pemrosesan dari *cloud*. Dalam penerapan *smart door lock*, *edge computing* memungkinkan proses autentikasi pengguna dan pengambilan keputusan terkait pembukaan atau penguncian pintu dilakukan di sisi perangkat, sehingga sistem keamanan tetap berjalan secara optimal meskipun koneksi internet mengalami gangguan atau tidak stabil. Hal ini sangat krusial untuk aplikasi keamanan, dimana respons cepat dan keandalan operasional menjadi prioritas utama (Dewi & Fikri, 2023).

2.1.6 Cloud Computing

Cloud computing merupakan model komputasi yang menyediakan sumber daya seperti penyimpanan data, pemrosesan, dan aplikasi melalui jaringan internet sehingga dapat diakses sesuai kebutuhan pengguna (Arbain et al., 2022). Teknologi ini memungkinkan pemanfaatan layanan komputasi secara fleksibel tanpa dibatasi oleh lokasi fisik pengguna (Setiawan et al., 2023). *Cloud computing* memiliki tiga aspek utama, yaitu karakteristik, model layanan, dan model implementasi. Karakteristik *cloud computing* meliputi *on-demand self service*, *broad network access*, *resource pooling*, *rapid elasticity*, dan *measured service*. Model layanan *cloud computing* terdiri dari *software as a service (SaaS)*, *platform as a service (PaaS)*, dan *infrastructure as a service (IaaS)*, sedangkan model implementasinya mencakup *public cloud*, *private cloud*, *hybrid cloud*, dan *community cloud*. Dalam sistem *IoT*, *cloud computing* berperan sebagai pusat penyimpanan data, *monitoring*, dan pengelolaan akses secara terpusat. Namun, ketergantungan penuh terhadap *cloud* dapat meningkatkan latensi serta menimbulkan risiko gangguan layanan ketika koneksi jaringan tidak stabil.

2.1.7 Arsitektur Hybrid Edge–Cloud Control

Arsitektur *hybrid edge–cloud control* merupakan pendekatan komputasi yang mengintegrasikan kemampuan pemrosesan lokal pada sisi *edge* dengan kapasitas komputasi dan penyimpanan terpusat pada sisi *cloud* untuk mendukung aplikasi *Internet of Things (IoT)* yang bersifat *real-time* dan berskala besar. Pendekatan ini



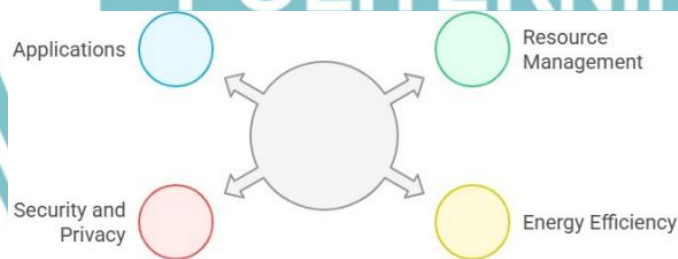
© Hak Cipta milik Politeknik Negeri Jakarta

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

dikembangkan sebagai respons terhadap keterbatasan arsitektur cloud-only yang mengalami permasalahan latensi tinggi, keterbatasan bandwidth, serta inefisiensi energi ketika menangani data *IoT* dalam jumlah besar (Pallikonda et al., 2025). Dalam arsitektur hybrid, lapisan *edge* bertanggung jawab terhadap pemrosesan data yang bersifat sensitif terhadap waktu, seperti pemrosesan awal data, deteksi kejadian, dan pengambilan keputusan cepat. Pemrosesan dilakukan sedekat mungkin dengan sumber data untuk meminimalkan waktu transmisi dan meningkatkan kecepatan respons sistem. Sementara itu, lapisan cloud difungsikan untuk menangani komputasi kompleks, analisis data skala besar, serta penyimpanan jangka panjang dan manajemen sistem secara terpusat (Maturi et al., 2024).

Penelitian Pallikonda et al. (2025) menunjukkan bahwa penerapan arsitektur *hybrid edge-cloud* dengan mekanisme dynamic task offloading mampu menurunkan latensi hingga sekitar 70 persen dan menghemat konsumsi energi hingga 55 persen dibandingkan pendekatan cloud-only. Selain itu, arsitektur ini mampu mencapai efisiensi offloading tugas hingga 92 persen, sehingga lebih andal dan skalabel untuk aplikasi *IoT real-time*. Untuk memperjelas konsep dan komponen utama dalam arsitektur *hybrid cloud-edge*, ilustrasi arsitektur tersebut ditunjukkan pada Gambar 2 (Pallikonda et al., 2025).



Gambar 2.2 Arsitektur *hybrid cloud-edge*

Gambar 2.2 menunjukkan arsitektur *hybrid cloud-edge* yang mendukung pemrosesan data *IoT* secara *real-time* dengan mengintegrasikan elemen *applications*, *resource management*, *energy efficiency*, serta *security and privacy*. Arsitektur ini memungkinkan pembagian beban komputasi antara sisi *edge* dan *cloud* secara optimal sehingga dapat meningkatkan kinerja, efisiensi energi, dan keamanan sistem (Pallikonda et al., 2025). Dalam konteks sistem *smart door lock*



Hak Cipta :

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

berbasis pengenalan wajah, pendekatan *hybrid edge-cloud* memungkinkan proses autentikasi dilakukan secara lokal di sisi *edge* untuk memperoleh respons cepat, sementara data akses dan monitoring jangka panjang dikelola pada sisi *cloud* agar sistem tetap andal meskipun terjadi gangguan jaringan.

2.2 Penelitian Terdahulu

Tinjauan terhadap studi terdahulu diperlukan sebagai landasan teoritis sekaligus pembandingan guna memastikan posisi dan kebaruan penelitian ini. Oleh karena itu, beberapa karya ilmiah yang relevan dengan pengembangan sistem smart door berbasis *Internet of Things (IoT)*, khususnya yang berkaitan dengan otentikasi wajah, kontrol akses pintu, serta pemanfaatan arsitektur *edge*, *cloud*, dan *hybrid*, dikaji sebagai acuan utama dalam perancangan dan implementasi sistem yang diusulkan. Ringkasan penelitian-penelitian tersebut disajikan pada tabel berikut.

Tabel 2.1 Penelitian Terdahulu

No	Jurnal	Hasil	Catatan
1	Sopandi & Hendrawati (2023)	Tujuan : Merancang sistem <i>smart door lock</i> berbasis <i>IoT</i> dengan verifikasi wajah untuk meningkatkan keamanan akses pintu. Teknologi : Raspberry Pi 3, Webcam, Solenoid Door Lock, Python, Convolutional Neural Network (CNN), Telegram Bot, Database MySQL. Cara Kerja : Sistem melakukan akuisisi citra wajah melalui webcam yang terhubung ke Raspberry Pi. Citra wajah diproses menggunakan metode CNN untuk proses pengenalan wajah. Jika wajah dikenali, solenoid door lock akan terbuka secara otomatis dan identitas pengguna ditampilkan pada LCD. Jika wajah tidak dikenali, sistem akan mengirimkan notifikasi berupa citra wajah ke pemilik melalui aplikasi Telegram serta mencatat aktivitas akses ke dalam database.	Sistem masih berbasis <i>standalone edge</i> tanpa arsitektur <i>hybrid</i> terstruktur sehingga tidak memiliki toleransi kesalahan saat koneksi terputus. Skripsi ini mengatasi hal tersebut dengan menerapkan arsitektur <i>hybrid edge-cloud</i> berbasis RabbitMQ (AMQP) yang mendukung sinkronisasi lokal dan mekanisme <i>buffering</i> saat gateway <i>offline</i>.



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

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

2	Syarifullah et al. (2023)	Tujuan : Menerapkan sistem keamanan rumah berbasis <i>IoT</i> menggunakan ESP32-CAM dengan metode edge detection Sobel untuk identifikasi wajah dan kontrol akses pintu. Teknologi : ESP32-CAM, Kamera OV2640, Solenoid Door Lock, Web Browser, Metode Edge Detection Sobel. Cara Kerja : ESP32-CAM menangkap citra wajah secara <i>real-time</i> dan memprosesnya menggunakan metode edge detection Sobel untuk mendeteksi karakteristik wajah. Jika wajah dikenali sebagai pemilik rumah, solenoid door lock akan terbuka secara otomatis. Jika tidak dikenali, hasil tangkapan citra dapat dipantau melalui web browser sebagai notifikasi keamanan.	Pemrosesan berjalan lokal sepenuhnya (<i>edge-only</i>) tanpa pemantauan log di <i>cloud</i> serta memiliki akurasi deteksi wajah yang rendah (metode Sobel). Solusi skripsi ini adalah mengintegrasikan sisi <i>edge</i> lokal dengan dasbor monitoring <i>cloud</i> terpusat serta menggunakan pustaka pengenalan wajah ekstraksi fitur yang lebih akurat pada gateway.
3	Putri et al. (2025)	Tujuan : Mengembangkan sistem keamanan pintu berbasis pengenalan wajah menggunakan algoritma YOLO untuk meningkatkan keamanan dan kemudahan akses pada sistem kunci digital. Teknologi : ESP32-CAM, Webcam, Solenoid Door Lock, Cloud Server, Python, YOLOv5 (Roboflow). Cara Kerja : Citra wajah diambil secara <i>real-time</i> melalui webcam dan diproses menggunakan model YOLO yang dijalankan pada cloud server. Jika wajah terdaftar terdeteksi dengan confidence $\geq 50\%$, sistem mengirimkan perintah melalui HTTP request ke ESP32-CAM untuk mengaktifkan relay dan membuka solenoid door lock. Hasil pengujian menunjukkan akurasi pengenalan wajah sebesar 94,4% dengan waktu prediksi sekitar 1 detik.	Pengenalan wajah sepenuhnya bergantung pada <i>cloud server</i> (arsitektur <i>cloud-only</i>), sehingga rentan terhadap latensi tinggi dan pintu gagal diakses saat luring. Solusi skripsi ini adalah memindahkan proses komputasi pengenalan wajah ke tingkat lokal (<i>edge gateway</i> Raspberry Pi) agar tetap responsif (~560 ms RTT) meskipun luring.



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

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

4	Kundang & Kadek (2021)	Tujuan : Mengembangkan sistem <i>smart door lock</i> berbasis <i>IoT</i> dengan arsitektur Device-to-Cloud (D2C) yang memanfaatkan layanan Amazon Web Services (AWS) untuk melakukan autentikasi wajah, kontrol akses pintu berbasis suara, serta pengelolaan data pengguna secara terpusat dan otomatis. Teknologi : Raspberry Pi, Raspberry Pi Camera Module, Arduino Uno, Amazon Alexa (pengambilan perintah suara), AWS Python SDK, Amazon Alexa Skill Kit, AWS DynamoDB (NoSQL fully managed database untuk penyimpanan dan pengindeksan data pengguna), AWS IAM (Identity and Access Management), AWS SNS (Simple Notification Service), Amazon Rekognition (layanan identifikasi wajah tanpa training model), Amazon Polly (text-to-speech), AWS Lambda (serverless computing), AWS <i>IoT</i> (komunikasi perangkat menggunakan MQTT), Amazon S3 (penyimpanan citra). Cara Kerja : Sistem bekerja dengan menangkap citra wajah menggunakan Raspberry Pi Camera Module, kemudian citra tersebut dikirim ke cloud untuk proses identifikasi wajah. Hasil identifikasi disampaikan kepada pemilik melalui notifikasi dan umpan balik suara. Pemilik dapat memberikan perintah suara melalui Amazon Alexa untuk membuka pintu, yang selanjutnya diproses oleh layanan cloud dan dikirim ke perangkat melalui MQTT. Raspberry Pi menerima perintah tersebut dan meneruskannya ke Arduino Uno untuk mengaktifkan mekanisme pengunci pintu.	Menggunakan arsitektur <i>full cloud</i> (D2C) yang sangat bergantung pada kestabilan internet untuk otentikasi sehingga latensi tinggi dan rentan gagal saat luring. Solusi skripsi ini adalah mengalihkan komputasi otentikasi wajah secara lokal di <i>gateway</i> Raspberry Pi, sementara <i>cloud</i> hanya digunakan untuk monitoring log riwayat akses dan sinkronisasi data.
5	Fachrudin (2023)	Tujuan : Mengembangkan sistem pintu cerdas yang mampu melakukan autentikasi pengguna dan pengelolaan hak akses ruangan secara terpusat melalui perangkat <i>gateway</i>, serta tetap dapat beroperasi pada lingkungan gedung bertingkat dengan meminimalkan area tanpa jangkauan (<i>dead zone</i>). Teknologi : Raspberry Pi sebagai <i>gateway</i>, ESP32 sebagai bridge dan node, jaringan mesh untuk komunikasi antar perangkat, RFID	Jaringan Mesh berbasis WiFi lokal membutuhkan konfigurasi rumit dengan daya tinggi serta otentikasi masih terbatas pada kartu RFID



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

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

	sebagai media autentikasi pengguna, magnetic door lock sebagai aktuator pengunci pintu, LCD sebagai antarmuka tampilan, komunikasi data berbasis jaringan lokal.	tanpa biometrik wajah. Solusi skripsi ini adalah memadukan komunikasi <i>wired</i> dan <i>wireless</i> (ESP Mesh/ESP-NOW) berdaya rendah serta menambahkan pengenalan wajah sebagai otentikasi utama dan MFA opsional.
--	---	---

Hak Cipta :

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

BAB III METODE PENELITIAN

3.1 Rancangan Penelitian

Rancangan penelitian ini menggunakan metode *Research and Development* (R&D) untuk menghasilkan produk berupa sistem *smart door lock* berbasis *Internet of Things* (IoT) dengan pendekatan *hybrid edge-cloud control* serta menguji kinerjanya secara nyata (Robi & Okra, 2023). Penelitian ini akan berfokus pada pengembangan Sistem *Smart Door Lock* berbasis *IoT* dengan Pendekatan *Hybrid Edge-Cloud Control*. Pendekatan *hybrid* ini (yang didukung komunikasi nirkabel *ESP Mesh* dan *ESP-NOW*) akan membuat sistem lebih andal, meminimalkan area tanpa sinyal (*dead zone*), serta memastikan pintu tetap dapat beroperasi dan melakukan autentikasi secara lokal meskipun layanan *cloud* sedang tidak tersedia atau terjadi gangguan internet. Sinkronisasi data hak akses dan riwayat pengguna akan dilakukan pada *gateway* (Raspberry Pi) dengan tujuan untuk membuat salinan data lokal dari *cloud service*, di mana antrean pesan sinkronisasi dikelola menggunakan protokol AMQP melalui RabbitMQ.

Teknik pengumpulan data dalam penelitian ini dilakukan melalui observasi langsung terhadap jalannya sistem serta pencatatan log sistem otomatis menggunakan skrip python dengan mengacu pada metode pengukuran parameter performa sistem dan *Quality of Service* (QoS) yang diterapkan oleh Simadiputra & Surantha (2021) serta Murjitama et al. (2024). Data kuantitatif yang dikumpulkan berfokus pada metrik *Quality of Service* (QoS) jaringan dan performa sistem, yang meliputi waktu respons (*response time*) autentikasi wajah, delay sinkronisasi data pengguna dari *cloud* ke *gateway* melalui RabbitMQ (AMQP), latensi transmisi nirkabel ESP-NOW, persentase kegagalan pengiriman data (*packet loss*), serta variasi waktu kedatangan paket (*jitter*).

Selanjutnya, teknik analisis data yang digunakan adalah analisis statistik deskriptif. Analisis dilakukan dengan menghitung metrik statistik deskriptif seperti nilai rata-rata (*mean*), minimum, maksimum, serta persentase *packet loss* dari data pengujian. Hal ini bertujuan untuk menggambarkan secara kuantitatif tingkat performa,



Hak Cipta :

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

kecepatan respons, dan keandalan sistem *smart door lock* pada kondisi saat perangkat *gateway* sedang *online* maupun *offline*.

3.2 Tahapan Penelitian

Tahapan penelitian pada perancangan dan implementasi sistem ini disusun secara sistematis dengan mengacu pada alur metode *Research and Development (R&D)*. Alur ini dirancang secara sistematis untuk memastikan proses pengembangan dari tahap awal analisis hingga pengujian dan evaluasi produk teknologi dapat berjalan secara terstruktur.

Rincian dan daftar kegiatan yang akan dilakukan untuk setiap tahapannya akan dipaparkan dalam uraian di bawah ini.

a) Studi Literatur

Studi literatur dilakukan untuk memperoleh pemahaman mengenai konsep dan teknologi yang berkaitan dengan *Internet of Things (IoT)*, *smart door lock*, pengenalan wajah, *edge computing*, *cloud computing*, serta arsitektur *hybrid edge-cloud control*. Studi ini bertujuan untuk mengetahui perkembangan teknologi yang telah dikembangkan sebelumnya, sekaligus mengidentifikasi kelebihan dan kekurangannya. Kekurangan yang ditemukan pada penelitian atau sistem terdahulu akan dijadikan dasar dalam menentukan arah dan fokus penelitian ini. Studi literatur dilakukan dengan menelaah artikel ilmiah, jurnal penelitian, buku referensi, serta dokumentasi perangkat keras dan pustaka perangkat lunak yang digunakan.

b) Analisis Kebutuhan Sistem

Pada tahap ini dilakukan analisis terhadap kebutuhan sistem yang akan dikembangkan, meliputi kebutuhan perangkat keras, perangkat lunak, arsitektur sistem, serta mekanisme autentikasi yang digunakan. Analisis ini bertujuan untuk menentukan spesifikasi sistem *smart door lock* berbasis *IoT* dengan pendekatan *hybrid edge-cloud control*, termasuk penentuan peran pemrosesan pada sisi *edge* dan *cloud*, serta konfigurasi autentikasi menggunakan kamera, RFID, dan keypad.



© 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

c) Perancangan Sistem

Tahap perancangan sistem dilakukan berdasarkan hasil studi literatur dan analisis kebutuhan. Pada tahap ini dirancang blok diagram sistem, arsitektur sistem hybrid edge–cloud, perangkat keras (gateway dan node), sistem sinkronisasi data, serta alur proses autentikasi pengguna.

d) Implementasi Sistem

Tahap implementasi meliputi perakitan perangkat keras dan pengembangan perangkat lunak sesuai dengan rancangan sistem yang telah dibuat. Implementasi dilakukan pada perangkat *edge* untuk menangani proses autentikasi dan kontrol aktuator secara *real-time*, serta integrasi dengan layanan *cloud* untuk penyimpanan data dan monitoring sistem.

e) Pengujian dan Pengambilan Data

Pengujian sistem dilakukan untuk memastikan seluruh fungsi berjalan sesuai dengan perancangan. Pengujian meliputi pengujian fungsionalitas sistem, waktu respons autentikasi, serta kinerja sistem pada saat perangkat *gateway* dalam kondisi *online* maupun *offline*. Data hasil pengujian dikumpulkan sebagai bahan evaluasi kinerja sistem.

f) Evaluasi dan Analisis

Tahap evaluasi dan analisis dilakukan dengan menganalisis data hasil pengujian untuk menilai keandalan sistem, efektivitas pemrosesan di sisi *edge*, serta peran *cloud* dalam mendukung sistem. Hasil analisis ini digunakan untuk menarik kesimpulan mengenai kinerja sistem *smart door lock* berbasis *IoT* dengan pendekatan *hybrid edge–cloud control*.

3.3 Objek Penelitian

Objek penelitian pada penelitian ini adalah sistem *smart door lock* berbasis *Internet of Things (IoT)* dengan pendekatan *hybrid edge cloud control* yang diterapkan di Ruang Dosen Gedung AA Politeknik Negeri Jakarta. Sistem yang diteliti meliputi perangkat keras dan perangkat lunak yang digunakan dalam proses autentikasi akses pintu, yaitu kamera sebagai autentikasi utama, serta RFID dan keypad sebagai autentikasi tambahan sesuai dengan konfigurasi yang dipilih oleh pengguna. Selain



© Hak Cipta milik Politeknik Negeri Jakarta

itu, objek penelitian juga mencakup mekanisme pemrosesan data pada sisi *edge* serta peran *cloud* dalam penyimpanan dan monitoring data akses. Fokus penelitian diarahkan pada kinerja sistem dalam memberikan respons akses pintu, keandalan pemrosesan di sisi *edge*, serta efektivitas sistem pada kondisi jaringan yang berbeda.



Hak Cipta :

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

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

Penelitian tentang sistem pintu pintar ini dikembangkan lebih lanjut melalui desain Kunci Pintu Pintar berbasis *IoT* dengan pendekatan *hybrid edge-cloud Control* berbasis wajah dan Otentikasi Multi-Faktor (MFA) opsional. Sistem ini dirancang untuk memiliki kemampuan melakukan pemindaian wajah dan proses verifikasi ganda, di mana penggunaan MFA fleksibel dan bergantung pada preferensi atau kebutuhan setiap pengguna. Sistem ini masih dapat melakukan proses otentikasi secara independen tanpa bergantung pada koneksi internet. Untuk mengakomodasi kebutuhan ini, teknologi transmisi data yang digunakan harus mampu meminimalkan *dead zone*, sehingga cakupan sistem luas dan stabil. Selain itu, arsitektur *hybrid* ini mensyaratkan bahwa proses autentikasi wajah dan MFA masih dapat dijalankan di tingkat *edge* meskipun layanan *cloud* atau koneksi internet tidak tersedia, hal ini dilakukan untuk memastikan ruangan tetap dapat diakses dan tidak mengganggu aktivitas pengunjung atau pengguna ruangan. *Gateway* atau perangkat *edge* juga diharuskan mampu menyinkronkan data pengguna tanpa *packet loss*, di mana semua data template wajah dan pengaturan hak akses MFA yang dipilih harus direplikasi dengan sempurna untuk memastikan akses ruangan tanpa gangguan. Untuk merealisasikan fungsionalitas tersebut, diperlukan analisis kebutuhan komponen sistem yang terbagi atas kebutuhan perangkat keras dan kebutuhan perangkat lunak.

4.1.1 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras dalam implementasi sistem *smart door lock* ini mencakup seluruh komponen elektronik fisik yang digunakan pada unit *node*, *bridge*, dan *gateway*. Spesifikasi kebutuhan perangkat keras tersebut disajikan pada Tabel 4.1 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 4.1 Spesifikasi Kebutuhan Perangkat Keras

No	Nama Komponen	Spesifikasi
1	ESP32 Node & Bridge	Prosesor Tensilica Xtensa Dual-Core 32-bit LX6, Wi-Fi 802.11 b/g/n & Bluetooth v4.2 BLE, Tegangan kerja 3.3V, SRAM 520 KB & Flash memory 4MB, Mendukung ESP Mesh & ESP-NOW
2	ESP32-CAM	Prosesor Tensilica Xtensa 32-bit LX6 dengan kamera OV2640, Resolusi kamera hingga 2 Megapixel (UXGA), Wi-Fi 802.11 b/g/n, Terintegrasi slot kartu MicroSD
3	Raspberry Pi 4B	Prosesor Broadcom BCM2711, Quad-core Cortex-A72 64-bit SoC @ 1.5GHz, RAM 4GB/8GB LPDDR4-3200 SDRAM, Wi-Fi 2.4/5.0 GHz & Bluetooth 5.0 BLE, Port Gigabit Ethernet, USB 3.0 & USB 2.0
4	Keypad 4x4	Tipe Matrix Membrane Keypad, Antarmuka 8-pin digital output, Desain tipis dan fleksibel
5	RFID Reader (PN532/RC522)	Frekuensi kerja 13.56 MHz RFID/NFC, Antarmuka komunikasi SPI / I2C / HSU, Jarak pemindaian kartu 2 cm hingga 5 cm
6	Relay 5V	Tegangan pemicu 5V DC, Arus kontak 10A 250VAC / 10A 30VDC, Konfigurasi SPDT (Single Pole Double Throw)
7	Solenoid Door Lock	Tegangan kerja 12V DC, Arus operasi: ~0.8A - 1.2A, Jenis Fail-Secure (terkunci saat tidak ada daya)
8	Adaptor 12V	Tegangan input 100-240V AC, Tegangan output 12V DC, Arus output minimal: 2A
9	LM2596	Modul Step-Down DC-DC Converter, Input 4.5V - 40V DC, Output 1.25V - 37V DC, Arus Output Maksimum 3A
10	LCD OLED	Resolusi 128x64 pixel, Antarmuka I2C, Tegangan kerja 3.3V - 5V, Konsumsi daya rendah



Hak Cipta :

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

11	LCD Raspberry Pi 7"	Resolusi 800x480 pixel, Layar sentuh kapasitif, Antarmuka DSI (Display Serial Interface), Mendukung Raspberry Pi
----	---------------------	--

4.1.2 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak dalam perancangan sistem ini mencakup sistem operasi, *database*, dan platform cloud untuk integrasi data secara terpusat. Spesifikasi kebutuhan perangkat lunak tersebut disajikan pada Tabel 4.2 berikut.

