Implementasi Aplikasi Smart Dorm Lock untuk Monitoring dan Controlling Pintu Pintar Asrama Telkom University berbasis Face Recognition

1st Haichel Anggy Paro Simamora
School of Electrical Engineering
Telkom University
Bandung, Indonesia
haichelanggi@student.telkomuniversity.ac.id

2nd Dr. Rita Purnamasari, S.T., M.T.
School of Electrical Engineering
Telkom University
Bandung, Indonesia
ritapurnamasari@telkomuniversity.ac.id

3rd Efri Suhartono, S.T., M.T.
School of Electrical Engineering
Telkom University
Bandung, Indonesia
efrisuhartono@telkomuniversity.ac.id

Abstrak – *Smart Dorm Lock* merupakan *mobile application* yang dirancang untuk mempermudah pengawas asrama dalam *monitoring* dan *controlling* penghuni asrama ketika akses keluar masuk asrama. Dengan memanfaatkan teknologi *face recognition*, aplikasi dapat memantau siapa saja yang memasuki asrama dan waktunya. Penelitian ini berfokus pada implementasi sisi *frontend* menggunakan *android studio*. Fitur utama pada aplikasi meliputi *History Access*, *Dorm List*, Registrasi, *Delete Dataset*, *Start Detection* dan *Stop Detection*. Hasil menunjukkan nilai *Throughput* 10,611 Kbps, *Packet Loss* 0%, *Delay* 292,7 ms dan *Jitter* 1,68 ms yang menandakan bahwa nilai tersebut merupakan kategori baik menurut standar ITU-TG.1010

Kata Kunci – *Smart Dorm Lock, Face Recognition, Android Studio, ITU-T G.1010*

I. PENDAHULUAN

Kemajuan teknologi *Internet of Things* (*IoT*) mendorong lahirnya sistem keamanan pintu pintar yang mampu dioperasikan secara otomatis dan terintegrasi dengan perangkat aplikasi. Sistem keamanan konvensional pada pintu asrama dinilai kurang efektif karena tidak menyediakan kemampuan pengawasan jarak jauh serta pencatatan aktivitas akses secara *real-time*. Penelitian sebelumnya di lingkungan Telkom University telah membuktikan keberhasilan penerapan pengolahan citra digital seperti metode *Histogram of Oriented Gradients* (HOG) dan *Haar Cascade* dalam mendeteksi wajah untuk membuka pintu asrama secara otomatis [1]. Namun, sistem tersebut berfokus pada aspek identifikasi wajah pada perangkat pintu dan belum menyediakan platform aplikasi yang mendukung proses monitoring dan controlling dari sisi pengguna atau pengawas asrama.

Berdasarkan hal tersebut, penelitian ini difokuskan untuk mengembangkan aplikasi *Smart Dorm Lock* berbasis *face recognition* yang tidak hanya berfungsi sebagai sistem autentikasi akses pintu, melainkan juga memungkinkan pengawasan pintu secara real-time, pencatatan histori akses, serta kontrol pembukaan pintu jarak jauh melalui perangkat Android. Dengan adanya fitur *monitoring* dan *controlling* dalam aplikasi, diharapkan pengelolaan keamanan pintu asrama menjadi lebih efisien, terpusat, dan modern seiring kebutuhan keamanan lingkungan yang semakin dinamis.

II. KAJIAN TEORI

Pengembangan aplikasi *Smart Dorm Lock* membutuhkan pemahaman terhadap berbagai teknologi dan konsep yang digunakan dalam membangun aplikasi yang fungsional dan mudah untuk diintegrasikan. Kajian teori ini menjelaskan tentang dasar yang mendukung implementasi, termasuk penggunaan *Android Studio, Kotlin* dan *Firebase*. Serta *Quality of Service* untuk menilai kualitas kualitas jaringan aplikasi dalam mengirim dan menerima data ke sistem.

A. *Android Studio*

Android Studio merupakan *Integrated Development Environment* (IDE) resmi yang dikembangkan oleh Google untuk membangun aplikasi berbasis *Android*. *Android Studio* dirancang khusus untuk mendukung seluruh siklus pengembangan aplikasi *mobile*, mulai dari penulisan kode, *debugging*, hingga *deployment* ke *Google Play Store* [2].

