

Pengembangan Aplikasi *Smart Dorm Key* Yang Dapat Memverifikasi Suara untuk Meningkatkan Keamanan Asrama

1st Muhammad Trisucipto
School of Electrical Engineering
Telkom University
Bandung, Indonesia

trisucipto@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

esuhartono@telkomuniversity.ac.id

Abstrak — Keamanan asrama di lingkungan universitas seringkali masih bergantung pada sistem konvensional yang rentan terhadap penyalahgunaan. Untuk mengatasi hal tersebut, dikembangkan sebuah aplikasi seluler sebagai komponen utama dari sistem *Smart Dorm Key* yang bertujuan untuk meningkatkan keamanan melalui verifikasi dua langkah. Aplikasi ini berfungsi sebagai antarmuka utama bagi pengguna untuk melakukan verifikasi suara, yang merupakan tahap pertama sebelum otentikasi biometrik sidik jari pada perangkat keras. Dikembangkan untuk platform Android menggunakan bahasa pemrograman Kotlin, aplikasi ini mengintegrasikan model

Kata kunci— Aplikasi Seluler, Convolutional Neural Network (CNN), Internet of Things (IoT), Keamanan Asrama, Kotlin, Mel-Frequency Cepstral Coefficients (MFCC), Pengenalan Suara.

I. PENDAHULUAN

Lingkungan asrama merupakan ekosistem vital dalam mendukung kehidupan akademik dan sosial mahasiswa, khususnya bagi mahasiswa tahun pertama di institusi pendidikan seperti Universitas Telkom. Di dalam lingkungan ini, keamanan menjadi prioritas utama untuk menjamin ketenangan, kenyamanan, serta melindungi aset pribadi mahasiswa dari berbagai risiko. Namun, banyak sistem keamanan asrama yang ada saat ini masih bergantung pada metode konvensional yang terbukti memiliki banyak keterbatasan.

A. Latar Belakang

Studi kasus di Asrama Telkom University menunjukkan bahwa sistem keamanan yang diterapkan masih bersifat tradisional, seperti penggunaan kunci pintu manual dan pencatatan tamu melalui

log book. Metode ini dinilai kurang memadai, rentan, dan tidak efisien. Kunci manual mudah hilang, dicuri, atau diduplikasi, yang secara langsung membuka celah bagi akses tidak sah. Di sisi lain [11], log book manual sangat bergantung

pada ketelitian petugas dan tidak memiliki mekanisme verifikasi identitas yang kuat, sehingga memungkinkan terjadinya kelalaian atau manipulasi data. Keterbatasan ini tidak hanya meningkatkan risiko kehilangan barang berharga mahasiswa, seperti laptop dan perangkat elektronik lainnya, tetapi juga menimbulkan kekhawatiran yang dapat mengganggu fokus belajar mereka [11].

Seiring dengan kemajuan teknologi yang pesat, khususnya dalam bidang *Internet of Things* (IoT) dan kecerdasan buatan (Artificial Intelligence), muncul peluang untuk merevolusi sistem keamanan konvensional. Teknologi rumah pintar [10], seperti kunci pintu otomatis, telah menunjukkan potensi besar dalam meningkatkan keamanan dan kenyamanan secara signifikan [6]. Berbagai metode autentikasi modern telah diterapkan, seperti

Radio Frequency Identification (RFID), Personal Identification Number (PIN), pengenalan wajah, dan biometrik sidik jari. Namun, penerapan metode tunggal seringkali masih menyisakan celah keamanan. Sebagai contoh, sistem berbasis pengenalan wajah dapat terganggu oleh kondisi pencahayaan yang buruk atau penggunaan aksesoris [1], sementara metode berbasis pola ketukan suara sangat sensitif dan tidak efektif karena rentan terhadap getaran atau suara bising di sekitarnya [5]

B. Rumusan Masalah

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

1. Bagaimana merancang dan membangun sebuah aplikasi seluler yang fungsional sebagai antarmuka utama untuk sistem keamanan asrama berbasis IoT?
2. Bagaimana mengimplementasikan teknologi pengenalan suara berbasis *machine learning* (MFCC dan CNN) ke dalam aplikasi untuk proses verifikasi pengguna yang akurat?
3. Bagaimana arsitektur perangkat lunak yang ideal untuk mendukung komunikasi real-time antara

aplikasi, backend, model machine learning, dan basis data?

4. Seberapa efektif dan andal performa aplikasi *Smart Dorm Key* berdasarkan pengujian akurasi verifikasi suara dan analisis *Quality of Service* (QoS) jaringan?