Tabel 4.2 Spesifikasi Kebutuhan Perangkat Lunak

No	Nama Perangkat Lunak	Spesifikasi / Deskripsi
1	Python 3	Versi 3.8 ke atas, Penggunaan untuk database lokal dan modul AMQP RabbitMQ pada Gateway
2	SQLite Database	Kegunaan DBMS lokal relasional pada Raspberry Pi untuk menyimpan kredensial user, template wajah, dan log aktivitas luring
3	RabbitMQ (AMQP)	Kegunaan Broker pesan asinkron untuk menjamin sinkronisasi data andal dengan mekanisme antrean buffering saat gateway offline
4	PostgreSQL Database	Kegunaan Basis data relasional utama di sisi cloud untuk penyimpanan data terpusat dan log riwayat akses global
5	Cloud Run Service	Kegunaan Platform hosting API serverless berbasis container untuk melayani pertukaran data backend secara terpusat
6	C/C++ (Arduino IDE)	Penggunaan Bahasa pemrograman untuk menulis firmware mikrokontroler ESP32 Node, ESP32-CAM, dan ESP32 Bridge

4.2 Perancangan Sistem

Perancangan sistem *smart door lock* berbasis *Internet of Things* (IoT) dengan pendekatan *hybrid edge-cloud control* ini dirancang untuk beroperasi pada



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

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

bangunan bertingkat dengan mengintegrasikan fungsionalitas perangkat *node* dan *gateway* secara nirkabel. Pada sisi *node* di setiap pintu, sistem mengandalkan ESP32 sebagai mikrokontroler utama yang terhubung dengan ESP32-CAM untuk pengambilan citra wajah, serta menyediakan modul RFID dan *keypad* untuk mendukung mekanisme *Multi-Factor Authentication* (MFA) yang bersifat opsional. Agar cakupan jaringan menjadi luas dan meminimalkan area tanpa sinyal (*dead zone*), sistem memanfaatkan komunikasi *ESP Mesh* dan *ESP NOW* dimana data dari setiap *node* diteruskan melalui perangkat *bridge* menuju *gateway*. Perangkat *gateway* yang menggunakan Raspberry Pi 4 berfungsi sebagai pusat pemrosesan tingkat *edge*, yang memungkinkan proses verifikasi wajah dan sinkronisasi data akses tetap dapat dieksekusi secara mandiri tanpa *packet loss* meskipun koneksi internet ke *cloud* sedang tidak tersedia. Apabila autentikasi berhasil diverifikasi, *gateway* akan memberikan instruksi kembali kepada *node* untuk membuka aktuator pengunci pintu melalui kendali modul *relay*.

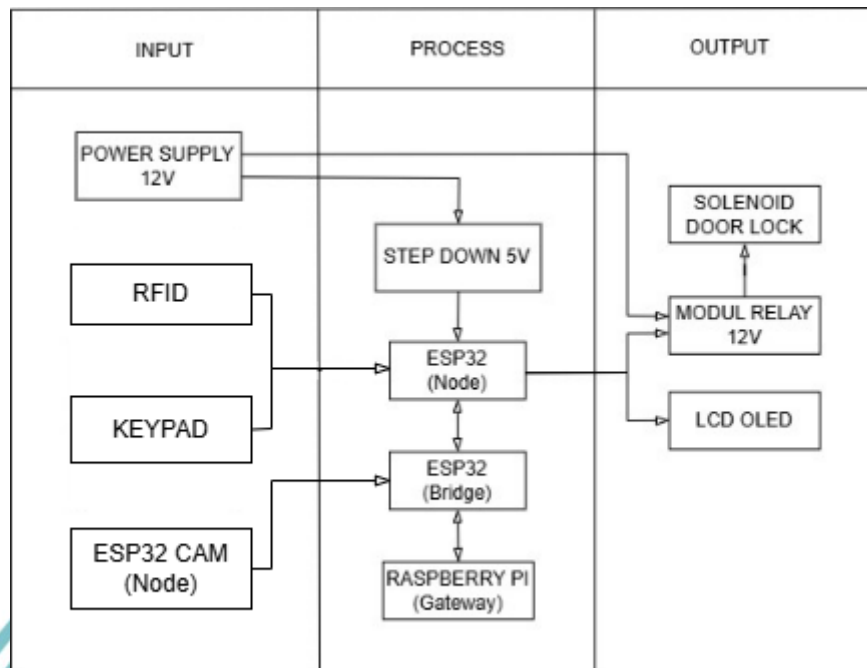
4.2.1 Blok Diagram

Berdasarkan Gambar 4.1, sistem menerima input berupa catu daya 12V, RFID, keypad, serta modul ESP32-CAM, di mana pada bagian proses, tegangan utama 12V diturunkan menggunakan step-down 5V untuk menyuplai perangkat ESP32 (Node). Alur kerja sistem diawali ketika perangkat input seperti RFID dan keypad mengirimkan data menuju ESP32 (*node*), sementara ESP32-CAM mengambil data citra wajah yang berfungsi sebagai autentikasi utama dan langsung diteruskan menuju ESP32 (*bridge*) menggunakan komunikasi ESP-NOW. Di dalam bagian proses, ESP32 (*node*) dan ESP32 (*bridge*) saling berkomunikasi dua arah menggunakan ESP Mesh, lalu data diteruskan dari ESP32 (*bridge*) secara dua arah pula menuju Raspberry Pi (*gateway*) yang bertindak sebagai pusat pemrosesan dan pengambil keputusan.



Hak Cipta :

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



Gambar 4.1 Blok Diagram Sistem

Setelah proses komputasi selesai dilakukan di tingkat *gateway*, perintah eksekusi akan dikirimkan kembali secara berjenjang melalui ESP32 (*bridge*) hingga diterima oleh ESP32 (*node*). Ketika perintah tersebut diterima, ESP32 (*node*) langsung mengaktifkan bagian output, yaitu mengirimkan data informasi untuk ditampilkan pada layar LCD OLED serta memicu modul relay 12V. Modul relay 12V ini, yang juga menerima suplai daya langsung dari catu daya 12V, akan bertindak sebagai sakelar untuk mengaktifkan aktuator solenoid door lock sehingga sistem pengunci pintu dapat terbuka.

4.2.2 Arsitektur Sistem

Berdasarkan Gambar 4.2, hubungan interkoneksi dan alur kerja antar komponen dalam arsitektur sistem dapat dibagi menjadi beberapa bagian utama. Pada sisi *node*, sistem mengandalkan keypad dan pembaca RFID sebagai *input* data pengguna yang akan dibaca oleh mikrokontroler ESP32 Node. Mikrokontroler ini juga terhubung secara langsung dengan modul ESP32-CAM yang bertindak sebagai pengambil citra wajah ketika user menekan huruf B pada *keypad*. Untuk mengeksekusi perintah akses pintu, ESP32 Node dihubungkan ke modul Relay 5V yang bertindak sebagai sakelar elektronik untuk mengalirkan tegangan dari Adaptor

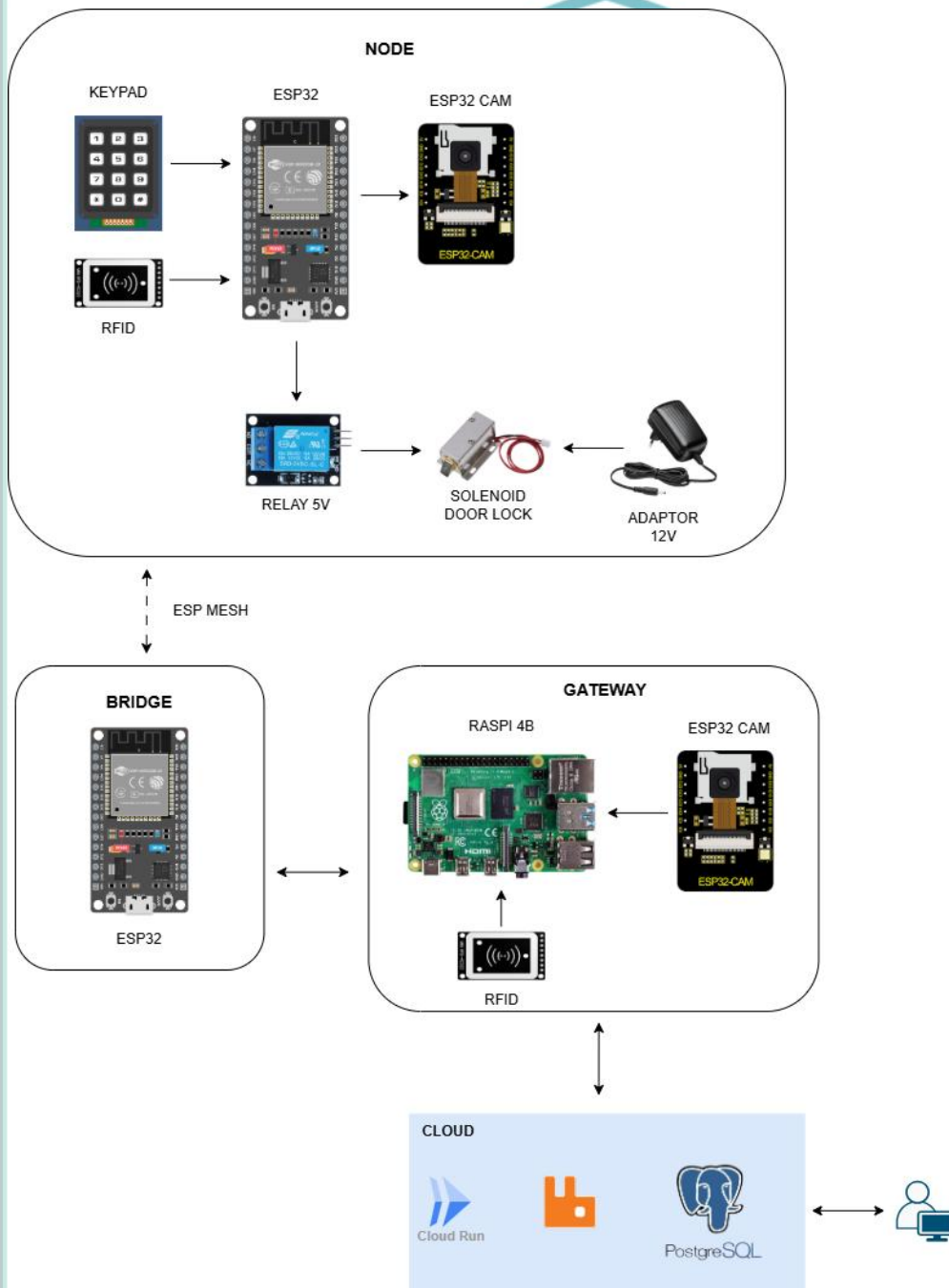


© Hak Cipta milik Politeknik Negeri Jakarta

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 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

12V menuju aktuator *Solenoid Door Lock*. Karena Raspberry Pi tidak dapat berkomunikasi secara langsung di dalam jaringan mesh nirkabel, perangkat ESP32 *Bridge* diimplementasikan sebagai jembatan perantara untuk menerima seluruh data dari ESP32 *Node* melalui protokol komunikasi *ESP Mesh* dan *ESP Now*, yang selanjutnya diteruskan ke perangkat *gateway* menggunakan koneksi serial fisik.



Gambar 4.2 Arsitektur 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

Pada sisi *gateway* yang berbasis Raspberry Pi 4B, perangkat ini bertindak sebagai pusat pemrosesan lokal pada tingkat *edge*. Raspberry Pi 4B menerima aliran data autentikasi dari ESP32 Bridge untuk mencocokkan citra wajah secara lokal, serta menyimpan riwayat log aktivitas ke dalam database lokal SQLite secara mandiri. Di *gateway* ini, terpasang pula modul ESP32-CAM dan RFID yang terhubung langsung untuk keperluan autentikasi. Sementara itu, proses sinkronisasi data dinamis antara database lokal pada *gateway* dengan sistem terpusat dikelola melalui koneksi internet nirkabel menuju *cloud*. Infrastruktur *cloud* ini terdiri dari broker pesan RabbitMQ (AMQP) untuk menangani antrean pesan sinkronisasi secara andal, serta database utama terpusat PostgreSQL yang menyimpan seluruh kredensial pengguna dan log akses global untuk dipantau oleh admin maupun pengguna melalui *website*

4.2.3 Perangkat Keras

Sistem *smart door lock* berbasis *Internet of Things* dengan pendekatan *hybrid edge-cloud control* ini terdiri dari dua jenis perangkat keras utama, yaitu *gateway* dan *node*. Perangkat *gateway* dapat diletakkan di ruang alat, dan secara khusus bertugas untuk membaca data *Multi-Factor Authentication* (MFA), melakukan autentikasi citra wajah, memverifikasi kecocokan data pengguna, mengelola pendaftaran, serta menyinkronkan hak akses untuk setiap ruangan. Sementara itu, perangkat keras kedua adalah *node* yang ditempatkan secara spesifik di pintu ruang dosen. *Node* berfungsi sebagai titik interaksi awal yang memberikan *trigger* untuk pengambilan citra wajah, menerima *input* fisik dari pengguna, serta mengatur aktuator (*solenoid door lock*) untuk memberikan atau menolak akses masuk ke dalam ruangan berdasarkan keputusan komputasi dari *gateway*.

Perangkat *gateway* terdiri dari sebuah *single-board computer* Raspberry Pi 4, layar sentuh LCD 7 inci, ESP32 Cam, modul RFID PN532, serta mikrokontroler ESP32. Raspberry Pi dipilih karena merupakan solusi komputasi berbiaya rendah dan efisien daya yang sanggup menangani pemrosesan data lokal seperti pengenalan wajah dan pembacaan data MFA untuk arsitektur *gateway*. Selain itu, Raspberry Pi lebih mudah dioperasikan dan harga modulnya relatif lebih terjangkau dibandingkan perangkat sejenis. Layar LCD 7 inci digunakan sebagai antarmuka



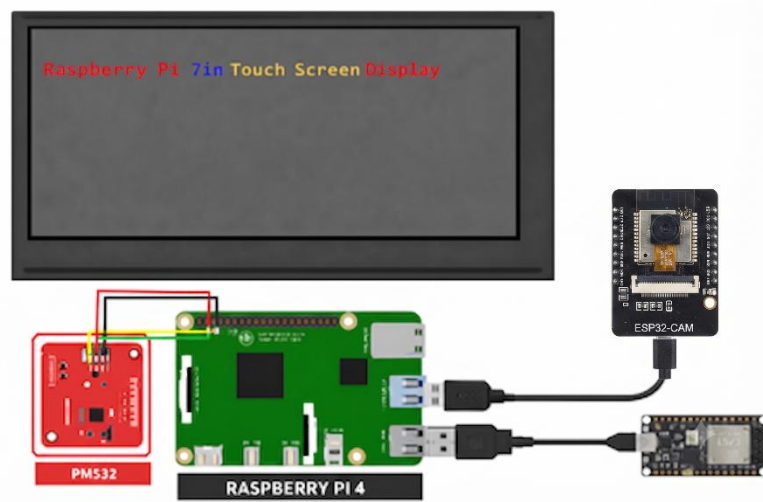
© Hak Cipta milik Politeknik Negeri Jakarta

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

admin untuk menampilkan informasi terkait status sistem, *log* sinkronisasi akses, dan *enrollment*. ESP32 Cam digunakan untuk mendukung pengambilan gambar saat proses pendaftaran wajah pengguna baru di tingkat administrator.

ESP32 pada *gateway* difungsikan sebagai modul *bridge* atau jembatan komunikasi, agar *gateway* (Raspberry Pi) dapat terhubung secara nirkabel dengan seluruh *node* menggunakan jaringan *ESP Mesh*. Gambar 4.3 memperlihatkan detail perakitan komponen keras pada *gateway*. Modul ESP32 dihubungkan ke Raspberry Pi menggunakan antarmuka USB dengan komunikasi serial, sehingga mempermudah pemasangan perangkat dan memungkinkan fleksibilitas penambahan *bridge* lain jika dibutuhkan guna mengurangi risiko *dead zone* pada gedung bertingkat. Modul RFID PN532 dihubungkan langsung ke *port* GPIO Raspberry Pi untuk keperluan pendaftaran kartu RFID admin atau pengguna.



Gambar 4.3 Rangkaian elektronik pada *gateway*

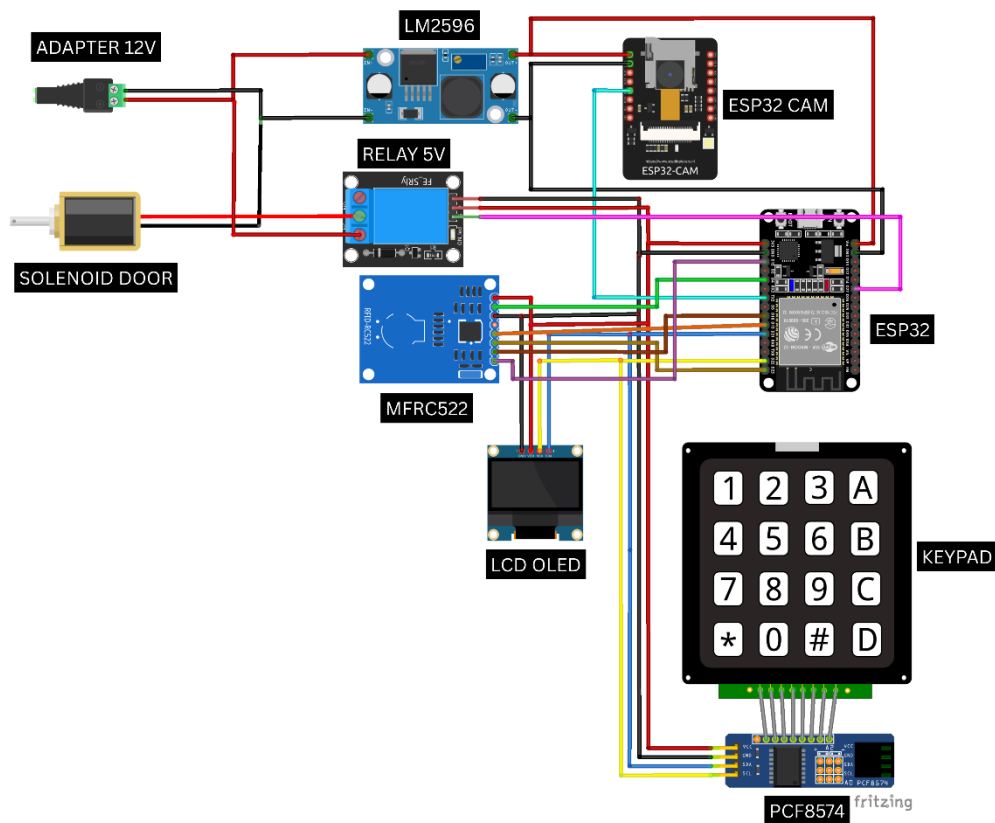
Selain difungsikan pada *gateway*, mikrokontroler ESP32 juga digunakan sebagai pusat kendali pada setiap *node* di berbagai pintu. Pada bagian *node* ini, ESP32 sama sekali tidak bertugas untuk menerima citra wajah, mengolah data, ataupun membaca data MFA secara mandiri. ESP32 di *node* hanya bertugas mendeteksi interaksi fisik awal dan mengirimkan *trigger* kepada modul ESP32-CAM. Setelah dipicu, ESP32-CAM akan langsung mengambil citra wajah dan mengirimkannya secara nirkabel menuju *bridge* untuk diteruskan ke *gateway*. Di samping



Hak Cipta :

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

memberikan *trigger*, ESP32 pada *node* meneruskan input mentah dari *keypad* atau RFID menuju *gateway*, menampilkan informasi pada layar LCD/OLED, serta mengeksekusi kendali aktuator pintu murni berdasarkan instruksi balasan yang telah dibaca dan diverifikasi oleh *gateway*.



Gambar 4.4 Detail rangkaian node

Berdasarkan Gambar 4.4, desain perangkat keras *node* dirancang menjadi satu kesatuan modul terintegrasi untuk alasan efisiensi ruang dan manajemen daya. Modul ini menggabungkan seluruh unit kontroler, antarmuka pengguna, dan sirkuit daya, yang terdiri dari ESP32 sebagai mikrokontroler utama, ESP32-CAM untuk pengambilan citra wajah (autentikasi utama), modul RFID RC522 (MFRC522), komponen keypad 4x4 untuk input PIN, modul layar LCD/OLED, dan sebuah I2C expander PCF8574 untuk meminimalkan penggunaan pin GPIO pada mikrokontroler.



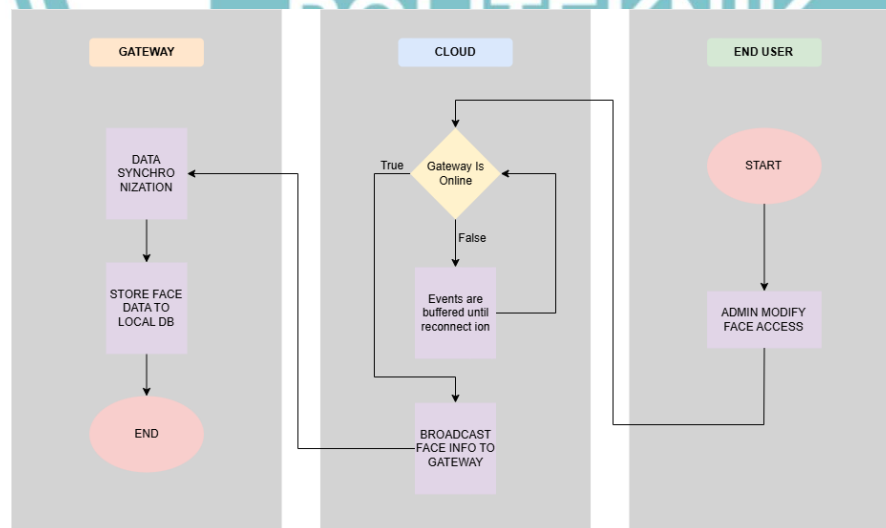
Hak Cipta :

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

Di dalam modul terintegrasi ini, terdapat pula sirkuit *stepdown* LM2596 untuk menyuplai daya stabil 5V ke mikrokontroler dan sensor, serta sebuah relay 5V untuk mengatur aliran arus listrik 12V dari adaptor menuju aktuator solenoid *door lock*. Dengan desain terpadu ini, seluruh komponen autentikasi, pengolahan data, regulasi tegangan, hingga sirkuit pemicu aktuator terhubung langsung dalam satu rangkaian tanpa memerlukan pemisahan fisik modul maupun kabel ethernet tambahan untuk mentransmisikan daya dan sinyal kontrol.

4.2.4 Sistem Sinkronisasi

Gambar 4.5 mengilustrasikan alur proses sinkronisasi data autentikasi wajah antara administrator, layanan *cloud*, dan perangkat *gateway*. Proses sinkronisasi ini dirancang untuk berjalan secara *real-time*. Setiap kali administrator melakukan modifikasi hak akses, seperti mendaftarkan data *template* wajah pengguna baru, mengubah pengaturan, atau mencabut hak akses pengguna dari suatu ruangan, perubahan tersebut akan diteruskan ke *cloud* untuk segera didistribusikan (*broadcast face info*) menuju *gateway*. Untuk memastikan pengiriman data yang presisi dan cepat, proses sinkronisasi ini dapat memanfaatkan protokol pengiriman pesan antarsistem (AMQP) dengan metode *direct exchange*.



Gambar 4.5 Proses Sinkronisasi

Dalam penerapan arsitektur *hybrid edge-cloud control*, ketersediaan koneksi internet menjadi faktor yang dikelola secara sistematis. Sistem pada *cloud* akan



Hak Cipta :

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

memeriksa status koneksi *gateway* terlebih dahulu. Apabila *gateway* terdeteksi sedang dalam kondisi *offline*, pesan sinkronisasi tersebut tidak akan hilang, melainkan ditampung sementara di dalam antrean *server* RabbitMQ hingga koneksi jaringan kembali pulih. Sebaliknya, jika *gateway* berstatus *online*, data akan langsung diterima oleh *gateway* untuk *data synchronization*. *Gateway* kemudian mengeksekusi perintah tersebut dengan menambahkan, memperbarui, atau menghapus data wajah langsung ke dalam basis data lokal. Melalui mekanisme *acknowledgement* atau konfirmasi penerimaan pesan, sistem memastikan *gateway* di tingkat *edge* selalu memiliki pangkalan data autentikasi wajah paling mutakhir untuk memproses pembukaan pintu secara mandiri.