B. *Kotlin*

Kotlin merupakan bahasa pemrograman modern yang dikembangkan oleh JetBrains dan secara resmi didukung oleh Google sebagai bahasa utama dalam pengembangan aplikasi *Android. Kotlin* dirancang untuk meningkatkan produktivitas *developer* melalui sintaks yang ringkas, aman dari kesalahan *null pointer*, serta kompatibel penuh dengan *Java* sehingga memudahkan proses migrasi maupun integrasi kode [3].

C. *Firebase*

Firebase merupakan sebuah *platform Backend-as-a-Service* (BaaS) yang dikembangkan oleh Google, dirancang untuk membantu pengembang membangun aplikasi *mobile* dan *web* dengan lebih cepat tanpa harus menangani kompleksitas pengelolaan server secara langsung. *Platform* ini menyediakan berbagai layanan yang saling terintegrasi, seperti *Firebase Realtime Database*, *Cloud Firestore, Authentication*, *Cloud Storage*, hingga *Firebase Cloud Messaging* (FCM) [4].

D. *Quality of Service (QoS)*

Quality of Service (QoS) merupakan mekanisme yang digunakan untuk menjamin kualitas layanan jaringan agar

sesuai dengan kebutuhan suatu aplikasi, khususnya pada aspek *bandwidth*, *delay*, *jitter*, dan *packet loss*. *QoS* bekerja dengan melakukan prioritas terhadap jenis trafik tertentu sehingga performa komunikasi data dapat berjalan stabil dan reliabel, terutama pada aplikasi *real-time* seperti video *streaming, VoIP*, maupun sistem *IoT* yang membutuhkan tingkat ketepatan waktu tinggi [5].

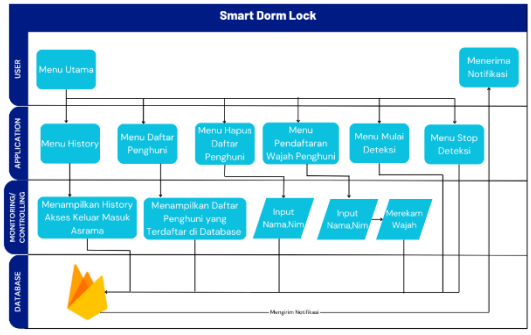
III. METODE

Metode yang digunakan dalam penelitian ini difokuskan pada proses perancangan dan implementasi aplikasi *Smart Dorm Lock* . Sebagai bagian dari sistem *Face Recognition*, aplikasi memiliki peran penting dalam menghubungkan pengawas asrama dengan berbagai fitur sebagai *monitoring* dan *controlling* sistem.

A. Perancangan Sistem

Penelitian ini menggunakan *Android Studio* sebagai komponen utama untuk pembuatan aplikasi mobile. Sistem ini memiliki fungsi *controlling* dan *monitoring* dalam menghubungkan pengguna dengan sistem *face recognition*. Untuk monitoring, pengawas asrama dalam melihat *history access* siapa yang memasuki asrama beserta dengan keterangan waktu yang *real-time* dan dapat melihat siapa saja penghuni asrama yang sudah terdaftar dalam aplikasi. Pada fungsi *controlling*, pengawas asrama dapat menjalankan sistem *face recognition* hanya dengan fitur pada aplikasi seperti, memulai dan menghentikan proses deteksi, mendaftarkan penghuni wajah penghuni asrama ke dalam sistem, dan menghapus data penghuni asrama dari sistem. Kombinasi dua proses ini membuat sistem mampu diintegrasikan dengan baik dan *real-time*.