4.2.5 Proses Autentikasi

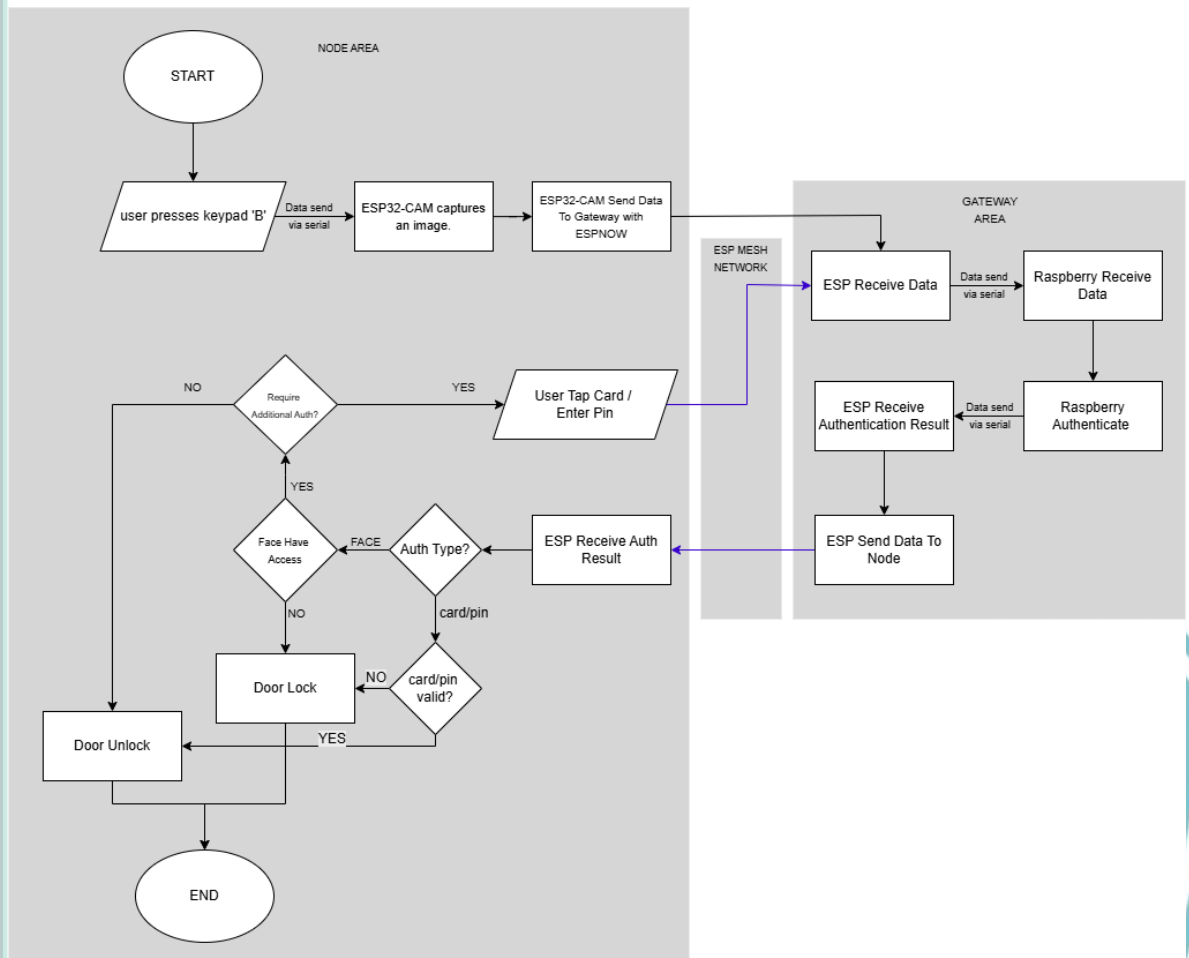
Gambar 4.6 menunjukkan alur autentikasi pengguna pada sistem yang diawali ketika pengguna menekan tombol 'B' pada *keypad* di area *node*. Data instruksi tersebut dikirimkan melalui komunikasi serial menuju ESP32-CAM, yang kemudian memicu modul kamera untuk mengambil gambar wajah pengguna. Data gambar tersebut lalu dikirimkan menuju *gateway* menggunakan protokol ESP-NOW melalui infrastruktur *ESP Mesh Network*. Pada area *gateway*, modul ESP akan menerima data tersebut dan meneruskannya menuju Raspberry Pi untuk keperluan proses autentikasi melalui komunikasi serial. Raspberry Pi kemudian akan melakukan autentikasi wajah, dan hasil dari autentikasi tersebut akan dikirimkan kembali menuju modul ESP *gateway* via komunikasi serial. Modul ESP *gateway* yang menerima data hasil autentikasi tersebut selanjutnya akan mengirimkannya kembali menuju *node* pengirim melalui *ESP Mesh Network*.



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

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



Gambar 4.6 Alur Autentikasi Pengguna

Lebih lanjut pada sisi *node*, modul ESP akan menerima data balasan dan mengecek tipe autentikasinya. Jika tipe autentikasi adalah wajah dan sistem menyatakan pengguna memiliki akses, alur akan berlanjut pada pengecekan apakah sistem memerlukan autentikasi tambahan. Jika tidak diperlukan, pintu akan langsung terbuka. Namun, jika autentikasi tambahan diwajibkan, sistem akan meminta pengguna untuk melakukan *tap* kartu atau memasukkan PIN. Data masukan ini akan kembali dikirimkan menuju *gateway* melalui *ESP Mesh Network* untuk divalidasi oleh Raspberry Pi. Ketika *node* menerima balasan hasil autentikasi untuk tipe kartu atau PIN, sistem akan memverifikasi validitasnya. Jika data yang divalidasi tersebut sesuai (*valid*), maka pintu ruangan bisa dibuka. Sebaliknya, baik pada tahap awal autentikasi wajah maupun tahap kartu/PIN, jika pengguna terdeteksi tidak memiliki akses atau data tidak valid, maka pintu akan tetap terkunci.



Hak Cipta :

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

Tahap implementasi sistem merupakan proses penerjemahan hasil perancangan arsitektur dan skema elektronik ke dalam bentuk fisik nyata. Implementasi ini mencakup penyusunan komponen perangkat keras (*hardware*) pada perangkat *node*, *bridge*, dan *gateway* serta penulisan kode program (*software*) baik untuk *firmware* mikrokontroler berbasis C/C++ maupun aplikasi pengendali utama lokal berbasis Python di Raspberry Pi. Pembagian implementasi ini dibagi ke dalam dua fokus utama, yaitu implementasi sistem perangkat lunak (*software*) yang mengatur logika aliran data dan sinkronisasi, serta implementasi sistem perangkat keras (*hardware*) yang mencakup tata letak fisik rangkaian elektronik dan integrasi mekanis sistem *smart door lock*.

4.3.1 Implementasi Sistem Perangkat Lunak

Implementasi sistem perangkat lunak memegang peranan krusial dalam mengatur alur logika verifikasi, komunikasi nirkabel, serta sinkronisasi data biometrik secara *real-time*. Perangkat lunak yang dibangun terdiri dari *firmware* pada sisi perangkat keras (ESP32 *Node*, ESP32-CAM, dan ESP32 *Bridge*) yang ditulis dengan bahasa C/C++, aplikasi *gateway* lokal berbasis Python yang mengelola pencocokan wajah menggunakan *OpenCV* dan *MobileFaceNet*, serta *database* lokal SQLite. Selain itu, bagian ini juga mencakup implementasi daemon sinkronisasi menggunakan protokol AMQP melalui broker RabbitMQ serta *dashboard user interface* untuk memudahkan administrator melakukan monitoring dan *enrollment* wajah pengguna baru secara langsung.

4.3.1.1 Implementasi Sistem Firmware Pada Gateway

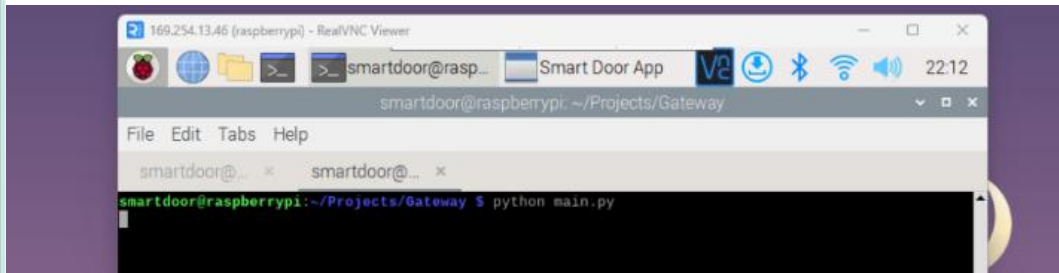
a. Setup Gateway

Proses konfigurasi *gateway* bertujuan untuk mempersiapkan perangkat keras agar dapat terintegrasi dengan layanan komputasi awan (*cloud*). Tahapan inisialisasi ini cukup dieksekusi satu kali. Prosedur tersebut dimulai dengan menjalankan perintah `python main.py`, sebagaimana yang diilustrasikan pada Gambar 4.7



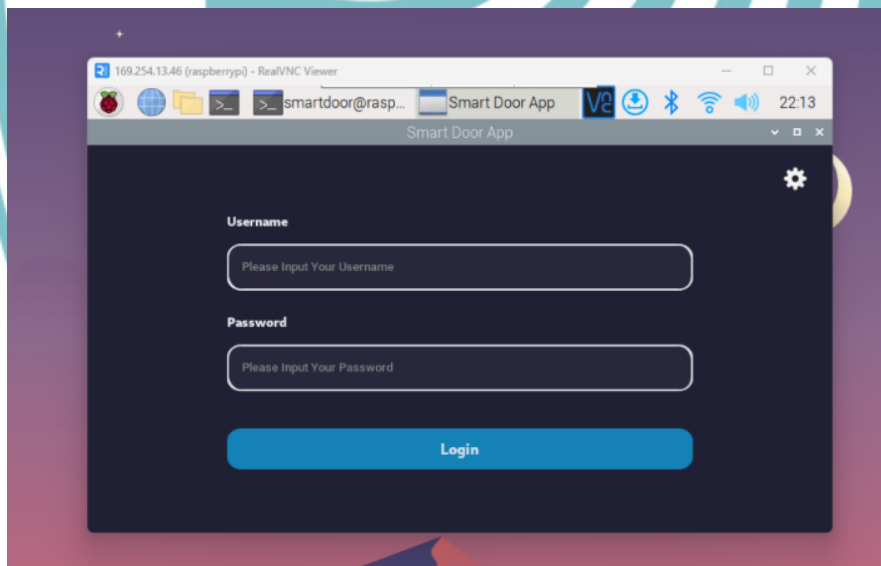
Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, 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 4.7 Mengaktifkan aplikasi gateway

Aplikasi akan beroperasi dan menampilkan antarmuka pengguna (*user interface*) sebagaimana diilustrasikan pada Gambar 4.8 Sebelum melakukan proses autentikasi menggunakan *username* dan *password*, administrator diwajibkan untuk menginput *API ID* dan *API KEY*. Form pengisian kredensial tersebut dapat diakses oleh administrator dengan mengeklik ikon roda gigi pada sudut kanan antarmuka aplikasi.



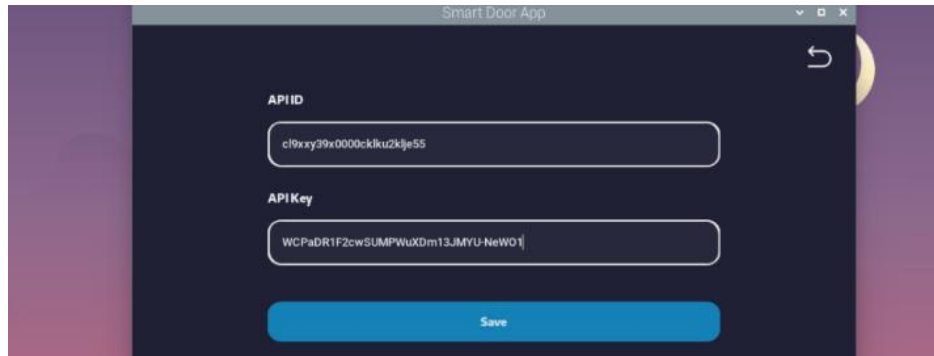
Gambar 4.8 Halaman login pada aplikasi gateway

Setelah mengeklik ikon roda gigi, administrator akan dialihkan menuju halaman konfigurasi *API ID* dan *API KEY*, sebagaimana diilustrasikan pada Gambar 4.9 Pada tahap ini, administrator diwajibkan untuk menginput *API ID* beserta *API KEY* yang diperoleh dari halaman situs web "Daftar *API key*". Selanjutnya, proses penyimpanan kredensial tersebut diselesaikan dengan mengeklik tombol "Save".



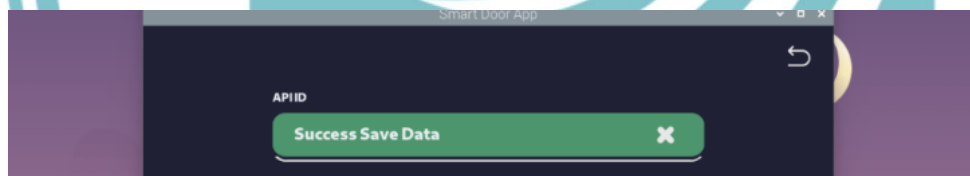
Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, 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 4.9 Pengaturan API IDE dan API KEY

Apabila proses konfigurasi API ID dan API KEY berhasil diselesaikan, sistem akan menampilkan *toast notification* sebagaimana diilustrasikan pada Gambar 4.10. Setelah data kredensial tersebut berhasil dimasukkan, administrator diwajibkan untuk menutup dan menjalankan ulang aplikasi. Tindakan ini bertujuan untuk memperbarui seluruh komponen sistem yang membutuhkan *secret key* tersebut guna membangun konektivitas dengan *server*.



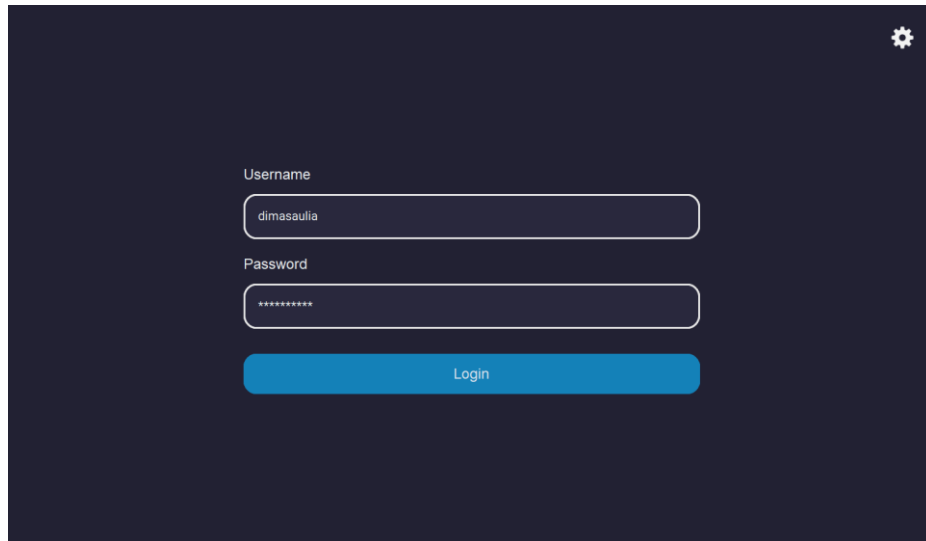
Gambar 4.10 Proses menyimpan API ID dan API KEY

Setelah aplikasi dijalankan ulang (*restart*), administrator akan diarahkan menuju halaman *login*, sebagaimana diilustrasikan pada Gambar 4.11. Fitur autentikasi ini diterapkan untuk memastikan bahwa aplikasi *gateway* terproteksi dari akses pihak yang tidak berwenang. Untuk memasuki sistem tersebut, diperlukan *username* dan *password* yang telah terdaftar pada layanan komputasi awan (*cloud*). Hak akses aplikasi ini dibatasi secara eksklusif bagi pengguna (*user*) yang memiliki peran (*role*) sebagai *Super Admin*, *Operator*, dan *Admin Teknis*.



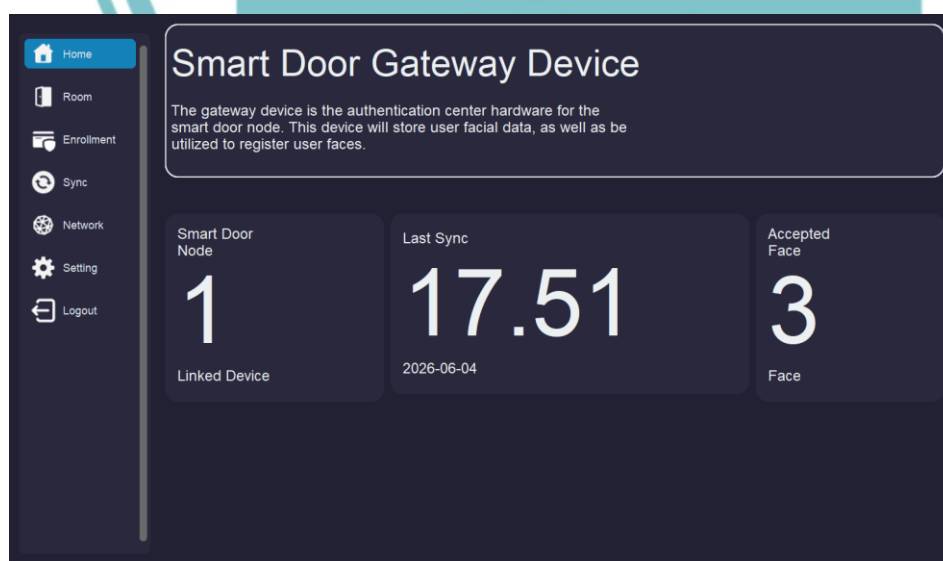
Hak Cipta :

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

Pasca penyelesaian proses autentikasi, administrator akan diarahkan menuju antarmuka utama aplikasi *gateway* sebagaimana diilustrasikan pada Gambar 4.12. Antarmuka tersebut secara struktural terdiri dari panel navigasi di sisi kiri yang memuat berbagai sub-menu fungsional, serta area konten utama di sisi kanan. Pada halaman *home*, sistem menyajikan informasi deskriptif mengenai fungsi perangkat *gateway* sebagai pusat autentikasi. Selain itu, halaman ini juga memuat beberapa panel informasi yang mengindikasikan jumlah perangkat *node* pintu pintar yang terhubung, waktu sinkronisasi terakhir, serta total data wajah pengguna yang telah tersimpan di dalam *database*.



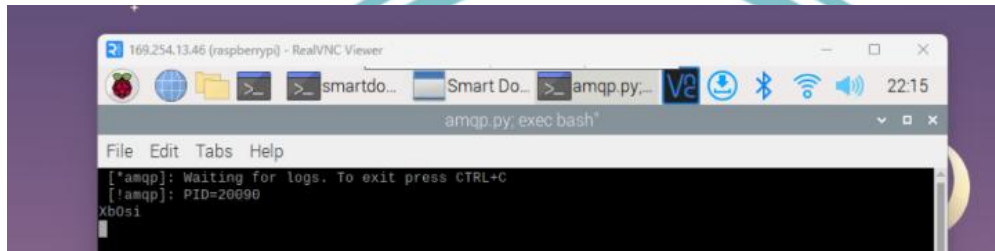
Gambar 4.12 Halaman home aplikasi gateway



Hak Cipta :

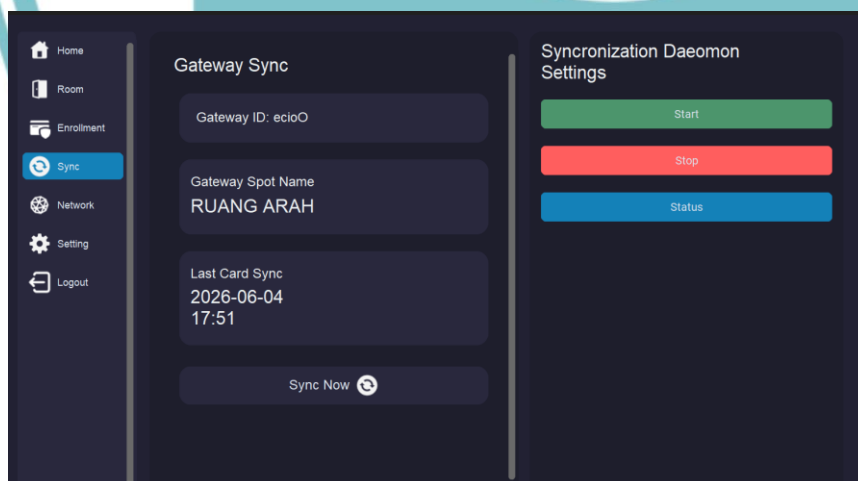
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan 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 autentikasi berhasil, sistem secara otomatis melakukan pemeriksaan di latar belakang untuk memverifikasi status operasi layanan sinkronisasi. Apabila layanan tersebut terdeteksi belum aktif, sistem akan langsung menginisiasi jalannya sinkronisasi sebagaimana ditunjukkan pada Gambar 4.13 Layanan sinkronisasi ini dirancang untuk terus beroperasi secara independen meskipun antarmuka aplikasi utama telah ditutup.



Gambar 4.13 Authentication daemon

Untuk mengetahui ID dari sebuah gateway administrator dapat masuk ke tab sync. Pada halaman tersebut akan terdapat kolom “Gateway ID” seperti yang tampak Gambar 4.14 Pada menu tersebut juga akan muncul nama gateway spot yang terhubung, serta waktu terakhir sinkronisasi.



Gambar 4.14 Menu sinkronisasi pada aplikasi gateway

b. Setup child node dan pembuatan ruangan

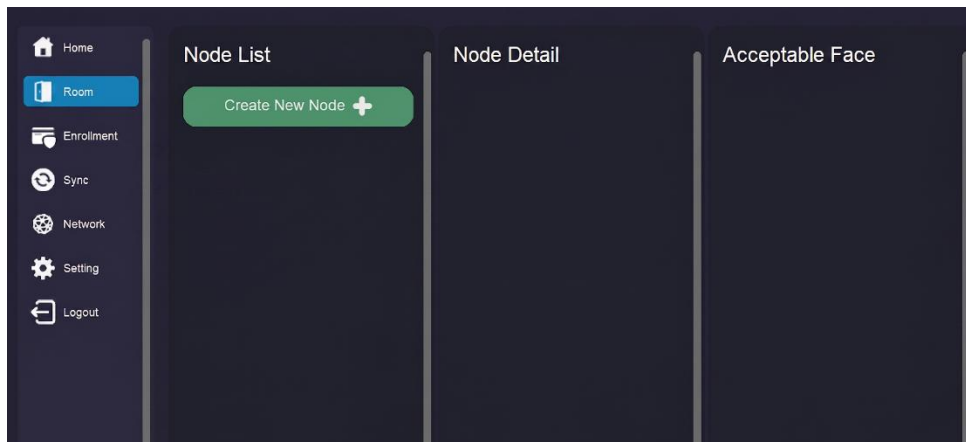
Administrator dapat meninjau daftar *child node* yang tersimpan di dalam *database* lokal melalui akses menu "Room". Selanjutnya, proses penambahan *child node* baru



Hak Cipta :

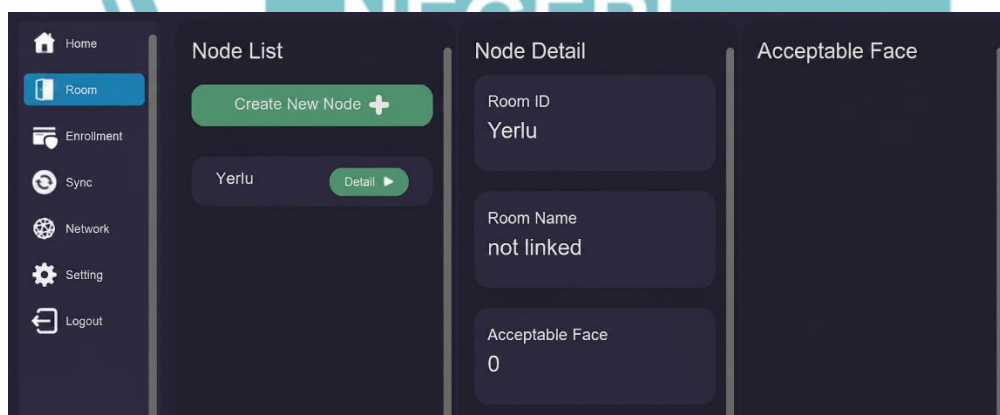
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan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

dapat dilakukan oleh administrator dengan memilih opsi "*Create Node*" sebagaimana diilustrasikan pada Gambar 4.15



Gambar 4.15 Menu room

Setelah proses penambahan *child node* baru berhasil diselesaikan, identitas dari *child node* tersebut akan terdaftar pada kolom *Node List* sebagaimana diilustrasikan pada Gambar 4.16 Guna meninjau informasi spesifik dari masing-masing *child node*, administrator dapat mengklik tombol *Detail* yang ditandai dengan ikon panah ke kanan. Eksekusi tindakan tersebut akan memicu sistem untuk menampilkan rincian informasi *node* pada kolom *Node Detail*, sekaligus memuat daftar entitas wajah yang telah terotorisasi pada kolom *Acceptable Face*.



Gambar 4.16 Menu “Room” ketika berhasil membuat child node

Setelah *child node* berhasil dibuat, administrator dapat menautkan perangkat tersebut dengan suatu ruangan tertentu, sebagaimana diuraikan pada bagian Proses Menambahkan Ruangan. Pasca penautan, sistem akan mengeksekusi sinkronisasi

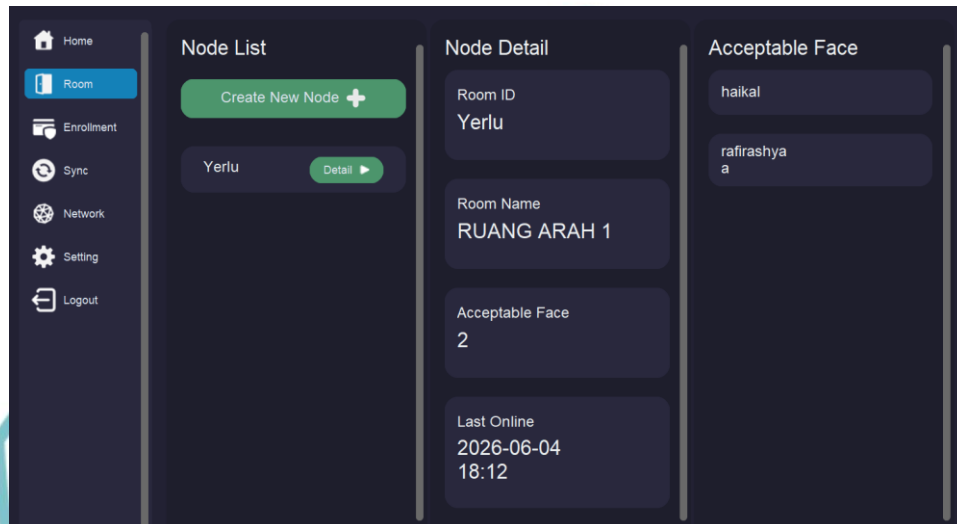


© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

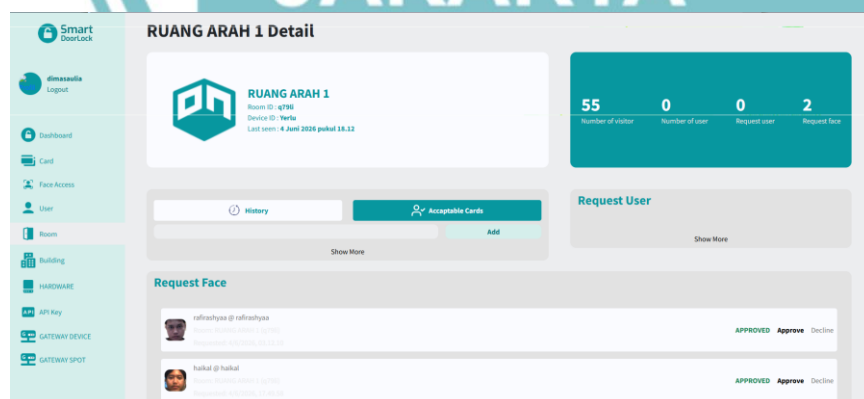
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan 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 secara otomatis. Apabila proses sinkronisasi tersebut berhasil diselesaikan, keterangan pada kolom *Room Name* akan diperbarui sesuai dengan nama yang dimasukkan pada tahap pembuatan ruangan, sebagaimana diilustrasikan pada Gambar 4.17



Gambar 4.17 Detail node list setelah pembuatan ruangan

Melalui halaman menu *Room* pada panel web, administrator dapat mengotorisasi akses biometrik pengguna secara manual. Langkah ini dieksekusi dengan memasukkan *username* pada kolom pencarian yang tersedia di bawah bagian pengelolaan hak akses, lalu menekan tombol *Add*. Setelah data wajah berhasil ditautkan dengan ruangan terkait, sistem akan memperbarui tampilan halaman web seperti yang ditunjukkan pada Gambar 4.18.



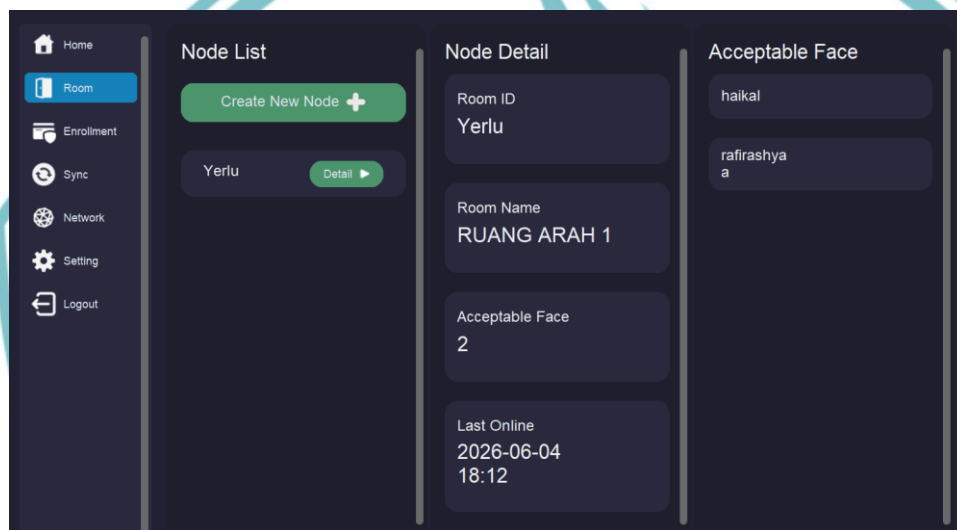
Gambar 4.18 Proses menabahkan akses wajah pada suatu ruangan



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, 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 administrator berhasil mendaftarkan data wajah, sistem akan secara langsung mentransmisikan instruksi tersebut menuju perangkat *gateway*. Pembaruan informasi ini dapat ditinjau lebih lanjut oleh administrator dengan mengeklik tombol *Detail* pada *node* yang bersangkutan. Identitas pengguna yang telah mendapatkan otorisasi akan terdaftar secara terperinci pada kolom *Acceptable Face*. Sementara itu, total akumulasi data wajah yang memiliki hak akses untuk ruangan tersebut akan direpresentasikan pada panel *Acceptable Face* di dalam area *Node Detail*, sebagaimana diilustrasikan pada Gambar 4.19.



Gambar 4.19 Daftar wajah yang memiliki akses ke ruangan

c. Mendaftarkan Wajah Pengguna

Proses pendaftaran identitas pengguna beserta data autentikasinya dapat diakses oleh administrator melalui menu *Enrollment*. Sebagaimana diilustrasikan pada Gambar 4.20, terdapat tiga kolom pengisian utama pada antarmuka tersebut. Kolom *USER ID & NAME* difungsikan untuk menetapkan identitas pemilik data. Selanjutnya, terdapat kolom *Card ID* dan *Card PIN* yang bersifat opsional. Terkait dengan penerapan keamanan berlapis atau *Multi-Factor Authentication*, aktivasi fitur ini tidak dipaksakan secara otomatis, melainkan dikonfigurasi secara manual melalui kotak centang *MFA Requirement*. Hal ini memberikan fleksibilitas di mana kewajiban penggunaan kartu maupun PIN dapat disesuaikan bergantung pada kebutuhan keamanan masing-masing pengguna. Rangkaian proses pendaftaran ini kemudian diselesaikan dengan perekaman biometrik wajah.



Hak Cipta :

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

Gambar 4.20 Menu Enrollment

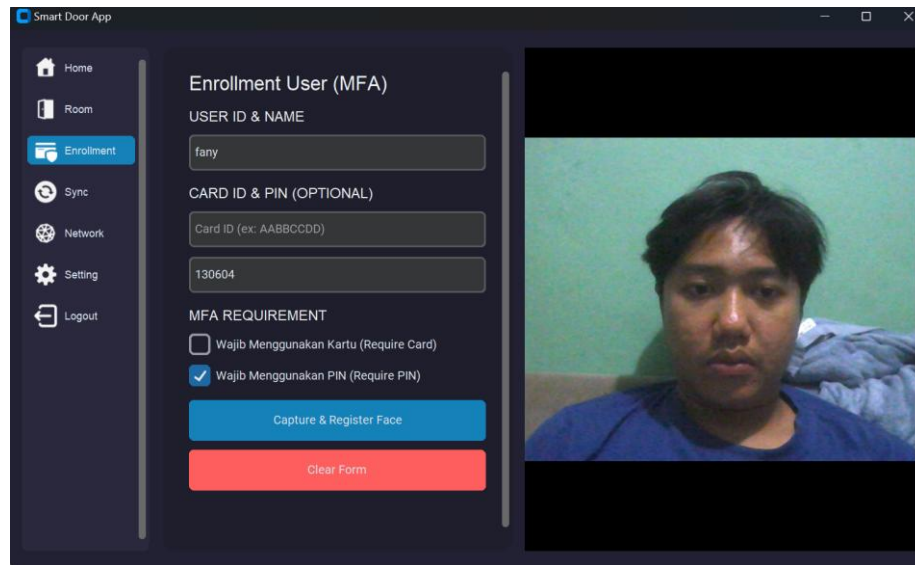
Sebagai bentuk implementasi dari tahapan pendaftaran, Gambar 4.20 mengilustrasikan proses pengisian kredensial secara spesifik untuk pengguna baru. Administrator memulai proses ini dengan memasukkan identitas pada kolom *USER ID & NAME*, dilanjutkan dengan mendefinisikan kombinasi kata sandi pada kolom *Card PIN*. Sesuai dengan pengaturan keamanan yang diterapkan untuk pengguna tersebut, administrator hanya mengaktifkan syarat wajib penggunaan PIN pada bagian *MFA Requirement* tanpa mengharuskan penggunaan kartu akses fisik. Rangkaian pendaftaran kredensial ini kemudian difinalisasi dengan menekan tombol *Capture & Register Face* guna mengeksekusi proses perekaman dan penyimpanan data biometrik wajah pengguna secara langsung ke dalam basis data sistem.



© Hak Cipta milik Politeknik Negeri Jakarta

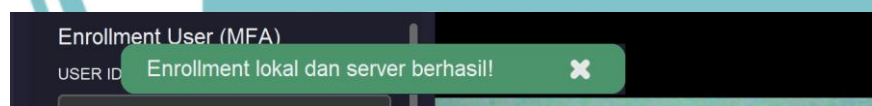
Hak Cipta :

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



Gambar 4.21 Proses enrolment

Apabila proses perekaman data biometrik dan kredensial pengguna telah berhasil dieksekusi, sistem akan menampilkan *toast notification* yang menyatakan keberhasilan pendaftaran pada tingkat lokal maupun server, sebagaimana direpresentasikan pada Gambar 4.22. Indikator visual berwarna hijau tersebut mengonfirmasi bahwa identitas pengguna beserta prasyarat autentikasinya telah divalidasi, tersinkronisasi, dan direkam secara permanen, baik di dalam basis data perangkat *gateway* maupun pada layanan komputasi awan terpusat.



Gambar 4.22 Proses enrolment berhasil

d. Mengaktifkan dan pengecekan sinkronisasi

Perangkat *gateway* secara berkala mentransmisikan sinyal *ping* menuju layanan *cloud* dengan interval waktu sepuluh menit. Dalam kondisi malfungsi jaringan yang mengakibatkan terputusnya koneksi sinkronisasi, interval *ping* tersebut akan dipercepat menjadi lima menit. Sesaat setelah *gateway* berhasil mendapatkan respons kembali dari *cloud*, sistem akan memverifikasi status operasional *authentication daemon*. Apabila layanan tersebut terdeteksi berhenti, sistem akan secara otomatis melakukan inisiasi ulang. Selanjutnya, guna memantau aktivitas sinkronisasi, administrator dapat mengakses menu *Sync*. Pada kolom kedua

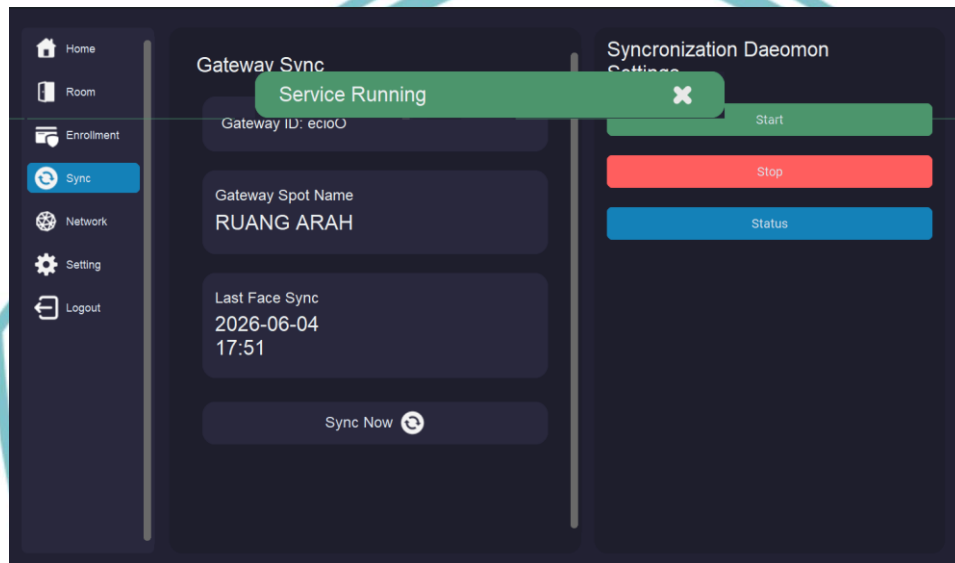


© Hak Cipta milik Politeknik Negeri Jakarta

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

antarmuka tersebut, administrator dapat memilih opsi *Status* di bawah bagian *Synchronization Daemon Settings*. Indikator *Service Running* akan ditampilkan apabila layanan beroperasi secara normal, sebagaimana diilustrasikan pada Gambar 4.23 Melalui panel menu yang sama, sistem juga menyediakan kontrol operasional bagi administrator untuk menghentikan maupun mengaktifkan layanan sinkronisasi secara manual.



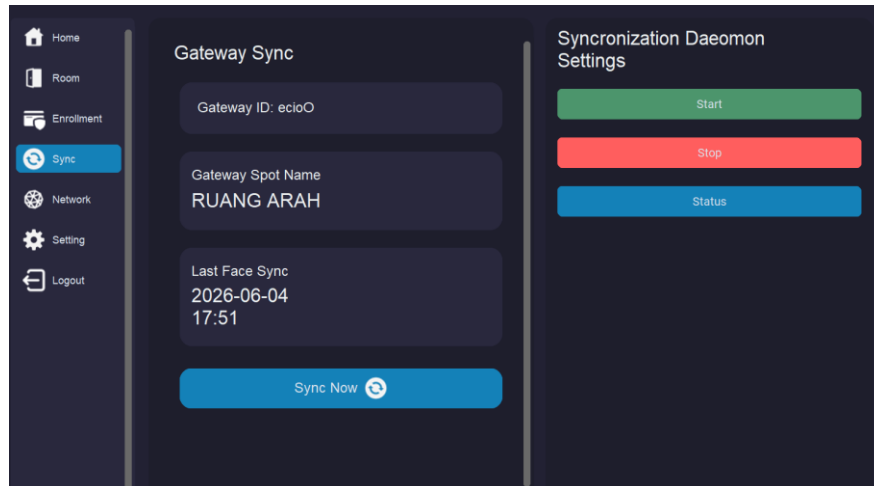
Gambar 4.23 Pengecekan layanan sinkronisasi

Pengguna juga memiliki opsi untuk menginisiasi proses sinkronisasi secara manual melalui tombol *Sync Now*. Opsi ini terletak pada bagian bawah kolom *Gateway Sync*, sebagaimana diilustrasikan pada Gambar 4.24 Eksekusi pada tombol tersebut akan memicu sistem untuk mengirimkan *request* pembaruan data ke layanan *cloud* melalui transmisi protokol HTTP.



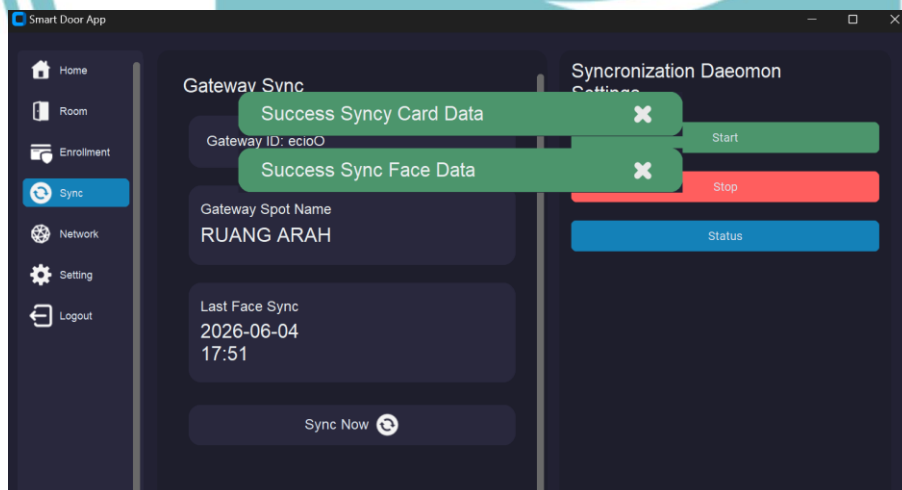
Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, 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 4.24 Proses sinkronisasi manual

Apabila proses sinkronisasi berhasil diselesaikan, sistem akan menampilkan notifikasi yang mengonfirmasi bahwa seluruh data telah disinkronisasikan, sebagaimana diilustrasikan pada Gambar 4.25. Keberhasilan proses ini akan berdampak pada pembaruan indikator *Last Face Sync*, yang berfungsi untuk merekam dan menampilkan catatan waktu paling mutakhir dari pelaksanaan sinkronisasi data wajah tersebut.



Gambar 4.25 proses sinkronisasi berhasil dilakukan

e. Menyalakan Fitur ESP Mesh Network

Guna mengaktifkan *ESP Mesh Network*, administrator perlu mengakses menu *Network*. Pada antarmuka tersebut, sistem akan menyajikan ketersediaan *port* pada kolom *Mesh AP Port* (sebagai contoh COM6) yang difungsikan sebagai jalur

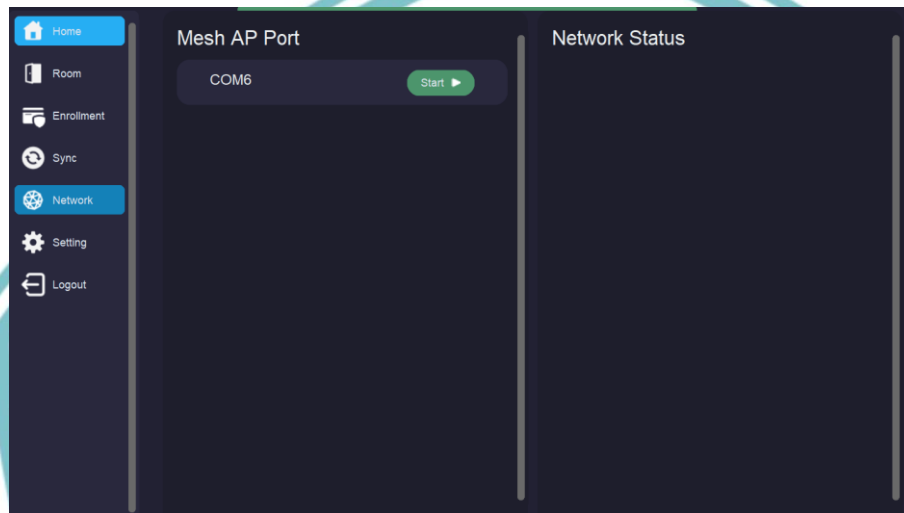


© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

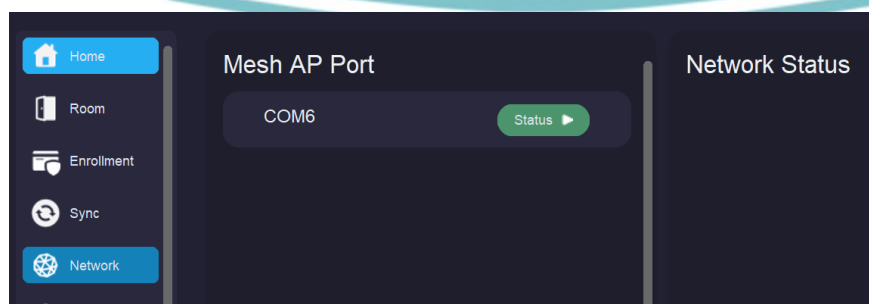
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan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

komunikasi dengan perangkat *bridge*. Perangkat *bridge* inilah yang selanjutnya akan menghubungkan *gateway* ke dalam *ESP Mesh Network*, sebagaimana diilustrasikan pada Gambar 4.26 Untuk menginisiasi komunikasi jaringan tersebut, administrator dapat mengeklik tombol *Start* pada *port* yang dituju. Secara arsitektural, satu sistem *gateway* ini dirancang agar memiliki kapabilitas untuk terhubung dengan beberapa *bridge* secara bersamaan.



Gambar 4.26 Menu ESP Mesh Network

Apabila inisiasi jaringan berhasil dieksekusi, antarmuka akan memperbarui tombol pada *port* tersebut menjadi tombol *Status*. Administrator dapat meninjau rincian informasi jaringan dengan mengeklik menu tersebut. Data yang ditampilkan mencakup kredensial *SSID* dan *Password* dari *mesh network*. Kredensial jaringan ini selanjutnya wajib diinput ke dalam konfigurasi masing-masing perangkat *node*, sebagaimana diilustrasikan pada Gambar 4.27 Panduan operasional mengenai pengisian data tersebut diuraikan secara lebih rinci pada bagian *Setup Node*.



Gambar 4.27 Detail Network

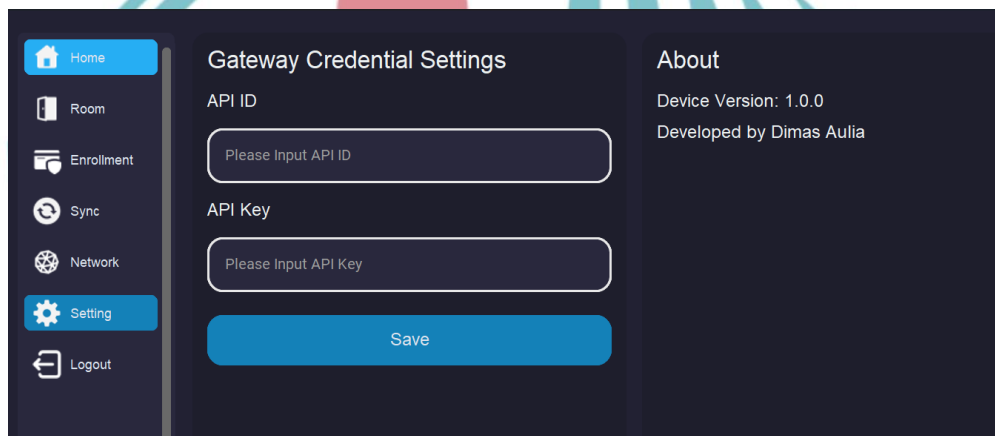


Hak Cipta :

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

f. Pengaturan API Key

Administrator diberikan otoritas untuk melakukan konfigurasi ulang *API Key* melalui menu *Setting*, sebagaimana diilustrasikan pada Gambar 4.28 Pada antarmuka ini, administrator dapat menginput kembali kredensial *API ID* beserta *API Key*, kemudian mengeklik tombol *Save* untuk menyimpan pembaruan konfigurasi. Guna memastikan pembaruan data tersebut diimplementasikan secara menyeluruh oleh sistem, administrator diwajibkan untuk menutup lalu menjalankan ulang aplikasi. Di samping fungsionalitas konfigurasi kredensial, halaman pengaturan ini juga menyajikan informasi teknis mengenai versi perangkat lunak yang sedang beroperasi.



Gambar 4.28 Menu setting

g. Logout

Menu logout digunakan ketika aktivitas di gateway selesai dilakukan. Menu ini akan membuat tampilan aplikasi menjadi halaman login. Sistem juga memiliki mekanisme untuk logout secara otomatis jika tidak terdapat aktivitas selama 30 menit.