B. Arsitektur Sistem



Gambar 3. 1 Arsitektur Sistem

Pada gambar 3.1 merupakan arsitektur aplikasi *Smart Dorm Lock* yang dirancang menggunakan empat lapisan utama yaitu *User*, *Application*, *Monitoring/Controlling*, dan *Database*. Pada lapisan *User*, pengawas asrama mengakses aplikasi melalui *menu utama* untuk berpindah ke berbagai fitur seperti *menu history*, *menu daftar penghuni*, *menu hapus penghuni*, *menu pendaftaran wajah penghuni*, *menu mulai deteksi*, dan *menu stop deteksi*. Lapisan *Application* berfungsi sebagai inti dari sistem aplikasi Android yang menyediakan antarmuka untuk menampilkan histori akses keluar–masuk pintu asrama, menampilkan daftar penghuni yang tersimpan pada database, melakukan input data penghuni berupa nama dan NIM, serta merekam wajah penghuni untuk proses registrasi sistem *face recognition*. Lapisan *Monitoring/Controlling* menangani proses pengawasan dan pengendalian, seperti menampilkan data akses secara *real-time* dan mengirimkan perintah dibukanya pintu, yang seluruh aktivitasnya direkam pada *Database* menggunakan layanan Firebase. Selain menyimpan data registrasi wajah dan identitas penghuni, lapisan *Database* juga mengirimkan *push notification* sebagai pemberitahuan masuknya akses pintu kepada pengguna aplikasi.

C. Implementasi Sistem

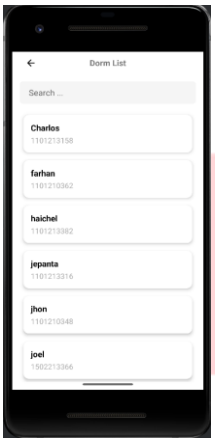
Aplikasi *Android monitoring* dan *controling* dalam sistem keamanan asrama merupakan sebuah perangkat lunak yang dirancang secara khusus untuk memantau dan mengelola aktivitas akses pintu asrama secara *real-time* melalui perangkat ponsel *Android*. Aplikasi ini berfungsi sebagai antarmuka pengguna yang memungkinkan pengawas asrama untuk melihat status sistem keamanan, merekam aktivitas keluar-masuk penghuni, serta mengelola data wajah yang telah tersimpan dalam database. Seluruh proses dilakukan dengan cepat dan efisien melalui satu platform yang mudah dioperasikan. Dalam sistem ini, aplikasi diberi nama "*Smart Dorm Lock*" yang merepresentasikan fungsi utamanya dalam menjaga keamanan akses pintu asrama berbasis pengenalan wajah.

Gambar 3. 2 Halaman Utama

Pada halaman ini menampilkan bagian utama dari aplikasi. Pada menu ini pengguna dapat memilih fitur yang akan dibuka selanjutnya seperti menu *history access*, *delete dataset*, *dorm list*, *stop detection* dan *detection*. Semua menu ini disusun secara sederhana dan mudah dipahami, sehingga memudahkan pengawas asrama dalam mengoperasikan aplikasi sesuai kebutuhan.

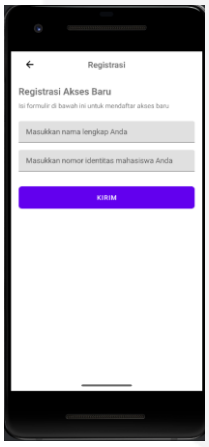
Gambar 3. 3 Halaman *History Access*

Pada halaman *history access* berisi sekumpulan data dalam aplikasi *Android* yang digunakan untuk menampilkan data historis dalam bentuk nama, NIM, jam, tanggal, bulan, dan tahun. Dengan demikian pada menu ini dapat terpantau siapa saja yang memasuki asrama. Selain itu, untuk memudahkan penjaga asrama dalam melakukan pencarian penghuni asrama bisa dilakukan melalui *searchbar* berdasarkan nama atau nim.



Gambar 3. 4 Halaman *Dorm List*

Pada halaman *Dorm List* berisi sekumpulan data penghuni asrama yang terdaftar di sistem *Smart Dorm Lock*. Data yang ditampilkan berupa nama mahasiswa dan NIM masing-masing penghuni yang sudah berhasil melakukan proses pendaftaran wajah melalui fitur *Regist*. Data yang ada pada halaman ini akan otomatis diperbaharui apabila terdapat penghuni asrama baru yang telah menyelesaikan proses pendaftaran wajah.



Gambar 3. 5 Halaman *Regist*

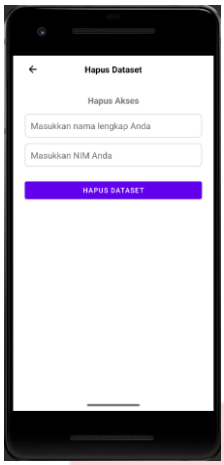
Pada halaman *Regist*, penghuni asrama dapat melakukan proses pendaftaran akses wajah ke dalam sistem *Smart Dorm Lock*. Proses ini dimulai dengan mengisi dua kolom formulir yang tersedia, yaitu nama lengkap dan nim. Formulir ini wajib diisi dengan benar karena akan menjadi identitas utama yang terkait langsung dengan data wajah di dalam sistem. Setelah data dimasukkan sistem akan secara otomatis mengaktifkan kamera perangkat untuk melakukan proses pengambilan gambar wajah.