4.3.1.2 Implementasi Sistem Firmware Pada Node ESP32

Sistem *firmware* pada perangkat ini dikembangkan menggunakan bahasa pemrograman C. Perangkat lunak tersebut dirancang dengan serangkaian kapabilitas fungsional utama, yang meliputi konfigurasi kredensial *SSID* dan kata sandi secara nirkabel, kapabilitas pembaruan sistem *Over-The-Air* atau OTA, serta mekanisme interkoneksi otomatis menuju perangkat *bridge*. Lebih lanjut, sistem ini



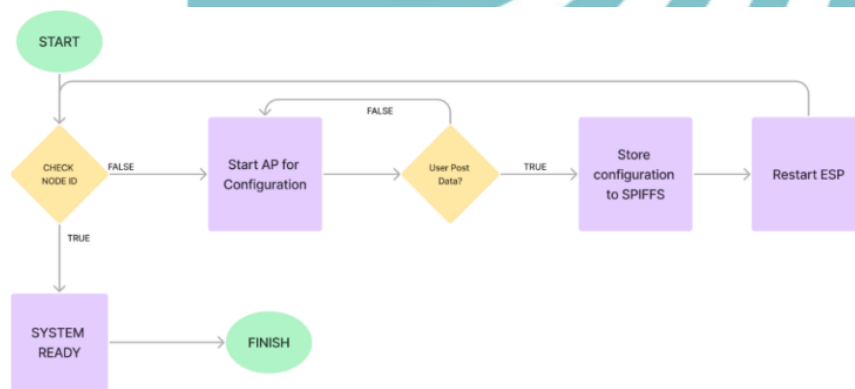
Hak Cipta :

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

juga terintegrasi dengan protokol transmisi dan penerimaan data, beserta fungsionalitas aktuasi untuk mengendalikan mekanisme penguncian pintu.

1) Setup Node

Gambar 4.29 mengilustrasikan tahapan konfigurasi awal pada perangkat *node*. Prosedur ini diimplementasikan guna memastikan *node* dapat terintegrasi ke dalam ekosistem *ESP mesh network*. Pada saat sistem diaktifkan untuk pertama kalinya, mikrokontroler akan mengeksekusi verifikasi terhadap ketersediaan kredensial jaringan, yang meliputi *SSID* dan kata sandi. Apabila data tersebut tidak ditemukan di dalam memori, sistem akan secara otomatis menginisiasi sebuah *Access Point* mandiri yang memancarkan jaringan dengan nama *ESP-MESH-MANAGER*. Sebagai tindak lanjut, administrator diwajibkan untuk mengoneksikan perangkat komputasi, baik berupa laptop maupun ponsel pintar, menuju antarmuka jaringan tersebut guna menyelesaikan proses pengaturan.



Gambar 4.29 Proses setup node

Setelah terhubung dengan antarmuka *Access Point*, administrator dapat mengakses alamat IP 192.168.4.1 melalui peramban web untuk memasuki portal konfigurasi perangkat. Akses menuju alamat tersebut akan memuat halaman pengaturan sebagaimana diilustrasikan pada Gambar 4.30 Administrator kemudian diwajibkan untuk menginput seluruh parameter data yang dibutuhkan, dilanjutkan dengan mengklik tombol *Save*. Pasca eksekusi perintah penyimpanan tersebut, sistem akan merekam data kredensial jaringan secara permanen ke dalam memori *Serial Peripheral Interface Flash File System* atau *SPIFFS*. Rangkaian proses ini akan diakhiri dengan inisiasi mulai ulang secara otomatis pada perangkat, 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

selanjutnya akan berupaya untuk membangun interkoneksi menuju jaringan baru yang telah dikonfigurasi.

Gambar 4.30 Halaman konfigurasi node

2) Update Firmware Node

Sebagai tahapan awal untuk memperbaiki sistem *firmware* yang beroperasi, administrator diwajibkan untuk mengakses mode administratif. Proses inisiasi menuju mode tersebut dilakukan dengan menekan tombol simbol pagar pada antarmuka *keypad*. Tindakan ini akan memicu sistem untuk menampilkan layar autentikasi yang mengharuskan administrator untuk menginput PIN perangkat. Apabila verifikasi kredensial PIN tersebut dinyatakan valid, antarmuka layar LCD akan memunculkan indikator teks *ADMIN MODE*. Lebih lanjut, pada layar antarmuka administratif tersebut, sistem juga menyediakan informasi fungsional yang diperlukan untuk mengeksekusi pengaturan ulang pada *ESP mesh network*.

Gambar 4.31 Halaman update firmware



© 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

Di dalam antarmuka *ADMIN MODE*, administrator diwajibkan menekan tombol B guna mengeksekusi proses pengaturan ulang sistem. Pasca pengaturan ulang tersebut, administrator harus mengoneksikan perangkat komputasi ke *Access Point* yang dipancarkan oleh mikrokontroler. Selanjutnya, akses menuju portal konfigurasi dilakukan dengan menginput alamat IP 192.168.4.1 pada peramban web, yang akan mengarahkan administrator pada antarmuka sebagaimana diilustrasikan pada Gambar 4.31 Pada halaman tersebut, administrator perlu mengklik tombol *Update Firmware* untuk menavigasi ke menu pembaruan perangkat lunak sistem, yang akan memuat tampilan seperti pada Gambar 4.69. Administrator kemudian diharuskan untuk mengunggah berkas *firmware* mutakhir yang telah dikompilasi ke dalam format biner dengan ekstensi *.bin*. Setelah proses injeksi *firmware* baru tersebut berhasil diselesaikan, perangkat akan secara otomatis melakukan inisiasi mulai ulang, dan administrator diwajibkan untuk menginput kembali kredensial *SSID* beserta kata sandi dari *ESP mesh network*.

3) Proses terhubung ke jaringan

Apabila sistem berhasil mendeteksi keberadaan kredensial *SSID* dan kata sandi di dalam memori *SPIFFS*, perangkat akan berupaya membangun koneksi menuju jaringan terkait. Rincian mengenai implementasi kode sumber untuk mengeksekusi prosedur penggabungan jaringan tersebut dapat ditinjau pada Gambar 4.32 Pada fase ini, perangkat *node* akan mentransmisikan paket data bertipe *connectionstartup* menuju perangkat *bridge* atau *gateway*. Transmisi pesan permintaan tersebut dieksekusi secara periodik dengan interval waktu lima detik. Selama perangkat *bridge* belum memberikan balasan, fungsionalitas *node* akan dibatasi sehingga tidak dapat merespons proses *check-in*, dan sistem akan terus melanjutkan pengiriman pesan. Segera setelah perangkat *bridge* mengonfirmasi penerimaan transmisi dan memberikan respons balik, sistem akan memperbarui status konektivitas menjadi *ready*, yang mengindikasikan bahwa perangkat *node* telah terhubung sepenuhnya dan siap untuk dioperasikan.



Hak Cipta :

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

```
// When First Start Up try to connect to gateway
while (isConnectionReady == false && APStatus == false) {
    mesh.update();
    // MENGIRIM PESAN SETIAP 5 DETIK
    uint64_t now = millis();
    if (now - connectionStartupCheckTime > 5000) {
        Serial.println("[x]: Sending Connection Startup");
        connectionStartupCheckTime = millis();
        String msg = "{\"type\":\"connectionstartup\", \"source\":\"\" +
            DATA_NODE + "\", \"destination\" : \"\" + DATA_GATEWAY +
            +\"\"}\"";
        connectionStartupRTOChecker = millis();
        mesh.sendSingle(DATA_GATEWAY, msg);
        isWaitingForConnectionStartupResponse = true;
        if (now > bootTimestamp + 5000) {
            displayWaitingConnection();
        }
    }

    // INFO: RESET BUTTON
    if (digitalRead(RESET_BUTTON) == HIGH) {
        meshReset();
    }
}
```

Gambar 4.32 Proses bergabung ke jaringan ESP mesh

4.3.2 Implementasi Sistem Perangkat Keras

Implementasi sistem perangkat keras merupakan tahap perakitan dan pengintegrasian seluruh komponen fisik yang digunakan dalam sistem *smart door lock*. Komponen perangkat keras ini dibagi menjadi beberapa bagian utama, yaitu perangkat *gateway* yang berfungsi sebagai pusat pemrosesan lokal pada tingkat *edge*, perangkat *bridge* yang menjembatani komunikasi nirkabel dengan komunikasi serial, serta perangkat *node* yang terpasang pada pintu untuk mengendalikan aktuator *solenoid door lock* serta membaca input autentikasi dari pengguna. Penjelasan mengenai detail fisik, tata letak komponen, dan rangkaian elektronika dari masing-masing perangkat tersebut dipaparkan pada sub-bab berikut.

4.3.2.1 Implementasi Sistem Pada Gateway

Perangkat *gateway* berfungsi sebagai pusat pemrosesan data lokal, penyimpanan basis data autentikasi lokal (*SQLite*), serta koordinasi jaringan *mesh*. Wujud fisik dari implementasi perangkat keras *gateway* ini ditampilkan secara jelas pada Gambar 4.33, yang menyajikan tampak depan dari seluruh rangkaian komponen yang terintegrasi.



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

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



Gambar 4.33 Tampilan *gateway*

Gambar 4.33 mengilustrasikan tampilan depan dan atas dari *casing gateway* yang dikembangkan menggunakan basis komputer papan tunggal Raspberry Pi 4B. Di bagian tengah selubung, terpasang layar sentuh LCD berukuran tujuh inci yang menampilkan antarmuka pengguna grafis dari aplikasi *Smart Door App*. Layar penampil tersebut menampilkan panel *dashboard* utama yang menyajikan informasi status sistem secara *real-time*, seperti status koneksi jaringan *mesh* yang mendeteksi tiga perangkat *node* terhubung, waktu sinkronisasi data terakhir ke *cloud*, serta jumlah kartu RFID terdaftar yang aktif. Pada sisi kiri layar, terdapat menu navigasi vertikal yang mencakup pilihan *Home*, *Room*, *Card*, *Sync*, *Network*, *Setting*, dan *Logout*. Aplikasi ini berjalan di atas sistem operasi Raspberry Pi OS, yang ditunjukkan dengan keberadaan *taskbar* khas sistem operasi tersebut pada bagian atas layar.

Pada bagian atas *gateway*, diintegrasikan dua modul sensor utama secara eksternal. Di sudut kiri atas layar LCD, terdapat sebuah modul ESP32-CAM dengan sensor kamera OV2640 yang dilindungi wadah putih khusus. Modul ini diletakkan



© 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

menghadap ke depan untuk mengambil citra wajah pengguna saat administrator melakukan pendaftaran wajah secara lokal. Sementara itu, tepat di sisi atas tengah layar, terintegrasi sensor pembaca kartu RFID PN532. Desain penempatan sensor RFID di posisi tertinggi ini disengaja agar administrator atau pengguna dapat melakukan pemindaian kartu RFID dengan mudah dan ergonomis tanpa menghalangi pandangan ke arah layar LCD. Seluruh komponen ini terhubung secara internal dan ditenagai oleh catu daya stabil, serta memiliki jalur kabel data/daya USB dan ethernet yang keluar dari bagian bawah perangkat untuk menjamin keandalan koneksi jaringan lokal.

Perangkat *bridge* diimplementasikan untuk menjembatani keterbatasan komunikasi antara komputer papan tunggal Raspberry Pi (pada *gateway*) yang tidak mendukung protokol nirkabel bawaan mikrokontroler ESP. Tampilan fisik dari perangkat keras *bridge* ini diilustrasikan secara lebih rinci pada Gambar 4.34.



Gambar 4.34 Perangkat keras *bridge*

Perangkat *bridge* menggunakan mikrokontroler ESP32 yang dikonfigurasi dengan antena eksternal guna memperkuat jangkauan nirkabel dan meningkatkan stabilitas penerimaan sinyal di dalam bangunan bertingkat. *Bridge* bertindak sebagai penerjemah yang menerima paket data autentikasi dari seluruh perangkat *node* melalui jaringan nirkabel *ESP Mesh* dan *esp now*, kemudian meneruskan data



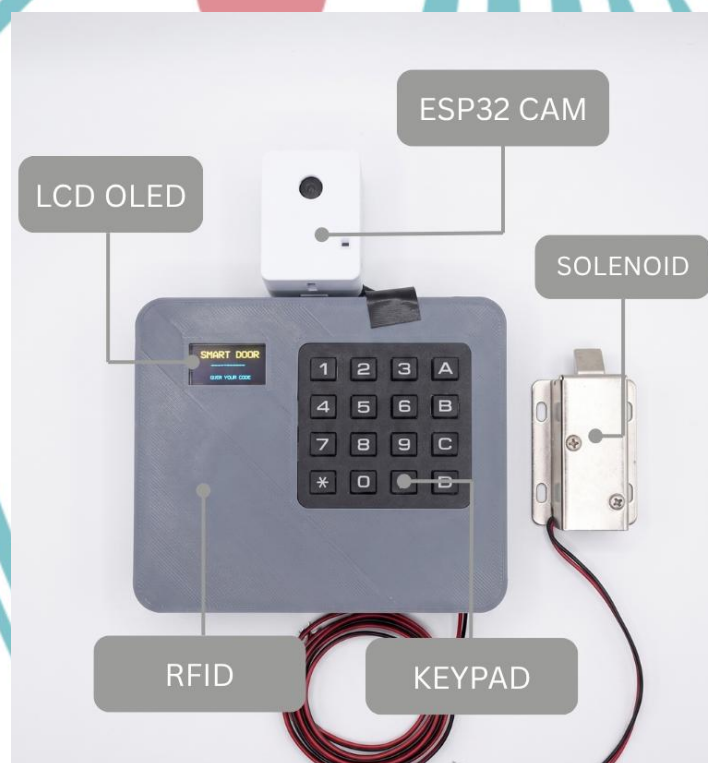
Hak Cipta :

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

tersebut ke *gateway* (Raspberry Pi) melalui koneksi serial fisik berbasis kabel USB. Penggunaan kabel USB ini juga berfungsi ganda sebagai penyuplai daya operasional untuk perangkat *bridge*. Dengan demikian, data autentikasi nirkabel dari *node* dapat diproses lebih lanjut oleh Raspberry Pi untuk verifikasi wajah dan pencocokan kredensial database lokal secara *real-time*.

4.3.2.2 Implementasi Sistem Pada Node ESP32

Perangkat *node* dipasang pada setiap pintu ruangan dosen untuk mengontrol akses masuk secara luring berdasarkan keputusan komputasi dari *gateway*. Tampilan depan dari perangkat keras *node* ini diilustrasikan pada Gambar 4.35.



Gambar 4.35 Perangkat keras *node*

Gambar 4.35 memperlihatkan wujud fisik luar dari perangkat *node*. Di bagian tampilan depan, terintegrasi beberapa komponen antarmuka fisik dan sensor untuk mendukung metode *Multi-Factor Authentication*. Pada tampilan depan bagian atas kiri, terdapat modul layar penampil LCD OLED berukuran 0,96 inci dengan resolusi 128x64 guna memberikan informasi kepada pengguna saat menggunakan sistem. Di sebelah kanan layar OLED, modul ESP32-CAM dipasang secara vertikal



Hak Cipta :

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

pada bagian atas. Kamera ini diposisikan dengan sudut horizontal menghadap lurus ke depan agar dapat menangkap citra wajah pengguna secara optimal ketika proses pemindaian wajah diaktifkan.

Di bagian tengah permukaan selubung *node*, terpasang tombol fisik berupa *keypad* matriks 4x4 dengan label tombol angka 0-9 dan karakter A-D serta simbol bintang dan pagar. Keypad ini berfungsi sebagai alat penginput kode keamanan PIN secara manual ketika pengguna memilih mode autentikasi berbasis PIN. Di balik permukaan casing abu-abu bagian bawah kiri, terpasang antena sensor RFID secara internal. Pengguna dapat menempelkan kartu akses mereka ke area tersebut untuk diverifikasi. Sebagai unit aktuator mekanis, *node* dihubungkan ke modul pengunci elektrik berupa *Solenoid Door Lock* 12V DC yang diletakkan secara terpisah di sisi dalam pintu dan dihubungkan secara elektrik melalui kabel ke modul *relay* di dalam *node* untuk mengaktifkan kunci pintu sesuai dengan perintah akses yang dikirim oleh *gateway*.

4.4 Pengujian

Tahapan pengujian dan evaluasi sistem dilakukan untuk mengukur serta menganalisis performa, keandalan, dan fungsionalitas dari sistem *smart door lock* berbasis *hybrid edge-cloud control* yang telah dibangun. Pengujian ini dibagi menjadi dua aspek utama, yaitu pengujian fungsionalitas melalui metode *black box testing* untuk memvalidasi kesesuaian fitur fisik, serta pengujian kinerja transmisi data dan sinkronisasi berdasarkan parameter *Quality of Service* (QoS). Pengukuran parameter QoS yang meliputi *delay / latency*, *jitter*, serta *packet loss* dilakukan untuk menilai efisiensi dan kestabilan komunikasi nirkabel (ESP-NOW dan ESP Mesh) serta sinkronisasi data biometrik antara *gateway* lokal dan *server cloud* menggunakan protokol AMQP. Seluruh hasil pengukuran ini akan dianalisis secara statistik untuk memberikan gambaran kuantitatif mengenai kinerja sistem dalam berbagai kondisi operasional.

4.4.1 Deskripsi Pengujian

Secara garis besar, metode pengujian yang akan diimplementasikan difokuskan pada pengujian sintesis. Pengujian ini dieksekusi berdasarkan skenario yang telah



© 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

dikonfigurasi secara spesifik guna memperoleh parameter yang mampu merepresentasikan performa serta kapabilitas optimal dari sistem yang telah dikembangkan. Guna mengakuisisi berbagai variasi data yang dapat merepresentasikan kemampuan sistem tersebut secara komprehensif, telah dirancang sejumlah skenario pengujian.

a) Pengujian Fungsionalitas

Pengujian fungsionalitas dilakukan guna mengevaluasi kinerja operasional dari setiap komponen dan fitur utama pada sistem *smart door lock*. Berbeda dengan pengujian kuantitatif lainnya yang menitikberatkan pada metrik performa jaringan, pengujian ini bertujuan untuk membuktikan bahwa seluruh logika alur sistem dapat merespons berbagai skenario masukan (*input*) sesuai dengan spesifikasi rancangan. Metode pengujian yang diimplementasikan adalah *black box testing*, di mana pengujian dieksekusi secara fungsional melalui interaksi langsung dengan perangkat keras dan perangkat lunak tanpa memedulikan arsitektur internal atau struktur kode program. Parameter keberhasilan dinilai berdasarkan kesesuaian antara respon aktual sistem, seperti pembukaan kunci pintu, perubahan status layar, dan perekaman log dengan hasil yang diharapkan.

b) One Trip Data Transmission

Pengujian transmisi satu arah atau *one trip transmission* dilaksanakan guna mengukur durasi waktu jeda transmisi atau *delay* dari paket data yang ditransmisikan dari perangkat *node* menuju *gateway* secara spesifik melalui protokol *ESP-NOW*. Gambar 4.36 mendemonstrasikan prosedur pengukuran waktu transmisi tersebut. Perangkat *node* akan menginisiasi proses pengiriman data. Pada saat transmisi berlangsung, waktu pengiriman akan dicatat melalui antarmuka komunikasi serial dan direkam secara otomatis pada perangkat komputer ke dalam format *Excel*. Di sisi lain, perangkat *gateway* selaku penerima data akan menjalankan prosedur yang ekuivalen, yakni mentransmisikan informasi notifikasi melalui komunikasi serial untuk mengonfirmasi kepada komputer bahwa paket data telah berhasil diterima. Guna memastikan integritas dan akurasi sinkronisasi waktu, proses pencatatan stempel waktu pengiriman maupun penerimaan data dieksekusi secara terpusat pada satu unit komputer yang sama. Rangkaian pengujian ini



Hak Cipta :

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

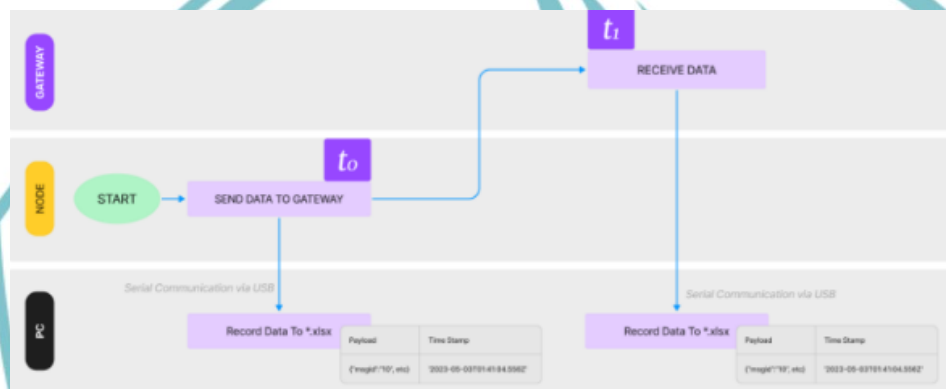
diulang dengan repetisi sebanyak 500 kali. Lebih lanjut diaplikasikan untuk mengalkulasi nilai *delay*, di mana kalkulasi tersebut diperoleh melalui selisih pengurangan antara waktu saat data selesai diterima dengan waktu saat data pertama kali dikirimkan.

$$\text{delay} = t_1 - t_0$$

t_0 = Waktu data dikirim

t_1 = Waktu data diterima

delay = Waktu transmisi satu arah



Gambar 4.36 Proses pengukuran latency one trip transmission

c) Full Trip Data Transmission

Penelitian terdahulu yang dilakukan oleh Yoppy et al., 2021 telah melaksanakan pengukuran waktu transmisi antarperangkat *node* guna mengevaluasi kinerja dari *ESP mesh network*. Pengujian pada studi tersebut menerapkan variasi topologi jaringan untuk mengidentifikasi perbedaan performa yang dihasilkan oleh masing-masing konfigurasi. Adapun pada penelitian ini, evaluasi hanya difokuskan pada analisis performa protokol *ESP-NOW*. Pengukuran kinerja dari kedua sistem tersebut didasarkan pada pengujian waktu jeda transmisi atau *delay*. Parameter yang dijadikan sebagai variabel utama pengujian meliputi jarak fisik antarperangkat *node* serta kuantitas *child node* yang terhubung di dalam ekosistem jaringan.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, 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 4.37 Alur pengukuran *full trip data latency*

Pengujian transmisi dua arah atau *full trip data* dilaksanakan guna mengukur total durasi transmisi yang dibutuhkan untuk mentransmisikan paket data dari perangkat *node* menuju *gateway*, hingga data tersebut dikembalikan dan diterima secara utuh oleh perangkat *node*. Skenario pengujian ini secara spesifik mengesampingkan tahapan autentikasi sistem. Paket data yang diterima oleh *gateway* akan secara langsung ditransmisikan ulang menuju *node* asal tanpa melalui tahapan pemrosesan komputasi apa pun. Pengujian ini akan dieksekusi dengan repetisi sebanyak 500 kali. Rangkaian pengujian tersebut bertujuan untuk mengakuisisi parameter waktu tunggu atau *latency time* yang direpresentasikan dalam satuan milisekon. Adapun rincian operasional dari prosedur pengukuran latensi ini divisualisasikan secara komprehensif pada Gambar 4.37

$$\text{latency} = t_3 - t_0$$

t_0 = waktu data dikirim

t_3 = waktu data diterima

latency = waktu transmisi bolak-balik

Nilai waktu transmisi dua arah atau *latency* diperoleh melalui perhitungan selisih waktu secara langsung. Kalkulasi ini dilakukan dengan cara mengurangkan waktu saat data berhasil diterima kembali oleh perangkat *node* pengirim dengan waktu awal saat data tersebut pertama kali ditransmisikan.

d) *Synchronization Delay*

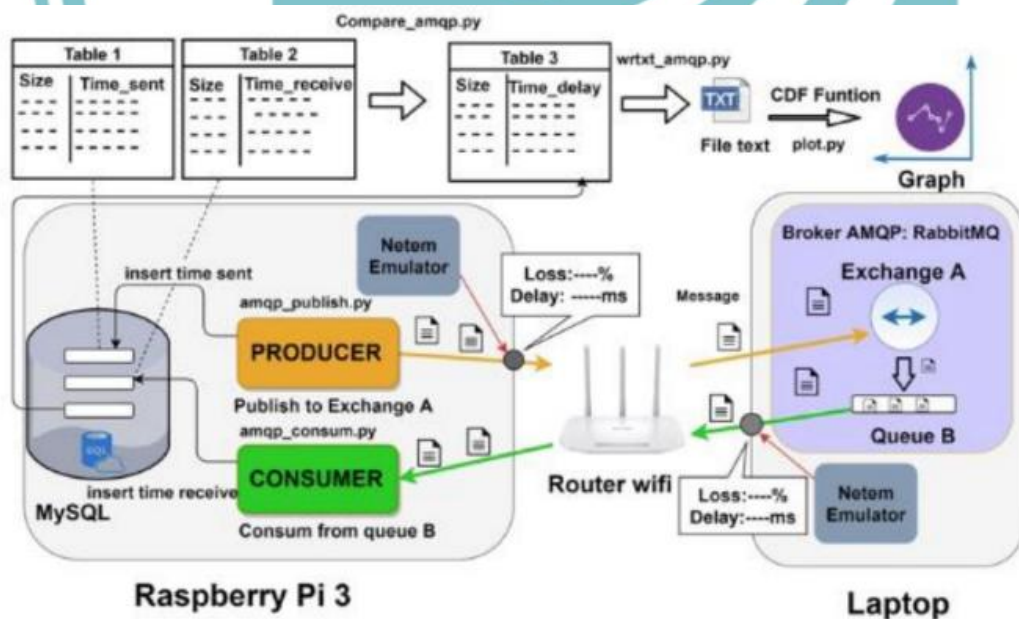


© Hak Cipta milik Politeknik Negeri Jakarta

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 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

Pengujian *synchronization delay* dilaksanakan guna mengukur total durasi yang dibutuhkan oleh layanan *cloud* dalam mentransmisikan paket data menuju *gateway*. Setiap data yang ditransmisikan oleh layanan *cloud* tersebut akan disematkan label penanda waktu. Selanjutnya, pada saat data berhasil diterima, perangkat *gateway* akan mengeksekusi kalkulasi selisih waktu guna menentukan durasi transmisi yang telah berlangsung. Berkaitan dengan hal tersebut, penelitian terdahulu yang dilaksanakan oleh Uy & Nam, 2021 telah melakukan komparasi terhadap durasi *delay* antara protokol MQTT dan AMQP. Sebagaimana diilustrasikan pada Gambar 4.38, pengujian pada studi tersebut menerapkan variasi interval waktu pengiriman data untuk mengevaluasi perbedaan performa yang dihasilkan. Adapun pada penelitian ini, parameter utama yang menjadi fokus evaluasi adalah kuantitas *producer* serta *consumer* yang terhubung menuju *server*. Skenario pengujian tersebut direalisasikan melalui metode simulasi penciptaan entitas *producer* dan *consumer* buatan dengan mengoptimalkan penggunaan fitur *thread* pada bahasa pemrograman Python.