Gambar 3. 6 Halaman *Start Detection*

Pada halaman Detection, pengguna dapat mengirimkan perintah untuk memulai proses deteksi wajah secara manual. Menu ini berfungsi sebagai pemicu agar sistem kembali menjalankan modul deteksi wajah, terutama dalam kondisi ketika sistem baru saja diaktifkan, sedang tidak dalam mode deteksi aktif, atau setelah sebelumnya digunakan untuk proses lain seperti registrasi wajah mahasiswa. Tampilan pada halaman ini menunjukkan bahwa aplikasi sedang dalam proses mengirimkan perintah deteksi ke perangkat pengolah, dan notifikasi berupa teks "Perintah deteksi wajah dikirim!" akan muncul sebagai indikator bahwa sistem telah menerima dan mulai menjalankan perintah tersebut.

Gambar 3. 7 Halaman *Stop Detection*

Pada halaman *Stop Detection*, tersedia tombol yang memungkinkan pengawas asrama untuk menghentikan proses deteksi wajah secara manual. Fitur ini sangat penting digunakan apabila sistem perlu beralih ke proses lain seperti registrasi wajah mahasiswa baru, penghapusan data dataset wajah, atau saat perangkat tidak sedang digunakan dalam waktu tertentu untuk menghemat sumber daya. Dengan menghentikan deteksi, sistem akan berhenti memproses input dari kamera hingga perintah deteksi diaktifkan kembali melalui menu *Detection*.

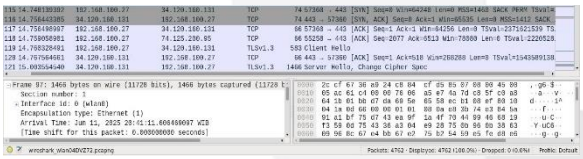


Gambar 3. 8 Halaman *Delete Dataset*

Sebagai pelengkap dari fitur registrasi, aplikasi juga menyediakan halaman *Delete Dataset*, yang berfungsi untuk menghapus data wajah mahasiswa dari sistem. Fitur ini hanya dapat diakses oleh pengawas asrama dan digunakan jika terdapat mahasiswa yang tidak lagi tinggal di asrama, sehingga datanya perlu dihapus untuk menjaga efisiensi dan keamanan sistem dan apabila terjadi kesalahan dalam pengisian nama atau nim saat proses registrasi, sehingga data perlu dihapus dan dilakukan registrasi ulang.

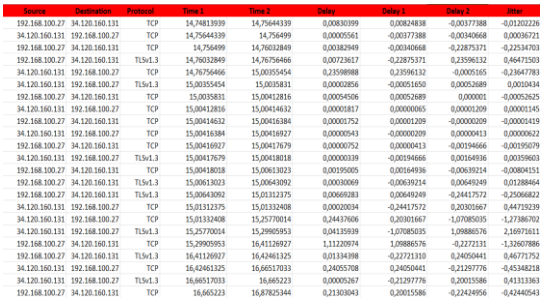
IV. HASIL DAN PEMBAHASAN

Pengujian *Quality of Service* (*QoS*) pada sistem *Smart Dorm Lock* dilakukan untuk mengevaluasi kinerja komunikasi antara perangkat sistem dan aplikasi. Parameter yang diuji meliputi *throughput*, *packet loss*, *delay*, *dan jitter*. Pengujian ini bertujuan untuk memastikan bahwa proses pengiriman notifikasi dan data verifikasi wajah berjalan secara *real-time*, sehingga sistem dapat memberikan respon yang cepat dan akurat dalam berbagai kondisi jaringan.



Gambar 4. 1 Tampilan *Wireshark*

Pada gambar 4.1 menunjukkan Tampilan aplikasi Wireshark menunjukkan aktivitas lalu lintas jaringan secara *real-time* menggunakan tools open-source yang berfungsi untuk memantau, merekam, dan menganalisis paket data yang melewati jaringan. Dalam konteks sistem Smart Dorm Lock, *Wireshark* digunakan untuk merekam seluruh aktivitas komunikasi data antara aplikasi Android dengan server Firebase, termasuk saat pengguna melakukan pendaftaran wajah, pendeteksian wajah, pengiriman data nama dan NIM, hingga proses kontrol pintu dan penerimaan notifikasi. Dengan memantau lalu lintas ini, seluruh tindakan yang dilakukan pada aplikasi yang terhubung langsung ke sistem keamanan pintu asrama dapat dianalisis secara detail, sehingga sangat membantu dalam proses *troubleshooting*, pengujian keamanan, serta pengukuran performa jaringan guna memastikan sistem berjalan secara *real-time* dan stabil.



Source	Destination	Protocol	Time 1	Time 2	Delay	Delay 1	Delay 2	Jitter
192.168.100.27	34.120.160.131	TCP	14,74813939	14,75644339	0,00830399	0,00824838	-0,00377388	-0,01202226
34.120.160.131	192.168.100.27	TCP	14,75644339	14,756499	0,00005561	-0,00377388	-0,00340668	0,00036721
192.168.100.27	34.120.160.131	TCP	14,756499	14,76032849	0,00382949	-0,00340668	-0,22875371	-0,22534703
192.168.100.27	34.120.160.131	TLSv1.3	14,76032849	14,76756466	0,00723617	-0,22875371	0,23596132	0,46471503
34.120.160.131	192.168.100.27	TCP	14,76756466	15,00355454	0,23598988	0,23596132	-0,0005165	-0,23647783
34.120.160.131	192.168.100.27	TLSv1.3	15,00355454	15,0035831	0,00002856	-0,00051650	0,00052689	0,0010434
192.168.100.27	34.120.160.131	TCP	15,0035831	15,00412816	0,00054506	0,00052689	0,000001	-0,00052625
34.120.160.131	192.168.100.27	TCP	15,00412816	15,00414632	0,00001817	0,00001065	0,00001209	0,00001145
192.168.100.27	34.120.160.131	TCP	15,00414632	15,00416384	0,00001752	0,00001209	-0,00000209	-0,00001419
34.120.160.131	192.168.100.27	TCP	15,00416384	15,00416927	0,00000543	-0,00000209	0,00000413	0,00000622
192.168.100.27	34.120.160.131	TCP	15,00416927	15,00417679	0,00000752	0,00000413	-0,00194666	-0,00195079
34.120.160.131	192.168.100.27	TLSv1.3	15,00417679	15,00418018	0,00000339	-0,00194666	0,00164936	0,00359603
192.168.100.27	34.120.160.131	TCP	15,00418018	15,00613023	0,00195005	0,00164936	-0,00639214	-0,00804151
192.168.100.27	34.120.160.131	TLSv1.3	15,00613023	15,00643092	0,00030069	-0,00639214	0,00649249	0,01288464
192.168.100.27	34.120.160.131	TLSv1.3	15,00643092	15,01312375	0,00669283	0,00649249	-0,24417572	-0,25066822
34.120.160.131	192.168.100.27	TCP	15,01312375	15,01332408	0,00020034	-0,24417572	0,20305667	0,44739239
34.120.160.131	192.168.100.27	TCP	15,01332408	15,25770014	0,24437606	0,20301667	-1,07085035	-1,27386702
34.120.160.131	192.168.100.27	TLSv1.3	15,25770014	15,29905953	0,04135939	-1,07085035	1,09886576	2,16971611
192.168.100.27	34.120.160.131	TCP	15,29905953	16,41126927	1,11220974	1,09886576	-0,2272131	-1,32607886
192.168.100.27	34.120.160.131	TLSv1.3	16,41126927	16,42461325	0,01334398	-0,22721310	0,24050441	0,46771752
34.120.160.131	192.168.100.27	TCP	16,42461325	16,66517033	0,24055708	0,24050441	-0,21297776	-0,45348218
34.120.160.131	192.168.100.27	TLSv1.3	16,66517033	16,665223	0,00005267	-0,21297776	0,20015586	0,41313363
192.168.100.27	34.120.160.131	TCP	16,665223	16,87825344	0,21303043	0,20015586	-0,22424956	-0,42440543

Gambar 4. 2 Perhitungan *QoS* dengan Excel

Terlihat pada gambar 4.2 IP address yang telah difilter untuk proses transmisi pengiriman paket selama pengujian adalah 192.168.100.27 yang merupakan *IP address* dari perangkat *Raspberry Pi* sedangkan 34.120.160.131 merupakan *IP address* dari aplikasi dimana seluruh data hasil deteksi wajah yang berhasil dari *Raspberry Pi* dikirim ke aplikasi.