Gambar 4.38 Skema pengujian delay

Sumber: Uy & Nam, 2021

e) *Synchronization Data Integrity*

Pengujian integritas sinkronisasi data atau *synchronization data integrity* dilaksanakan guna memastikan bahwa data yang bernaung pada layanan *cloud*



Hak Cipta :

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

memiliki presisi yang ekuivalen dengan data yang direplikasi oleh perangkat *gateway*. Prosedur pengujian ini dieksekusi melalui simulasi modifikasi hak akses wajah pada ruangan dengan mengimplementasikan skrip pemrograman Python. Skenario simulasi tersebut secara spesifik mencakup siklus penambahan identitas kartu baru, pencabutan hak akses kartu eksisting, serta modifikasi informasi kelengkapan pada data kartu. Selama proses simulasi berlangsung, *server* secara otomatis merekam log seluruh riwayat transmisi data yang diarahkan menuju *gateway*, sementara pada sisi penerima, perangkat *gateway* akan mencatat seluruh paket data yang berhasil diakuisisi. Sebagai tahapan evaluasi akhir, proses komparasi antara data yang ditransmisikan oleh *server* dan data yang direkam oleh *gateway* akan dianalisis secara komprehensif pascapengujian dengan memanfaatkan teknik analisis data berbasis bahasa pemrograman Python.

4.4.2 Prosedur Pengujian

Tabel 4.3 menunjukkan berbagai macam skenario pengujian yang akan dilakukan. Skenario ini bertujuan untuk menguji sistem pada kondisi terbaik hingga terburuk.

Tabel 4.3 Tabel prosedur pengujian

Tipe Pengujian	Skenario Pengujian
Pengujian Fungsionalitas	Pengujian fungsionalitas modul inisialisasi, registrasi wajah, pemicuan akses, transmisi wajah, pengenalan wajah, <i>Multi-Factor Authentication</i> (RFID dan keypad), pengendalian solenoid, <i>edge autonomy</i> , dan sinkronisasi log.
<i>One Trip Data Transmission</i>	Satu <i>Child Node</i> Dalam Jaringan, Terpisah Jarak Sejauh 1 meter
	Satu <i>Child Node</i> Dalam Jaringan, Terpisah Jarak Sejauh 3 meter
	Satu <i>Child Node</i> Dalam Jaringan, Terpisah Jarak Sejauh 5 meter
<i>Full Trip Data Transmission</i>	Satu <i>Child Node</i> Dalam Jaringan, Terpisah Jarak Sejauh 1 meter



Hak Cipta :

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

	Satu <i>Child Node</i> Dalam Jaringan, Terpisah Jarak Sejauh 3 meter
	Satu <i>Child Node</i> Dalam Jaringan, Terpisah Jarak Sejauh 5 meter
<i>Synchronization Delay</i>	Broadcast data dengan 2 <i>Client (Gateway)</i> Terkoneksi Ke <i>Server</i>
	Broadcast data dengan 50 <i>Client (Gateway)</i> Terkoneksi Ke <i>Server</i>
	Broadcast data dengan 100 <i>Client (Gateway)</i> Terkoneksi Ke <i>Server</i>
<i>Synchronization Data Integrity</i>	<i>Gateway</i> Terus Aktif Selama 3 Jam dengan Proses Modifikasi Data Terjadi Setiap 1 Menit
	<i>Gateway</i> Aktif Selama 1 Jam, nonaktif selama 1 jam, dan aktif kembali 1 jam. Proses Modifikasi Data Terjadi Setiap 1 Menit

4.4.3 Data Hasil Pengujian

Bagian ini memaparkan seluruh data yang diperoleh dari rangkaian kegiatan pengujian yang telah dilaksanakan, baik pengujian fungsionalitas sistem maupun pengujian kinerja transmisi data. Data hasil pengujian fungsionalitas disajikan untuk menunjukkan tingkat keberhasilan logika alur sistem dalam berbagai kondisi operasional. Sementara itu, data pengujian kinerja disajikan dalam bentuk metrik statistik deskriptif dan grafik analisis untuk mengevaluasi parameter QoS pada skenario pengiriman data nirkabel *one trip*, *full trip*, *synchronization delay*, serta *synchronization data integrity*. Pembahasan pada setiap sub-bab ditujukan untuk menilai kestabilan, kecepatan respon, dan efisiensi arsitektur *hybrid edge-cloud* yang diusulkan.

4.4.3.1 Pengujian Fungsionalitas

Pengujian fungsionalitas dilakukan untuk memastikan setiap komponen *hardware* dan modul *software* pada sistem dapat beroperasi sesuai dengan perancangan. Proses pengujian ini mencakup mekanisme inisialisasi jaringan, pendaftaran hak



Hak Cipta :

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

akses baru, proses *multi-factor authentication*, hingga *gateway autonomy* saat mengalami pemutusan koneksi internet. Tabel 4.4 menunjukkan hasil yang lebih detail untuk pengujian fungsionalitas komponen dan fitur pada sistem.

Tabel 4.4 Hasil Pengujian Fungsionalitas

No	Fitur / Komponen yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Status
1	Inisialisasi Sistem dan Modul Jaringan	<i>Gateway</i> (Raspberry Pi 4) dan <i>child node</i> (ESP32) dinyalakan dari kondisi awal.	Layar LCD 7 inci pada <i>gateway</i> aktif, modul ESP32 pada <i>node</i> berhasil terhubung ke <i>bridge</i> menggunakan <i>ESP Mesh</i> , dan modul ESP32-CAM siap menerima pemicu.	Berhasil
2	Mode Pendaftaran (<i>Register Mode</i>)	Admin mendaftarkan data wajah pengguna baru via LCD 7 inci <i>gateway</i> .	Kamera <i>webcam gateway</i> aktif menangkap gambar wajah, memetakan geometri wajah, dan menyimpan data pengguna baru ke <i>database</i> lokal SQLite <i>gateway</i> .	Berhasil
3	Pemicuan Deteksi Wajah (<i>Access Trigger</i>)	Pengguna menekan tombol pemicu/PIN awal pada <i>node</i> untuk meminta akses masuk.	Mikrokontroler ESP32 pada <i>node</i> mendeteksi <i>input</i> fisik dan memicu ESP32-CAM secara nirkabel untuk menangkap gambar wajah pengguna.	Berhasil
4	Transmisi Citra Wajah (<i>Wireless Transmission</i>)	ESP32-CAM mengirimkan citra wajah hasil tangkapan menuju <i>gateway</i> .	Gambar wajah berhasil dikirim via protokol ESP-NOW menuju <i>bridge</i> , diteruskan via serial ke Raspberry Pi, dan tersimpan di memori sementara <i>gateway</i> .	Berhasil



© 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

5	Autentikasi Pengenalan Wajah (<i>Face Recognition</i>)	Pencocokan wajah dilakukan oleh <i>gateway</i> lokal menggunakan model MobileFaceNet.	Wajah dikenali berdasarkan kecocokan geometri dengan skor di atas <i>threshold</i> , LCD <i>gateway</i> menampilkan hasil verifikasi, dan status akses dinyatakan sah.	Berhasil
6	<i>Multi-Factor Authentication</i> (MFA) - Kartu RFID	Pengguna memindai kartu RFID terdaftar pada modul RFID <i>node</i> (dalam kondisi MFA aktif).	<i>Node</i> membaca UID kartu dengan akurat, data kartu dikirim via <i>ESP Mesh</i> ke <i>gateway</i> , dan diverifikasi sesuai dengan hak akses yang terdaftar.	Berhasil
7	<i>Multi-Factor Authentication</i> (MFA) - PIN Keypad	Pengguna mengetikkan kombinasi PIN yang valid pada <i>keypad node</i> setelah verifikasi wajah.	<i>Keypad</i> mendeteksi ketukan dengan tepat, data PIN dikirim via <i>ESP Mesh</i> , dan berhasil lolos pencocokan kredensial di <i>gateway</i> .	Berhasil
8	Pengendalian <i>Solenoid Door Lock</i>	<i>Gateway</i> mengirimkan sinyal kendali "akses disetujui" ke <i>node</i> jika verifikasi wajah dan MFA sah.	<i>Node</i> menerima sinyal, memicu relay 12V untuk menarik <i>plunger solenoid door lock</i> sehingga pintu terbuka selama 5 detik, dan LCD OLED menampilkan status akses berhasil.	Berhasil
9	Penolakan Akses Pengguna Asing (<i>Access Denied</i>)	Orang dengan wajah, kartu RFID, atau PIN yang tidak terdaftar mencoba mengakses pintu.	Sistem menolak hak akses, solenoid pintu tetap terkunci rapat, LCD OLED <i>node</i> memunculkan pesan "Akses Ditolak", dan upaya akses ilegal dicatat ke <i>log</i> .	Berhasil



© 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

10	Kondisi ketika sistem sedang <i>offline</i>	Pengguna melakukan verifikasi wajah dan MFA saat koneksi internet <i>gateway</i> ke <i>cloud</i> terputus.	Sistem tetap berfungsi normal secara <i>offline</i> , keputusan pembukaan pintu diproses secara lokal oleh <i>gateway</i> , dan <i>log</i> aktivitas disimpan sementara pada memori lokal.	Berhasil
11	Sinkronisasi <i>Log Akses</i> (AMQP RabbitMQ)	Jaringan internet pada <i>gateway</i> terhubung kembali setelah berjalan dalam mode <i>offline</i> .	<i>Gateway</i> mendeteksi koneksi aktif dan secara otomatis mengirimkan seluruh <i>log</i> akses lokal ke server <i>cloud</i> menggunakan protokol AMQP melalui <i>broker</i> RabbitMQ.	Berhasil
12	Replikasi Hak Akses Baru dari <i>Cloud</i>	Administrator memodifikasi konfigurasi hak akses atau data wajah pengguna melalui <i>dashboard web cloud</i> .	<i>Gateway</i> menerima pembaruan secara <i>real-time</i> via AMQP, mereplikasi perubahan ke <i>database</i> SQLite lokal secara utuh tanpa ada paket data yang hilang.	Berhasil

Berdasarkan data hasil pengujian yang tertera pada Tabel 4.4, seluruh fitur utama dan komponen pendukung sistem berhasil berjalan dengan status berhasil. Implementasi arsitektur *edge autonomy* terbukti mampu menjaga fungsionalitas esensial sistem keamanan pintu tetap aktif secara lokal meskipun interaksi ke layanan *cloud* terputus. Selain itu, seluruh *log* aktivitas dan pembaruan data hak akses dapat disinkronisasikan kembali secara utuh tanpa ada paket data yang hilang setelah koneksi *gateway* menuju *broker* RabbitMQ terhubung kembali.

4.4.3.2 One Trip Data Transmission

Pengujian *one-trip data transmission* dilakukan untuk mengukur performa pengiriman data citra satu arah dari perangkat *node* menuju *gateway* menggunakan protokol nirkabel ESP-NOW. Parameter utama yang diukur dalam pengujian ini



Hak Cipta :

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

adalah *delay* transmisi dan *packet loss*. Pengujian dilakukan secara berulang sebanyak 500 kali untuk masing-masing variasi jarak, yaitu 1 meter, 3 meter, dan 5 meter, guna mendapatkan data statistik yang representatif mengenai keandalan sistem dalam berbagai kondisi jarak. Rangkuman data hasil pengujian transmisi satu arah pada ketiga variasi jarak tersebut disajikan pada Tabel 4.5.

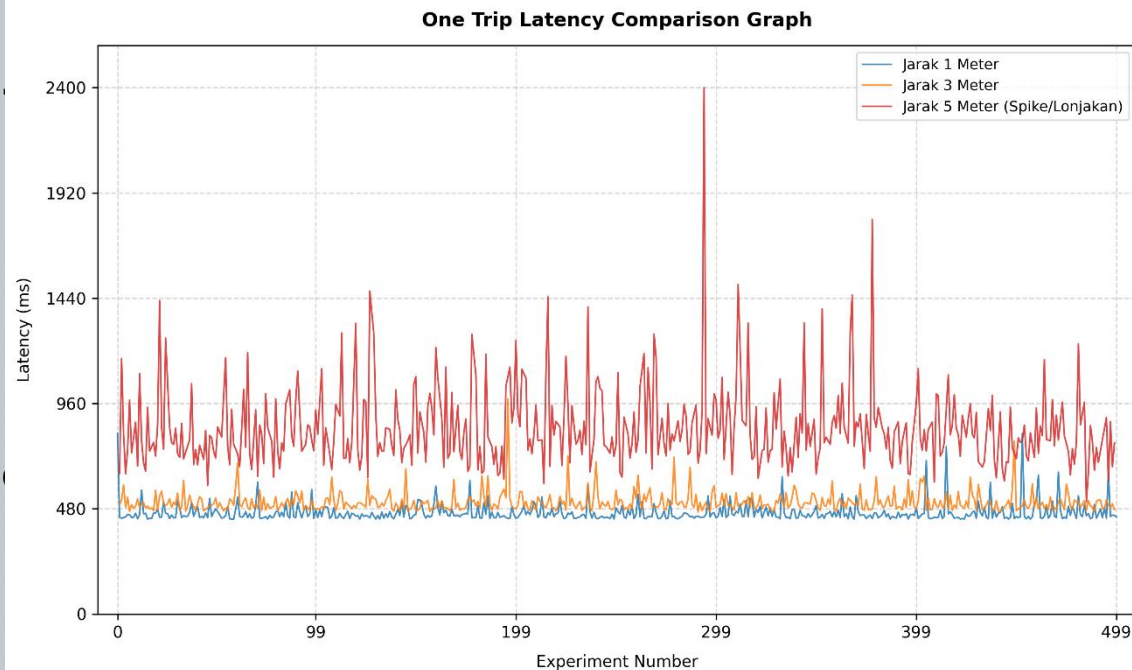
Tabel 4.5 Hasil pengujian *one trip*

Jarak	Total data	Total data valid	Packet loss (%)	Delay (ms)				Jitter (ms)
				Mean (ms)	Median (ms)	Max (ms)	Min (ms)	
1 Meter	500	500	0	462,38	449,19	823,17	433,1	19
3 Meter	500	486	2,8	510,45	495,2	980,5	465,3	24,5
5 Meter	500	440	12	850,12	810,8	2400,4	510,2	95,2

Berdasarkan data pada Tabel 4.5, dapat dianalisis bahwa performa transmisi satu arah sangat dipengaruhi oleh jarak fisik antar perangkat. Pada pengujian jarak 1 meter, sistem menunjukkan tingkat keandalan yang sangat tinggi dengan *packet loss* sebesar 0 % serta nilai *mean* sebesar 462,38 ms dan *jitter* yang rendah sebesar 19 ms, menandakan transmisi yang sangat stabil. Pada pengujian jarak 3 meter, mulai terdeteksi kegagalan transmisi berupa *packet loss* sebesar 2,8% dengan rata-rata delay meningkat menjadi 510,45 ms dan jitter 24,50 ms.

Ketika pengujian dilakukan pada jarak 5 meter, terjadi lonjakan penurunan performa yang cukup signifikan. Rata-rata waktu transmisi meningkat tajam menjadi 850,12 ms dengan delay maksimal mencapai 2400,4 ms. Tingkat *packet loss* melonjak hingga 12 % disertai perubahan waktu pengiriman (*jitter*) yang tinggi mencapai 95,20 ms. Pelemahan daya pancar sinyal (*path loss*) dan gangguan sinyal di sekitar area pengujian pada jarak 5 meter menyebabkan kegagalan pengiriman beberapa bagian paket citra wajah, sehingga memicu terjadinya *timeout*. Perbandingan karakteristik naik turunnya waktu transmisi untuk masing-masing variasi jarak ini secara lebih detail dapat dilihat pada grafik visualisasi Gambar 4.39.

- Hak Cipta :**
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, 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 4.39 Hasil pengujian *one trip*

Grafik pada Gambar 4.39 memvisualisasikan tren *delay* ini secara jelas. Grafik garis berwarna biru (1 meter) berada secara konstan di bagian bawah, grafik berwarna oranye (3 meter) sedikit lebih tinggi dengan naik turun yang kecil, sedangkan grafik berwarna merah (5 meter) berubah naik turun secara drastis dengan pola yang tajam disertai *gap* yang merepresentasikan kegagalan pengiriman data. Hal ini membuktikan pentingnya pembatasan jarak jangkauan ESP-NOW di bawah 5 meter guna menjamin keandalan pengiriman citra wajah pada sistem ini.

4.4.3.3 Full Trip Data Transmission

Pengujian *full-trip data transmission* dilakukan untuk mengukur total waktu pengiriman data bolak-balik dari *node* ke *gateway* melalui *ESP32 Bridge*, lalu kembali lagi ke *node* untuk mengaktifkan aktuator *solenoid door lock*. Pengujian ini menggunakan *payload* citra wajah berukuran 21 KB yang dibagi ke dalam 94 *chunk* transmisi dengan jeda 2 ms antar *chunk*. Pengujian dilakukan sebanyak 500 kali untuk setiap variasi jarak. Rangkuman data hasil pengujian transmisi bolak-balik pada ketiga variasi jarak tersebut disajikan pada Tabel 4.6.



Tabel 4.6 Hasil pengujian *full trip*

Jarak	Total data	Total data valid	Packet loss (%)	Delay (ms)				Jitter (ms)
				Mean (ms)	Median (ms)	Max (ms)	Min (ms)	
1 Meter	500	500	0	104,59	105	150,2	90,1	9,65
3 Meter	500	497	0,6	352,4	340,5	580,1	310,2	28,30
5 Meter	500	492	1,6	663,2	654	1028	592	45,52

Berdasarkan data pada Tabel 4.6, dapat dianalisis bahwa performa transmisi *full-trip* dipengaruhi oleh jarak antar perangkat. Pada pengujian jarak 1 meter, sistem menunjukkan tingkat keandalan yang sangat tinggi dengan *packet loss* sebesar 0,00% serta nilai *mean* sebesar 104,59 ms and *jitter* yang rendah sebesar 9,65 ms, menandakan transmisi yang sangat stabil. Pada pengujian jarak 3 meter, rata-rata delay RTT meningkat menjadi 352,4 ms dengan median sebesar 340,5 ms and jitter 28,3 ms, serta tingkat *packet loss* yang sangat rendah yaitu sebesar 0,6%.

Ketika pengujian dilakukan pada jarak 5 meter, rata-rata waktu RTT meningkat menjadi 663,2 ms dengan delay maksimal mencapai 1028,00 ms. Tingkat *packet loss* tercatat sebesar 1,60% disertai jitter sebesar 45,52 ms. Meskipun terjadi peningkatan *delay* dan *packet loss* akibat gangguan sinyal di udara atau pelemahan sinyal, sistem tetap dinilai sangat andal dan responsif untuk melakukan autentikasi di dalam gedung. Perbandingan karakteristik naik turunnya waktu RTT untuk masing-masing variasi jarak ini secara lebih detail dapat dilihat pada grafik visualisasi Gambar 4.40.

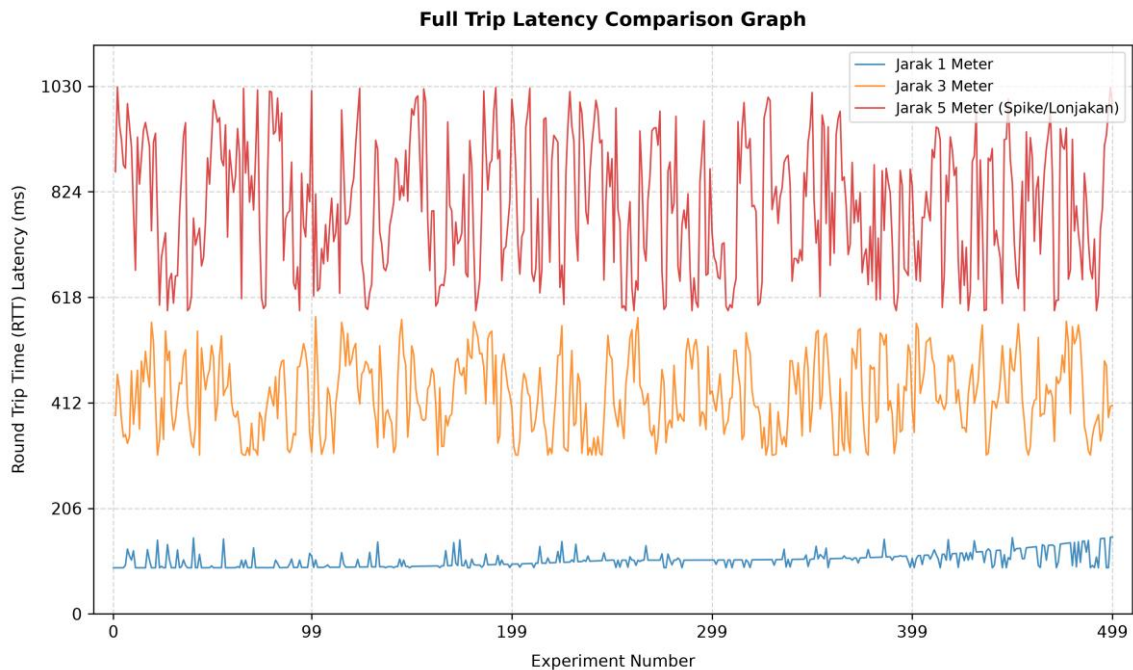
Hak Cipta :

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

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, 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 4.40 Hasil pengujian *full trip*

Grafik pada Gambar 4.40 memvisualisasikan tren *Round Trip Time* secara temporal untuk masing-masing variasi jarak selama 500 kali pengujian. Grafik garis berwarna biru yang merepresentasikan pengujian jarak 1 meter berada secara konstan di bagian paling bawah dengan pola garis lurus yang sangat rapat dan minim gejolak. Pola ini menunjukkan kondisi propagasi gelombang elektromagnetik yang sangat ideal tanpa adanya hambatan berarti, sehingga transmisi data bolak-balik dapat berlangsung dengan tingkat stabilitas yang sangat tinggi. Ketika jarak pengujian ditingkatkan menjadi 3 meter, posisi kurva garis berwarna oranye bergeser naik ke arah tengah grafik disertai munculnya fluktuasi kecil di sepanjang garis. Fluktuasi ini mengindikasikan mulai adanya pelemahan daya pancar nirkabel serta *delay* akibat proses pengiriman ulang paket di lapisan *medium access control* ketika terjadi kesalahan deteksi.

Pada jarak terjauh yaitu 5 meter, kurva garis berwarna merah berada di posisi paling atas dan memperlihatkan pola naik-turun yang sangat drastis dengan banyak lonjakan tajam. Lonjakan-lonjakan tajam ini menunjukkan bahwa perangkat harus melakukan upaya pengiriman ulang paket secara berulang-ulang untuk mengatasi interferensi dan pelemahan sinyal di udara sebelum data berhasil diterima secara



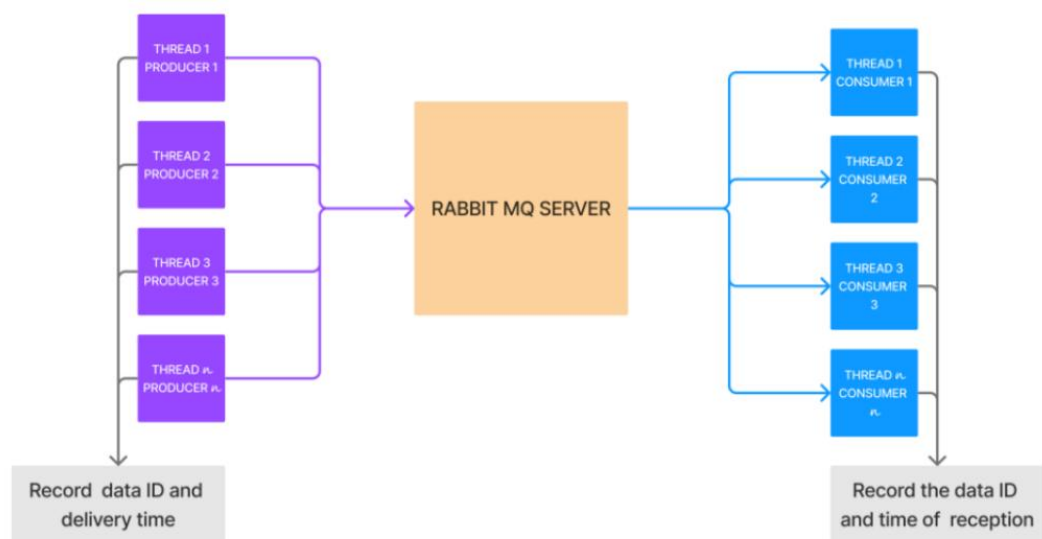
Hak Cipta :

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

utuh. Selain itu, pada grafik berwarna merah ini juga terdeteksi adanya beberapa *gap* yang merepresentasikan kegagalan pengiriman paket secara permanen. Meskipun karakteristik grafik pada jarak 5 meter menunjukkan ketidakstabilan sinyal yang cukup tinggi, performa waktu respon sistem secara keseluruhan masih berada dalam batas toleransi yang sangat baik untuk pembukaan kunci pintu secara cepat. Guna menjaga keandalan pengiriman data agar tetap optimal dengan kegagalan akses seminimal mungkin, disarankan agar jarak jangkauan komunikasi nirkabel antar perangkat dipertahankan pada kisaran 1 hingga 3 meter.

4.4.3.4 Synchronization Delay

Pengujian *delay* dilakukan dengan cara memberikan beban kerja yang tinggi pada server RabbitMQ. Kondisi beban tinggi ini disimulasikan melalui pembuatan koneksi tiruan yang saling bertukar data, di mana proses pembuatan koneksi tersebut memanfaatkan fitur *multithreading* pada bahasa pemrograman Python.



Gambar 4.41 Metode pengujian latency

Metode pengujian *latency* pada server lokal dilakukan melalui tiga skema variasi beban, seperti yang ditunjukkan pada Gambar 4.41 Ketiga skema tersebut melibatkan interaksi simultan dari 2, 50, dan 100 koneksi, di mana istilah koneksi ini merepresentasikan jumlah pasang *RabbitMQ producer* serta *RabbitMQ consumer* yang aktif terhubung ke server dalam waktu yang bersamaan.



© 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

Server dijalankan secara langsung pada sebuah laptop Lenovo LOQ 15 yang dilengkapi dengan kapasitas RAM sebesar 28 GB. Laptop ini bertindak sebagai *host* utama dengan spesifikasi prosesor Intel Core i5-13450HX (16 CPU) berkecepatan ~2,4 GHz. Sementara itu, untuk pengujian beban sistem, simulasi perangkat *producer* dan *consumer* diimplementasikan menggunakan perangkat eksternal berupa Raspberry Pi 4B yang terhubung ke server melalui jaringan *Wireless Local Area Network* (WLAN).

Pada setiap skenario pengujian, dilakukan pengiriman data secara berkala dengan interval jeda selama 0,2 detik di setiap pengirimannya. Ukuran payload data yang ditransmisikan dari server ke gateway ditetapkan konstan sebesar 4 KB. Melalui pengujian ini, akan diperoleh nilai delay yang berfungsi sebagai indikator utama dalam menilai kapabilitas server saat menangani lalu lintas pesan yang masuk. Nilai delay itu sendiri merupakan total waktu yang diperlukan oleh data untuk berpindah dari sisi pengirim hingga sampai ke penerima. Rangkuman data hasil pengujian delay sinkronisasi secara bersamaan pada berbagai variasi beban kerja disajikan pada Tabel 4.7.

Tabel 4.7 Hasil pengujian *Synchronization Delay*

Koneksi	Total data	Total data valid	Packet loss (%)	Delay (ms)				Jitter (ms)
				Mean (ms)	Median (ms)	Max (ms)	Min (ms)	
2	1000	1000	0	66,3	58,2	695,8	10,3	21
50	25000	25000	0	484	19,2	15225,4	6,4	151,1
100	50000	50000	0	788	46,7	14968,4	6	181,8

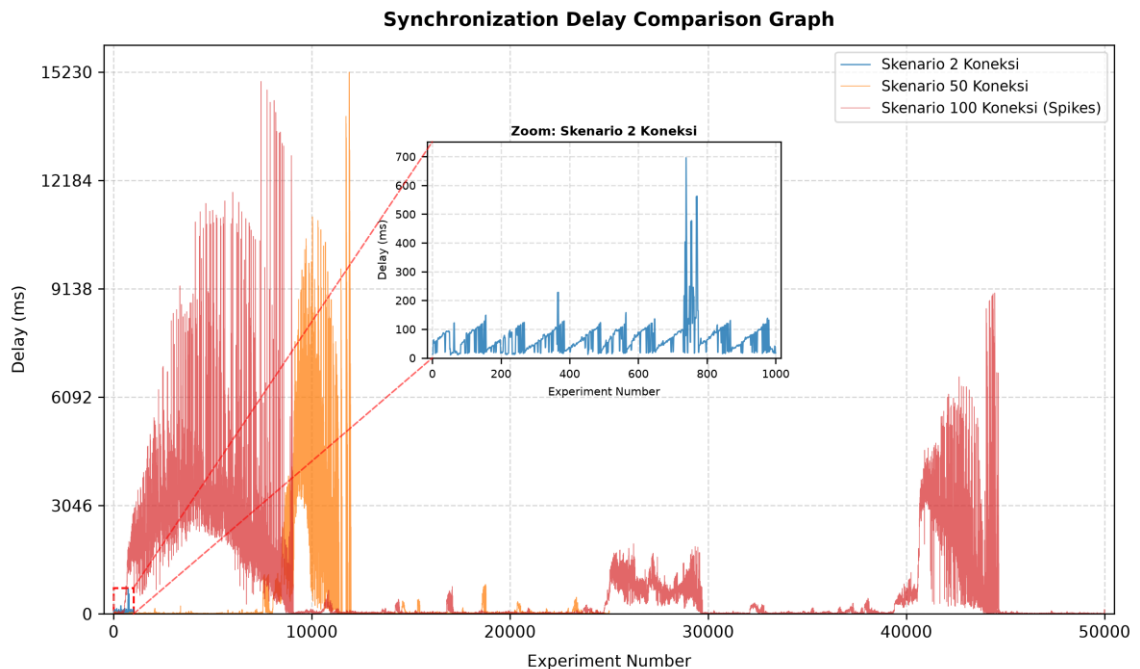
Berdasarkan data pada Tabel 4.7, dapat dianalisis bahwa performa sinkronisasi sangat dipengaruhi oleh tingkat beban kerja secara bersamaan yang diterima oleh server. Pengujian menunjukkan keandalan pengiriman pesan yang sempurna dengan tingkat keberhasilan mencapai 100% tanpa adanya data yang hilang di seluruh skenario pengujian. Pada beban kerja rendah yaitu 2 koneksi secara bersamaan, sistem dapat memproses paket data dengan sangat cepat dan stabil. Ketika beban kerja ditingkatkan menjadi 50 dan 100 koneksi secara simultan, *delay*



Hak Cipta :

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

rata-rata mengalami peningkatan serta memicu terjadinya variasi *delay* yang jauh lebih lebar akibat peningkatan beban kerja komputasi pada server RabbitMQ dan antrean pemrosesan pesan asinkron.



Gambar 4.42 Grafik delay sinkronisasi secara bersamaan

Grafik pada Gambar 4.42 memvisualisasikan perbandingan tren *delay* sinkronisasi data secara temporal untuk skenario 2, 50, dan 100 koneksi secara bersamaan pada satu sumbu koordinat. Kurva garis berwarna biru yang merepresentasikan skenario 2 koneksi secara bersamaan yang membentang sangat pendek hingga batas pengujian ke-1000 dan berada secara konstan di bagian paling bawah grafik tanpa adanya lonjakan berarti. Pola ini menunjukkan bahwa pada tingkat beban kerja rendah, jaringan dan server dapat melakukan pemrosesan data dengan kestabilan yang sangat tinggi dan tingkat variabilitas *delay* yang minimal.

Sementara itu, kurva garis berwarna oranye (50 koneksi) dan merah (100 koneksi) membentang jauh lebih panjang di sepanjang sumbu horizontal masing-masing hingga pengujian ke-25000 dan ke-50000. Kedua kurva tersebut menunjukkan pola baseline yang rapat di bagian dasar grafik tetapi memiliki kerapatan lonjakan-lonjakan tajam yang menjulang ke atas secara acak. Lonjakan tajam pada kurva merah terlihat lebih padat dan tinggi dibandingkan kurva oranye, yang



Hak Cipta :

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

mencerminkan terjadinya antrean penjadwalan komputasi di lapisan server RabbitMQ serta persaingan ketat dalam penulisan database lokal secara bersamaan di sisi *gateway* seiring bertambahnya beban lalu lintas data secara masif. Meskipun diwarnai oleh anomali lonjakan waktu tunda sesaat pada kondisi beban secara bersamaan tinggi, baseline delay dari kedua skenario tersebut sebagian besar tetap berada di kisaran waktu yang sangat cepat dan dapat ditoleransi dengan baik oleh sistem.

4.4.3.5 Synchronization Data Integrity

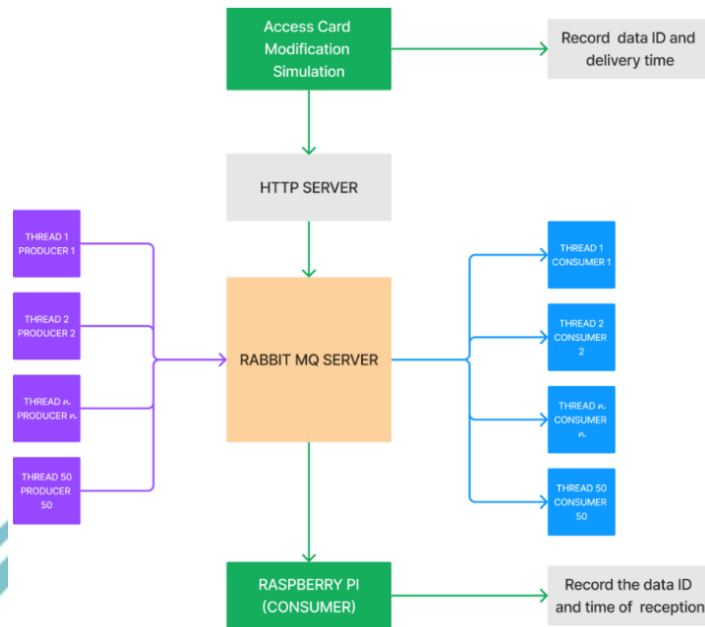
Synchronization data integrity merupakan pengujian yang dilakukan untuk menjamin bahwa seluruh data kiriman *producer* telah diterima dengan sukses oleh *consumer*. Hal ini penting karena menyangkut sistem manajemen akses, sebab kegagalan penerimaan data dapat menyebabkan kelonggaran keamanan sehingga orang yang tidak berhak bisa mengakses ruangan dengan bebas. Untuk mensimulasikan beban kerja server yang sibuk, pengujian menggunakan 50 *producer* dan *consumer* buatan yang saling bertukar data tanpa henti. Pengujian dilakukan dengan memanggil API yang mengatur perubahan akses ruangan seperti menambahkan data wajah ke suatu ruangan, mencabut hak akses wajah tertentu, dan mengubah detail data wajah yang ada. Proses modifikasi data tersebut dieksekusi secara otomatis dengan interval waktu setiap 1 menit sekali.

API yang diakses akan meneruskan data ke *gateway* menggunakan protokol AMQP, lengkap dengan pelabelan ID dan waktu kirim dari HTTP Server. *Gateway* kemudian mencatat waktu penerimaan data tersebut dalam log secara berkala pada setiap siklus perubahan data. Bagan alur pengujian ditunjukkan pada Gambar 4.43. Data yang terkumpul dari riwayat perubahan per menit tersebut akan dianalisis untuk mengukur nilai *packet loss* sebagai indikator tingkat keberhasilan transmisi data.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, 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 4.43 Proses pengujian *Synchronization* data integrity

Pengujian integritas data sinkronisasi dilakukan untuk menguji sejauh mana sistem dapat menjamin sinkronisasi data konfigurasi hak akses antara server cloud dan database lokal gateway. Skenario pertama dilakukan dengan kondisi koneksi internet yang terus aktif selama 3 jam penuh dengan beban sibuk (50 koneksi simultan). Skenario kedua mensimulasikan kegagalan jaringan nirkabel secara dinamis, di mana gateway diputus dari koneksi internet selama 1 jam di tengah pengujian, kemudian dihubungkan kembali pada jam terakhir untuk menguji kemampuan pemulihan data otomatis. Rangkuman data hasil pengujian kedua skenario tersebut disajikan pada Tabel 4.8.

Tabel 4.8 Hasil pengujian *Synchronization Data Integrity*

Skenario	Total data	Total data valid	Packet loss (%)	Delay (ms)				Jitter (ms)
				Mean (ms)	Median (ms)	Max (ms)	Min (ms)	
Terus Terkoneksi 3 Jam	180	180	0	203,7	129,6	3886,4	77	163,2

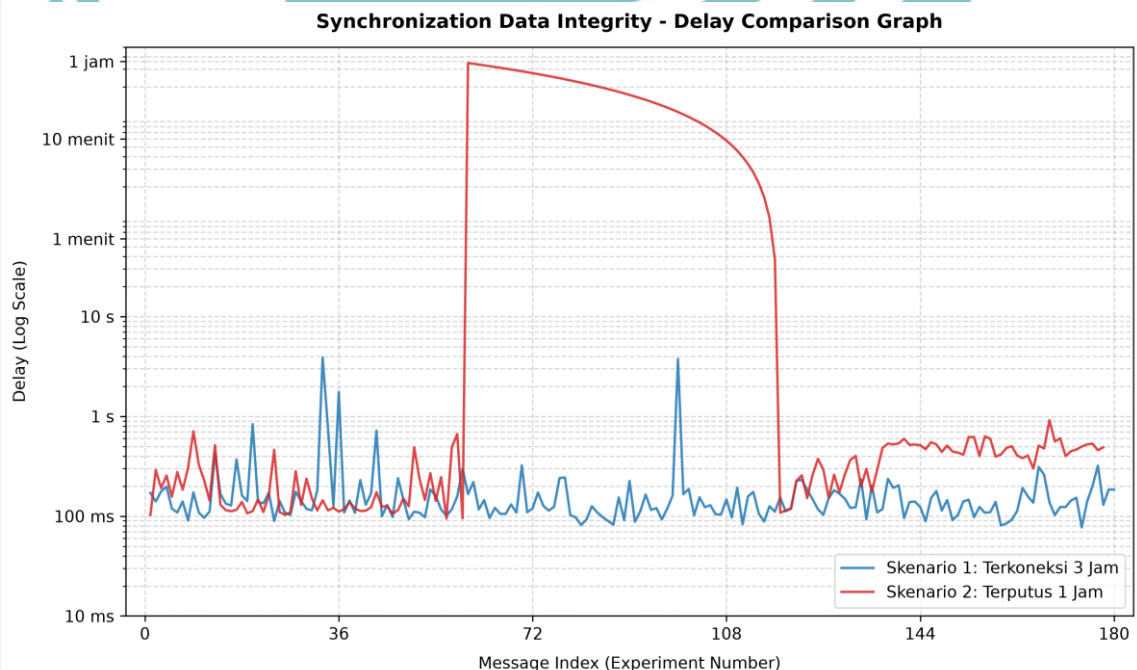


Hak Cipta :

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

Kondisi Jaringan Dinamis	178	178	0	570,6	471,5	3.464.220,1	6,4	151,1
--------------------------	-----	-----	---	-------	-------	-------------	-----	-------

Berdasarkan data pada Tabel 4.8, kedua skenario mencatat tingkat keberhasilan pengiriman paket yang sempurna sebesar 100% tanpa adanya data yang hilang .Hal ini membuktikan keandalan sistem dalam mempertahankan integritas data hak akses. Pada skenario pertama dengan koneksi terus-menerus aktif, proses replikasi data asinkron berjalan sangat responsif dan efisien. Sementara pada skenario kedua, meskipun terjadi *delay* rata-rata yang sangat besar akibat terputusnya koneksi nirkabel selama 1 jam, sistem tetap berhasil melakukan sinkronisasi otomatis terhadap seluruh data perubahan yang tertahan setelah koneksi pulih kembali, sehingga tidak ada satu pun konfigurasi hak akses yang terlewat atau rusak.



Gambar 4.44 Grafik perbandingan delay synchronization data integrity

Grafik pada Gambar 4.44 memvisualisasikan tren waktu tunda sinkronisasi secara temporal untuk skenario terkoneksi terus menerus dan skenario jaringan dinamis.



© 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

Sumbu vertikal dipetakan menggunakan skala logaritma agar visualisasi fluktuasi latensi rendah pada tingkat milidetik dan akumulasi latensi tinggi pada tingkat jam dapat disajikan secara bersamaan dengan sangat jelas. Sementara itu, sumbu horizontal merepresentasikan indeks pesan dari 0 hingga 180. Karena simulasi modifikasi data dikonfigurasi terjadi secara berkala setiap 1 menit sekali, maka indeks pesan pada sumbu horizontal ini secara langsung berkorelasi dengan waktu berjalan dalam satuan menit. Dalam korelasi ini, indeks 0 sampai 60 merepresentasikan jam pertama yaitu menit ke-0 hingga 60, indeks 60 sampai 120 merepresentasikan jam kedua yaitu menit ke-60 hingga 120 ketika koneksi WLAN dimatikan, dan indeks 120 sampai 180 merepresentasikan jam ketiga yaitu menit ke-120 hingga 180 saat koneksi diaktifkan kembali. Kurva garis berwarna biru yang merepresentasikan skenario terkoneksi 3 jam terus-menerus terlihat melayang secara stabil di bagian bawah grafik pada kisaran 70 ms hingga 4.000 ms. Pola ini mengonfirmasi bahwa dalam kondisi jaringan normal tanpa gangguan, replikasi data asinkron dapat segera diproses dengan latensi yang sangat rendah dan variasi delay yang minimal.

Sebaliknya, kurva garis berwarna merah yang menggambarkan skenario kondisi jaringan dinamis berawal dari tingkat baseline yang sama, kemudian mengalami kenaikan sangat tajam membentuk kurva menyerupai kubah besar yang mencapai puncak tertinggi di bawah garis indikator 1 jam, sebelum akhirnya menurun kembali secara teratur menuju tingkat baseline stabil. Visualisasi ini merepresentasikan kondisi ketika koneksi terputus selama 1 jam sehingga seluruh pesan modifikasi ditumpuk di dalam antrean server RabbitMQ dan menyebabkan akumulasi waktu tunda terus berjalan. Begitu koneksi pulih kembali, pesan-pesan tersebut langsung diproses secara berurutan, menyebabkan pesan pertama yang disinkronkan memiliki delay akumulasi tertinggi sebesar sekitar 57 menit dan secara bertahap menurun secara linier hingga antrean kosong dan sistem kembali beroperasi pada kondisi real-time. Di samping keandalan pengiriman pesan yang sempurna, kinerja komputasi gateway selama pemrosesan antrean pasca-pemulihan tersebut juga terpantau sangat efisien dengan penggunaan beban CPU rata-rata sebesar 16,28% serta penggunaan memori RAM sebesar 15,06%.



Hak Cipta :

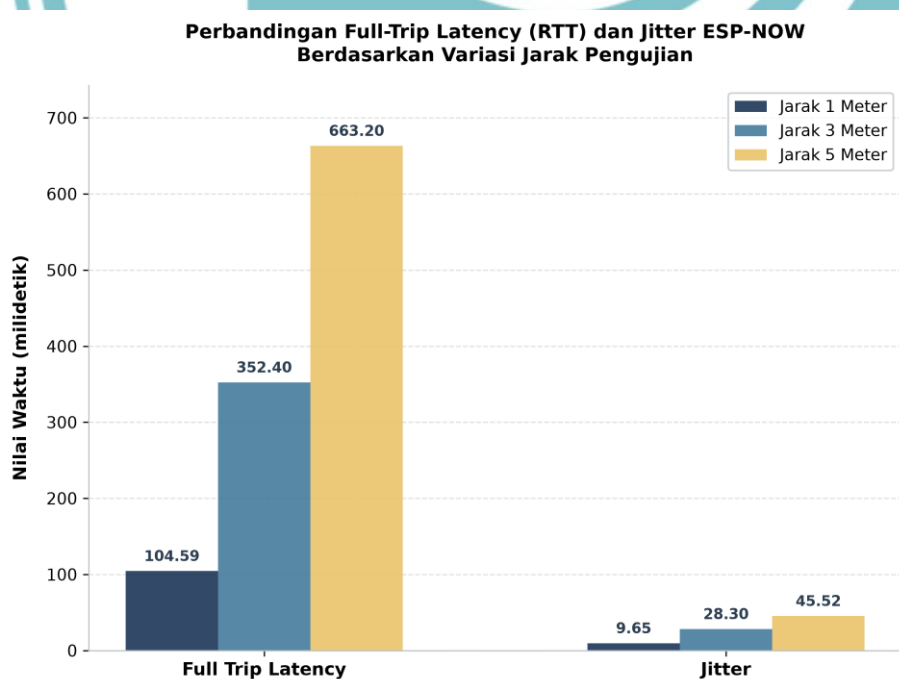
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan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

4.5 Analisis Data

Setelah melakukan seluruh tahapan pengujian dan mengumpulkan data, pada sub-bab ini dipaparkan hasil analisis mendalam terhadap karakteristik kinerja sistem. Analisis data dibagi menjadi dua bagian utama, yaitu analisis performa transmisi nirkabel ESP-NOW dan analisis performa sinkronisasi database lokal pada gateway. Seluruh analisis difokuskan secara mendalam pada performa arsitektur hybrid edge-cloud control yang dirancang dalam penelitian ini untuk menunjukkan efisiensi dan keandalan sistem dalam berbagai kondisi operasional.

1) ESP-NOW

Karakteristik transmisi nirkabel ESP-NOW diuji secara bolak-balik (full-trip) pada tiga variasi jarak yaitu 1 meter, 3 meter, dan 5 meter untuk melihat pengaruh jarak fisik terhadap waktu tunda dan stabilitas pengiriman paket data citra wajah. Hasil pengujian menunjukkan adanya korelasi positif yang sangat kuat antara jarak fisik perangkat dengan latensi RTT serta jitter. Perbandingan visual dari metrik waktu tunda tersebut disajikan pada Gambar 4.45.