Quality of Service merupakan parameter-parameter yang menunjukkan kualitas paket data jaringan. Pengujian *QoS* ini dikhususkan untuk perhitungan *Troughput, Packet Loss, delay* dan *jitter*. *Delay* merupakan berapa lama waktu yang dibutuhkan saat data melakukan proses transmisi dari asal ke tujuan sedangkan *jitter* merupakan variasi delay antara paket data yang diterima *packet loss*. Di bawah ini adalah tabel yang menunjukan standar *ITU-T G.1010* untuk *QoS* setiap kategori yang akan digunakan untuk membandingkan dengan data yang dihasilkan menggunakan *software Wireshark*.

Tabel 4. 1 Standar ITU-T G.1010 untuk *Delay*

Kategori	Delay (ms)	Indeks
Sangat Bagus	<150 ms	4
Bagus	150 ms s/d 300 ms	3
Sedang	300 ms s/d 450 ms	2
Jelek	>450 ms	1

Tabel 4. 2 Standar ITU-T G.1010 untuk *Jitter*

Kategori	*Jitter* (ms)	Indeks
Sangat Bagus	0 ms	4
Bagus	1 ms s/d 75 ms	3
Sedang	75 ms s/d 125 ms	2
Jelek	125 ms s/d 225 ms	1

Tabel 4. 3 Standar ITU-T G.1010 untuk *Packet Loss*

Kategori	*Packet loss* (%)	Indeks
Sangat Bagus	0%	4
Bagus	1% s/d 3%	3
Sedang	3% s/d 15%	2
Jelek	>25%	1

Tabel 4. 4 Hasil Pengujian *Quality of Service (QoS)*

Througput	Packet Loss	Delay	Jitter
Jumlah *Bytes/Time Span*	[(Paket Dikirim – Paket Diterima)/Paket Dikirim]x100	Total *delay* = 237,96702540	Total *jitter* = 1,37116008
= 1326,383 x 8 = 10,611 Kbps	= 0%	= 292,7 ms	= 1,68 ms

Didapatkan hasil yang memuaskan untuk pengujian *QoS* dalam parameter *Throughput, Packet Loss, Delay* dan *jitter*. Pengujian ini menggunakan dua protokol utama yang telah di filter hanya Parameter Hasil Kategori berdasarkan standar *ITU-T G.1010* untuk *QoS* Throughput 10,611 Kbits nilai *throughput* ini menunjukkan kecepatan pengiriman data dari perangkat *Raspberry Pi* ke aplikasi yang tergolong cukup baik untuk aplikasi *IoT* yang hanya mengirimkan notifikasi atau data ringan seperti informasi wajah yang terdeteksi. Dengan rata-rata ukuran paket yang relatif kecil yaitu 496 *Bytes*, sistem mampu mempertahankan kestabilan dalam transmisi data. Nilai *throughput* ini juga menandakan bahwa sistem memiliki efisiensi yang memadai dalam memanfaatkan jaringan, yang penting untuk menjaga keandalan notifikasi *real-time*. *Packet Loss* 0% [Sangat Bagus], *Delay* 292,7 ms [Bagus] dan *jitter* 1,68 ms [Bagus] untuk protokol TCP dan *TLSv1.2*. Protokol TCP digunakan untuk mengirim data dalam jaringan berbasis IP sedangkan protokol TLSv1.2 digunakan untuk menyediakan komunikasi yang aman dan terenskripsi melalui jaringan komputer.