Gambar 4.45 Grafik analisis *latency* dan *jitter* pengujian full-trip ESP-NOW

Berdasarkan tren grafik pada Gambar 4.45, kurva latency dan jitter memperlihatkan peningkatan secara bertahap seiring bertambahnya jarak pengujian. Peningkatan ini disebabkan oleh pelemahan daya pancar gelombang radio di ruang bebas serta

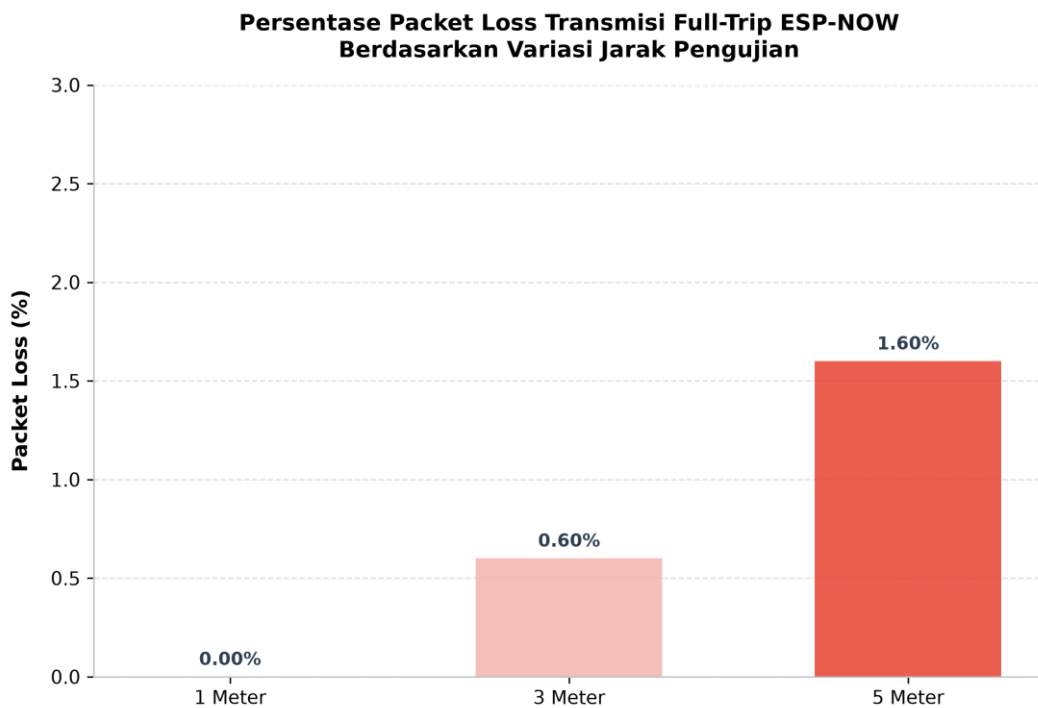


© Hak Cipta milik Politeknik Negeri Jakarta

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

meningkatnya hambatan fisik yang memicu kegagalan transmisi pada lapisan fisik. Kondisi tersebut memaksa protokol ESP-NOW untuk melakukan percobaan pengiriman ulang paket secara berulang-ulang, yang secara langsung menambah akumulasi waktu tunda total sebelum data akhirnya dapat diterima dengan lengkap. Di samping peningkatan delay, keandalan pengiriman paket juga dipengaruhi oleh jarak fisik, seperti yang divisualisasikan pada Gambar 4.46.



Gambar 4.46 Grafik analisis packet loss pengujian full-trip ESP-NOW

Visualisasi persentase packet loss pada Gambar 4.46 menegaskan bahwa kestabilan koneksi nirkabel berada pada tingkat terbaik pada jarak dekat. Pada jarak 1 meter dan 3 meter, sistem menunjukkan keandalan mutlak dengan tingkat kehilangan paket yang nihil. Namun, saat jarak diperluas hingga 5 meter, redaman sinyal yang lebih kuat mulai memicu terjadinya kegagalan transmisi permanen yang menghasilkan persentase packet loss yang kecil. Hal ini membuktikan bahwa batas jangkauan optimal untuk transmisi citra wajah lokal menggunakan ESP-NOW adalah berada pada rentang 1 hingga 3 meter guna menjamin kesuksesan pengiriman data tanpa kegagalan.

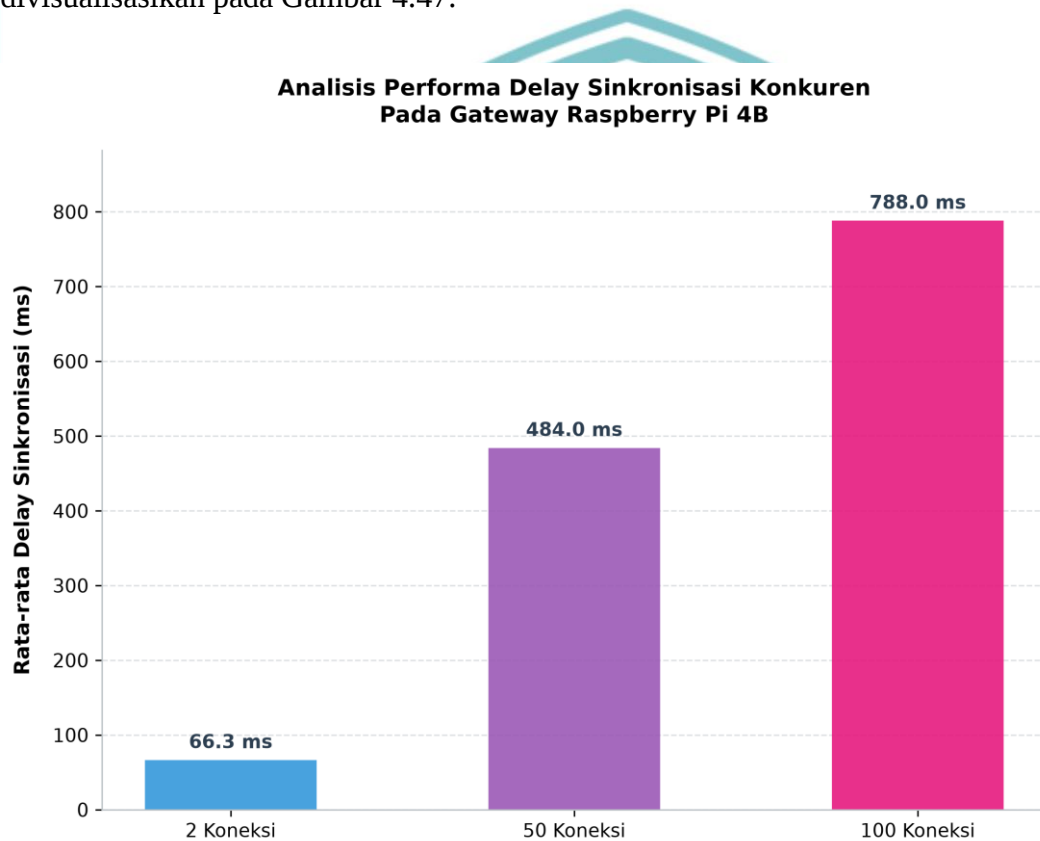
2) Sinkronisasi



Hak Cipta :

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

Analisis performa sinkronisasi secara bersamaan ditujukan untuk mengukur batas kapabilitas gateway dalam memproses antrean replikasi data dari server cloud secara simultan pada tiga tingkat beban kerja yang berbeda. Rangkuman waktu respon rata-rata untuk masing-masing skenario beban kerja tersebut divisualisasikan pada Gambar 4.47.



Gambar 4.47 Grafik analisis performa delay sinkronisasi secara bersamaan

Grafik pada Gambar 4.47 menunjukkan peningkatan rata-rata waktu respon sinkronisasi yang sebanding dengan penambahan jumlah koneksi secara bersamaan yang aktif secara bersamaan. Pada beban kerja rendah yaitu 2 koneksi, pemrosesan antrean berjalan dengan sangat cepat karena beban komputasi server dan gateway masih sangat minim. Ketika beban kerja ditingkatkan menjadi 50 dan 100 koneksi secara bersamaan secara simultan, waktu respon rata-rata bergeser naik tetapi masih berada dalam tingkat performa yang sangat efisien dan responsif di bawah ambang batas satu detik.

Keberhasilan sistem dalam mempertahankan performa di bawah satu detik pada beban secara bersamaan tinggi didukung oleh spesifikasi perangkat keras gateway



© Hak Cipta milik Politeknik Negeri Jakarta

yang menggunakan Raspberry Pi 4B dengan prosesor quad-core dan kapasitas RAM yang besar. Selain itu, arsitektur perangkat lunak yang menggunakan broker pesan RabbitMQ dengan protokol AMQP terbukti sangat efektif dalam mengelola antrean pesan secara asinkron di memori sebelum ditulis secara bertahap ke database lokal SQLite. Mekanisme ini mencegah terjadinya hambatan pemrosesan CPU dan masalah penguncian database yang biasanya terjadi pada sistem penulisan database secara bersamaan. Keandalan sinkronisasi ini juga didukung oleh integritas data yang sangat kuat berkat mekanisme *message acknowledgment* (ACK) pada RabbitMQ, yang menjamin bahwa jika gateway dalam kondisi offline, pesan sinkronisasi akan tetap tersimpan aman dalam antrean server cloud dan akan direplikasi secara otomatis segera setelah gateway kembali online. rusak seluruh data aktivitas log akan hilang secara permanen. Sebaliknya, sistem ini mengintegrasikan broker RabbitMQ (AMQP) dengan mekanisme *message acknowledgment* (ACK), sehingga saat *gateway* dalam kondisi *offline*, pesan pembaruan dari *cloud* tetap tersimpan aman di antrean dan disinkronkan kembali otomatis setelah koneksi kembali aktif *packet loss* 0,00%

POLITEKNIK
NEGERI
JAKARTA

Hak Cipta :

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

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, 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 seluruh tahapan perancangan, implementasi, dan pengujian yang telah dilaksanakan pada sistem *smart door lock* berbasis *Internet of Things (IoT)* dengan pendekatan *hybrid edge-cloud control*, dapat ditarik beberapa kesimpulan utama sebagai berikut:

1. Arsitektur *hybrid edge-cloud control* berhasil dirancang dengan membagi pemrosesan secara terdistribusi antara perangkat *edge* untuk autentikasi wajah, RFID, dan PIN, gateway lokal Raspberry Pi 4B untuk pengelolaan database lokal SQLite, serta platform cloud untuk penyimpanan data terpusat yang diintegrasikan secara asinkron menggunakan broker pesan RabbitMQ.
2. Sistem berhasil diimplementasikan secara fisik pada Ruang Dosen Gedung AA Politeknik Negeri Jakarta. Implementasi perangkat *bridge* nirkabel berhasil mengatasi kendala jangkauan sinyal ESP-NOW yang terdegradasi akibat jarak. Pada pengujian *one trip* jarak 1 meter, 3 meter, dan 5 meter, rata-rata *latency* meningkat dari 89,89 ms menjadi 312,69 ms dan 510,45 ms dengan *packet loss* tertinggi 12,00% pada jarak 5 meter. Sedangkan pada pengujian *full trip*, rata-rata *latency* meningkat dari 104,59 ms menjadi 352,4 ms dan 663,2 ms dengan *packet loss* sebesar 1,60% pada jarak 5 meter. Membatasi jangkauan di bawah 3 meter dengan *bridge* terdekat terbukti dapat menjaga pengiriman citra wajah 21 KB berjalan optimal dengan *packet loss* minimal.
3. Analisis kinerja menunjukkan sistem memiliki keandalan respons yang sangat tinggi baik dalam kondisi *online* maupun *offline*. Pada pengujian secara bersamaan sinkronisasi data, arsitektur asinkron RabbitMQ pada Raspberry Pi 4B terbukti sangat efisien dengan rata-rata delay hanya 66,3 ms pada 2 koneksi, 484 ms pada 50 koneksi, dan 788 ms pada 100 koneksi dengan tingkat *packet loss* tetap 0%. Saat *offline*, akses pintu diproses lokal menggunakan database SQLite pada gateway, dan log aktivitas disinkronkan otomatis tanpa kehilangan data saat koneksi internet pulih kembali berkat mekanisme *message acknowledgment* RabbitMQ.



5.2 Saran

Adapun beberapa saran yang dapat diajukan untuk pengembangan sistem ini di masa mendatang adalah sebagai berikut:

1. Menambahkan *end-to-end encryption* pada paket data citra wajah yang dikirimkan via ESP-NOW guna menjamin privasi dan keamanan data biometrik pengguna dari ancaman penyadapan.
2. Menerapkan metode kompresi data atau serialisasi biner (seperti Protocol Buffers atau MessagePack sebagai pengganti JSON) pada payload pesan RabbitMQ guna mengoptimalkan konsumsi bandwidth jaringan, khususnya ketika terjadi pengiriman pesan sinkronisasi log massal setelah gateway kembali daring (online).
3. Menerapkan protokol keamanan tambahan seperti TLS pada broker pesan RabbitMQ untuk memastikan proses sinkronisasi data database lokal dengan *cloud* berjalan melalui saluran yang terenkripsi aman.

Hak Cipta :

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



POLITEKNIK
NEGERI
JAKARTA



Hak Cipta :

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

- Arbain, A., Muhammad, M. A., Septiana, T., et al. (2022). Learning hoax news pada local dan cloud computing deployment menggunakan Google App Engine. *JITET*, 10(3), 183. <http://dx.doi.org/10.23960/jitet.v10i3.2646>
- Barus, E., Pardede, K. M., & Manjorang, J. A. P. Br. (2024). Transformasi digital: Teknologi cloud computing dalam efisiensi akuntansi. *Jurnal Sains dan Teknologi*, 5(3), 904–911. <https://doi.org/10.55338/saintek.v5i1.2862>
- Bara, M. S. B., & Hendrawati, T. D. (2023). Rancangan *smart door lock* berbasis *IoT* dengan verifikasi wajah. Dalam *Prosiding SEMNASTERA (Seminar Nasional Teknologi dan Riset Terapan)* (hlm. 451–454). Politeknik Sukabumi.
- Dewi, I. P., & Fikri, R. (2023). Optimalisasi keamanan rumah dengan implementasi sistem notifikasi gerbang cerdas berbasis *Internet of Things*. *Journal of Computer System and Informatics*, 4(4), 816–829. <https://doi.org/10.47065/josyc.v4i4.4004>
- Ishtaiwi, A., Nabot, A., Alzubi, O. A., et al. (2025). HECOCP: Hybrid edge-cloud optimistic concurrency protocol for sensor data transactional services. *IAJIT*, 22(4), Article 10. <https://doi.org/10.34028/iajit/22/4/10>
- Isrofi, A., Utama, S. N., & Putra, O. V. (2021). Rancang bangun robot pemotong rumput otomatis menggunakan wireless kontroler modul ESP32-CAM berbasis *Internet of Things*.
- Li, Z., Fei, F., & Zhang, G. (2022). Edge-to-cloud *IIoT* for condition monitoring in manufacturing systems with ubiquitous smart sensors. *Sensors*, 22(15), 5901. <https://doi.org/10.3390/s22155901>
- Maturi, M. H., Podicheti, S., & Kumar, D. (2024). Optimizing energy efficiency in edge-computing environments with dynamic resource allocation.
- Mudaliar, M. D., & Sivakumar, N. (2020). *IoT* based real time energy monitoring system using Raspberry Pi. *Internet of Things*, 12, 100292. <https://doi.org/10.1016/j.IoT.2020.100292>
- Murjitama, F. L., Raihan, H. N., Adiwijaya, R. P., et al. (2024). The *smart door lock* using face recognition access based on *Internet of Things*. *TEKNIKA*, 13(2), 199–203. <https://doi.org/10.34148/teknika.v13i2.816>
- Nizam, M. N., Yuana, H., & Wulansari, Z. (2022). Mikrokontroler ESP32 sebagai alat monitoring pintu berbasis web. *JATI*, 6(2), 767.



Hak Cipta :

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

- Permadi, A., et al. (2024). *Edge computing in IoT: Challenges and opportunities for bringing computation closer to data sources*. Journal of Computing and Emerging Systems. <https://journals.stmjournals.com/joces/article=177260>
- Putri, H., Hadiyoso, S., Fatoni, S. B. P., Octaviany, V., Wulandari, A., Aprilina, R., & Rosmiati, M. (2025). Security system for door locks using YOLO-based face recognition. *International Journal on Informatics Visualization*, 9(1), 224–230.
- Robi, M., & Okra, R. (2023). Perancangan sistem informasi pengajuan judul skripsi dengan notifikasi WhatsApp di IAIN Bukittinggi. *Indo Green Journal*, 1(1), 1–10. <https://idngreen.com/index.php/green>
- Saragih, K. A., & Kurniawan, R. (2025). Sistem penyiraman otomatis berbasis *IoT* dengan logika fuzzy Sugeno untuk pengendalian kelembaban tanah di greenhouse. *Algoritma*, 22(1). <https://doi.org/10.33364/algoritma/v.22-1.2327>
- Simadiputra, V., & Surantha, N. (2021). Rasefiberry: Secure and efficient Raspberry-Pi based gateway for smarthome *IoT* architecture. *Bulletin of Electrical Engineering and Informatics*, 10(2), 1035–1045. <https://doi.org/10.11591/eei.v10i2.2741>
- Subito, M., Blestania, T., Amin, N., & Dewi, S. (2023). Rancang bangun sistem monitoring level permukaan air berbasis *Internet of Things (IoT)*. *FS*, 14(1). <https://doi.org/10.54757/fs.v14i1.247>
- Syarifullah, M. R., Prayudha, J., & Syahril, M. (2023). Penerapan edge detection sistem pada proses identifikasi keamanan rumah berbasis ESP Cam. *Jurnal Sistem Komputer TGD*, 2(6), 325–332.
- Tawakal, M. I., & Ramdhani, Y. (2021). Smart lock door menggunakan akses e-KTP berbasis *Internet of Things*. *JTI*, 3(1). <https://doi.org/10.51977/jti.v3i1.417>
- Waruwu, M. (2024). Metode penelitian dan pengembangan (R&D): Konsep, jenis, tahapan dan kelebihan. *Jurnal Ilmiah Profesi Pendidikan*, 9(2), 1220–1230. <https://doi.org/10.29303/jipp.v9i2.2141>
- Zein, A., Eriana, E. S., & Farizy, S. (2024). Sistem informasi pengelolaan gedung sekolah menggunakan *Internet of Things*. *Dinamika: Jurnal Pengabdian Masyarakat*, 2(2), 110–115. <https://doi.org/10.56457/dinamika.v2i2.665>



Hak Cipta :

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



Muhammad Haikal Fadhillah lahir di Bogor pada tahun 2004. Merupakan anak ketiga dari tiga bersaudara, berdomisili di Bogor. Lulus pendidikan dasar di SDN Citeureup 04 pada tahun 2016. Kemudian melanjutkan pendidikan menengah pertama di SMPN 1 Cibinong dan lulus pada tahun 2019. Kemudian melanjutkan pendidikan menengah kejuruan di SMKN 1 Cibinong dan lulus pada tahun 2022. Kemudian melanjutkan pendidikan sebagai mahasiswa Diploma Empat (D4) Politeknik Negeri Jakarta (PNJ) dengan jurusan Teknik Informatika dan Komputer dengan prodi Teknik Multimedia dan Jaringan.

**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

```

Hasil_Uji_FullTrip_1m.csv
1  Test_Ke,Waktu_Mulai_t0(ms),Waktu_Selesai_t3(ms),RTT_Delay(ms)
2  1,5156,5271,115
3  2,7258,7371,113
4  3,8881,8987,106
5  4,10818,10919,101
6  5,13308,13423,115
7  6,15076,15185,109
8  7,17242,17336,94
9  8,19291,19394,103
10 9,20976,21077,101
11 10,23377,23479,102
12 11,25564,25659,95
13 12,27235,27351,116
14 13,29318,29411,93
15 14,31695,31784,89
16 15,33468,33574,106
17 16,35522,35614,111
18 < Alt+H abis bimbingan.md View 7 edited files plot_full_trip_comparison.py Alt+L >
19 18 38627 38734 107
  
```

**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

```

Hasil_Uji_FullTrip_5m.csv
1  Test_Ke,Waktu_Mulai_t0(ms),Waktu_Selesai_t3(ms),RTT_Delay(ms)
2  1,5467,6213,746
3  2,7937,8675,738
4  3,9539,10232,693
5  4,11635,12390,755
6  5,14026,14735,709
7  6,16471,17153,682
8  7,18747,19423,676
9  8,21215,21908,693
10 9,23385,24087,702
11 10,25793,26573,780
12 11,27645,28335,690
13 12,30071,30770,699
14 13,32065,32768,703
15 14,34354,35050,696
16 15,36337,37053,716
17 16,38331,38423,670
18 17,40325,39423,670
19 18,42319,40423,669

< Alt+H abis bimbingan.md View 7 edited files plot_full_trip_comparison.py Alt+L >

Hasil_Uji_FullTrip_10m.csv
20 19,42668,43452,784
21 20,44563,45275,712
22 21, TIMEOUT, TIMEOUT, TIMEOUT
23 22,49272,50021,749
24 23,51289,52082,793
25 24,53594,54359,765
26 25,55946,56740,794
27 26,57806,58601,795
28 27,59991,60793,802
29 28, TIMEOUT, TIMEOUT, TIMEOUT
30 29,64038,64844,806
31 30,65668,66458,790
32 31,67799,68573,774
33 32,70154,71017,863
34 33,72116,72841,725
35 34,73745,78865,5120
36 35,76035,76735,700
37 36,78035,78735,700

< Alt+H abis bimbingan.md View 7 edited files plot_full_trip_comparison.py Alt+L >

```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

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

```

Hasil_Uji_FullTrip_15m.csv
1  Test_Ke,Waktu_Mulai_t0(ms),Waktu_Selesai_t3(ms),RTT_Delay(ms)
2  1,TIMEOUT,TIMEOUT,TIMEOUT
3  2,TIMEOUT,TIMEOUT,TIMEOUT
4  3,9909,11098,1189
5  4,TIMEOUT,TIMEOUT,TIMEOUT
6  5,TIMEOUT,TIMEOUT,TIMEOUT
7  6,TIMEOUT,TIMEOUT,TIMEOUT
8  7,18058,19276,1218
9  8,TIMEOUT,TIMEOUT,TIMEOUT
10 9,21824,22930,1106
11 10,23712,24865,1153
12 11,TIMEOUT,TIMEOUT,TIMEOUT
13 12,TIMEOUT,TIMEOUT,TIMEOUT
14 13,TIMEOUT,TIMEOUT,TIMEOUT
15 14,32206,33362,1156
16 15,TIMEOUT,TIMEOUT,TIMEOUT
17 16,TIMEOUT,TIMEOUT,TIMEOUT
18 < Alt+H abis bimbingan.md View 7 edited files plot_full_trip_comparison.py Alt+L >
19 18 TIMEOUT TIMEOUT TIMEOUT
  
```

```

Hasil_Uji_FullTrip_20m.csv
1  Test_Ke,Waktu_Mulai_t0(ms),Waktu_Selesai_t3(ms),RTT_Delay(ms)
2  1,TIMEOUT,TIMEOUT,TIMEOUT
3  2,7773,9287,1514
4  3,TIMEOUT,TIMEOUT,TIMEOUT
5  4,TIMEOUT,TIMEOUT,TIMEOUT
6  5,TIMEOUT,TIMEOUT,TIMEOUT
7  6,15789,17348,1559
8  7,TIMEOUT,TIMEOUT,TIMEOUT
9  8,20342,21901,1559
10 9,TIMEOUT,TIMEOUT,TIMEOUT
11 10,TIMEOUT,TIMEOUT,TIMEOUT
12 11,25968,27454,1486
13 12,TIMEOUT,TIMEOUT,TIMEOUT
14 13,TIMEOUT,TIMEOUT,TIMEOUT
15 14,TIMEOUT,TIMEOUT,TIMEOUT
16 15,34632,36190,1558
17 16,36680,38865,1457
18 < Alt+H abis bimbingan.md View 7 edited files plot_full_trip_comparison.py Alt+L >
19 18 40907 42295 1388
  
```



© 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_Uji_OneTrip.csv
704 713,1782672795806.10,1782672796253.61,447.50
705 714,1782672799257.11,1782672799707.36,450.24
706 715,1782672802708.12,1782672803156.07,447.95
707 716,1782672806158.07,1782672806608.17,450.10
708 717,1782672809609.58,1782672810057.39,447.81
709 718,1782672813060.64,1782672813511.67,451.04
710 719,1782672816511.55,1782672816958.14,446.59
711 720,1782672819961.58,1782672820411.14,449.56
712 721,1782672823413.29,1782672823872.33,459.05
713 722,1782672826864.34,1782672827314.81,450.47
714 723,1782672830314.34,1782672830762.62,448.28
715 724,1782672833764.74,1782672834213.48,448.73
716 725,1782672837216.59,1782672837664.22,447.64
717 726,1782672840667.39,1782672841115.06,447.68
718 727,1782672844118.11,1782672844566.39,448.28
719 728,1782672847568.53,1782672848019.64,451.11
720 729,1782672851019.67,1782672851469.38,449.71
721 < Alt+H abis bimbingan.md View 7 edited files plot_full_trip_comparison.py Alt+L >
722 731,1782672857922.05,1782672858369.19,447.14
  
```

Source code: <https://github.com/ultearsativa/SMARTDOORLOCK.git>

POLITEKNIK
NEGERI
JAKARTA