V. KESIMUPLAN

Berdasarkan hasil pengujian Quality of Service (QoS) pada komunikasi data yang dilakukan, diperoleh nilai *throughput* rata-rata sebesar 10,611 Kbps dengan rata-rata ukuran paket sebesar 439 Bytes, serta nilai *delay*, *jitter*, dan *packet loss* yang masih berada dalam rentang standar layanan jaringan, sehingga dapat disimpulkan bahwa performa jaringan sudah cukup baik dan stabil untuk mendukung pengiriman notifikasi serta proses kontrol *sistem Smart Dorm Lock* secara *real-time* tanpa terjadi gangguan akses. Dengan demikian, aplikasi ini dapat menjadi solusi yang diandalkan untuk mengatasi masalah kemanan pada asrama secara cepat dan tepat.

REFERENCES

[1] M. I. Yoren, R. Purnamsari, E. Suhartono (2024). "Penerapan Metode Histogram Oriented Of Gradients Dan Haar-Cascad Pada Pintu Asrama Pintar Telkom University." *e-Proceeding of Engineering*, vol. 11, no. 6, pp. 6478-6489, Dec. 2024.

[2] Wibowo, A., dan Ramadhani, T. 2021. "Pengembangan Aplikasi Mobile Menggunakan Android Studio dan Kotlin." Jurnal Teknologi Informasi dan Komputer, Vol. 9, No. 2, pp. 112–119.

[3] Gherlas, M. N., & Muntean, C. I. (2020). Kotlin vs Java in Android development: A comparative study. *IEEE 22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, 405–409.

[4] Google, "Firebase Documentation," *Firebase*, 2025. [Online]. Tersedia: https://firebase.google.com/docs. [Diakses: 10-Juli-2025].

[5] Sulaiman, M., & Budiarto, R. (2021). Implementation of Quality of Service (QoS) in network traffic management to improve transmission efficiency. *Indonesian Journal of Electrical Engineering and Computer Science*, 23(2), 698–706.

[6] Kurniasyah, I., & Rakhmawati, L. 2025. "Smart Door Lock Innovation Using Integration of Bluetooth Low Energy and MQTT IoT Protocol." Indonesian Journal of Electrical and Electronics Engineering (INAJEEE), vol. 8, no. 1.

[7] Ayuni, R., & Mauliana, P. 2025. "Smart Door Lock System Berbasis Aplikasi Blynk dengan Face Recognition menggunakan Metode Convolutional Neural Network (CNN)." Kohesi: Jurnal Sains dan Teknologi, vol. 7, no. 4, pp. 91–100.

[8] Nur, M., Sulistyowati, H. S., & Nurrohman, A. 2024. "Penerapan Face Recognition untuk Model Smart Lock Door berbasis IoT." Jurnal Teknologi Informasi dan Digital, vol. 2, no. 1, pp. 152–166.

[9] Saputra, R., & Surantha, N. 2024. "Smart and real-time door lock system for an elderly user based on face recognition." Bulletin of Electrical Engineering and Informatics, vol. 10, no. 3.

[10] R. A. Pratama, "Pengembangan Aplikasi Android Dengan Kotlin: Panduan Praktis Untuk Pemula," Duniadata.org, vol. 1, pp. 1-21, 2024.

[11] Sugiyatno. (2023). Pengiriman informasi real time menggunakan teknologi database Firebase pada aplikasi mobile Android. FAHMA – Jurnal Informatika Komputer, Bisnis dan Manajemen, 21(2), 46–55.

[12] Suyatno, & Prasetyo, E. D. (2017). Implementasi Raspberry Pi untuk rancang bangun sistem monitoring dan kontrol jarak jauh pada miniatur greenhouse berbasis IoT. Jurnal Teknologi dan Sistem Komputer, 4(1)

[13] Mujtahidah, A. K., & Oktiawati, U. Y. 2021. "Implementasi dan Analisis QoS pada Smart Door yang Terintegrasi dengan Aplikasi Telegram." *JISE*, vol. 2, no. 1, pp. 17–23.

[14] Nur, M., Sulistyowati, H. S., & Nurrohman, A. 2024. "Penerapan Face Recognition untuk Model Smart Lock Door berbasis IoT." *Jurnal Teknologi Informasi dan Digital*, vol. 2, no. 1, pp. 152–166.

[15] Raharjo, M. A., Ahmad, A., & Wabula, Y. 202X. "Desain dan Analisis Quality of Service Pada Sistem Otomasi Rumah Menggunakan Wireless Sensor Network." *JOSH*, vol. 6, no. 3.