C. Tujuan

Berdasarkan latar belakang dan analisa masalah, terdapat beberapa tujuan dibuatnya penelitian terkait, diantaranya:

1. Menjelaskan arsitektur perangkat lunak yang digunakan, mencakup frontend aplikasi seluler (Kotlin), backend REST API, layanan machine learning, dan integrasi basis data (Firebase).
2. Mendeskripsikan implementasi fitur-fitur utama pada aplikasi, termasuk pendaftaran pengguna, autentikasi, antarmuka verifikasi suara, dan dasbor pemantauan log aktivitas untuk admin.
3. Menganalisis hasil pengujian performa aplikasi, terutama pada akurasi model pengenalan suara dalam berbagai kondisi (normal, bising, serak) dan kualitas layanan jaringan (*delay* dan *jitter*).
4. Menyajikan bukti kelayakan bahwa aplikasi seluler dengan verifikasi suara dapat menjadi solusi yang efektif untuk meningkatkan keamanan di lingkungan asrama.

II. KAJIAN TEORI

Desain sistem aplikasi seluler *Smart Dorm Key* dirancang untuk menjadi antarmuka utama bagi pengguna.

A. Aplikasi Seluler dalam Sistem IoT

Internet of Things (IoT) adalah sebuah paradigma di mana objek fisik dilengkapi dengan sensor, perangkat lunak, dan teknologi lain yang memungkinkan mereka untuk terhubung dan bertukar data melalui internet. Dalam konteks sistem keamanan, IoT memungkinkan perangkat keras seperti kunci pintu elektronik untuk dikendalikan dan dipantau dari jarak jauh. Aplikasi seluler memainkan peran sentral dalam ekosistem ini, berfungsi sebagai jembatan antara pengguna dan perangkat fisik. Aplikasi menyediakan antarmuka pengguna grafis (Graphical User Interface - GUI) yang memungkinkan pengguna untuk melakukan tindakan kompleks, seperti mendaftarkan data biometrik, melakukan verifikasi, dan memantau log aktivitas, dengan cara yang mudah dan intuitif [11].

Dalam proyek ini, aplikasi dikembangkan untuk sistem operasi Android, mengingat penetrasinya yang luas di kalangan mahasiswa. Bahasa pemrograman yang dipilih adalah Kotlin, yang direkomendasikan secara resmi oleh Google untuk pengembangan Android. Kotlin menawarkan sintaks yang lebih ringkas, fitur keamanan null-safety yang lebih baik, dan interoperabilitas penuh dengan Java,

menjadikannya pilihan yang modern dan efisien untuk membangun aplikasi yang andal dan mudah dikelola [11].

B. Teknologi Pengenalan Suara (*Voice Recognition*)

Pengenalan suara adalah teknologi yang memungkinkan sistem untuk mengidentifikasi individu berdasarkan karakteristik unik dari suara mereka. Teknologi ini digerakkan oleh *machine learning*, yang melatih model untuk mengenali pola-pola kompleks dalam sinyal suara manusia. Proses ini secara umum melibatkan dua tahap utama: ekstraksi fitur dan klasifikasi.

1. Ekstraksi Fitur: *Mel-Frequency Cepstral Coefficients* (MFCC)

Sinyal suara mentah terlalu kompleks untuk dianalisis secara langsung oleh model machine learning. Oleh karena itu, diperlukan teknik ekstraksi fitur untuk mengubah sinyal menjadi representasi numerik yang ringkas namun informatif. *Mel-Frequency Cepstral Coefficients* (MFCC) adalah teknik standar industri dalam pemrosesan suara yang dirancang untuk meniru persepsi pendengaran manusia. Telinga manusia lebih sensitif terhadap perubahan frekuensi rendah dibandingkan frekuensi tinggi, sebuah properti yang dimodelkan oleh Skala Mel [1]. Proses ekstraksi MFCC melibatkan serangkaian langkah matematis yang sistematis:

- a. *Pre-Emphasis*: Memperkuat komponen frekuensi tinggi pada sinyal untuk menyeimbangkan spektrum energi [1].
- b. *Frame Blocking*: Memecah sinyal suara menjadi segmen-segmen kecil yang tumpang tindih (disebut *frame*), biasanya berdurasi 20-30 milidetik, untuk analisis jangka pendek [1].
- c. *Windowing*: Menerapkan fungsi jendela (misalnya, *Hamming window*) pada setiap *frame* untuk mengurangi diskontinuitas di tepian *frame* [1].
- d. *Fast Fourier Transform* (FFT): Mengubah setiap *frame* dari domain waktu ke domain frekuensi untuk menganalisis kandungan frekuensinya [1].
- e. *Mel-Frequency Wrapping*: Menerapkan filter bank pada skala Mel ke spektrum daya untuk memodelkan persepsi pendengaran manusia [1].
- f. *Discrete Cosine Transform* (DCT): Menerapkan DCT pada energi log dari filter bank untuk menghasilkan koefisien MFCC yang tidak berkorelasi [1].

Hasil akhir dari proses ini adalah serangkaian vektor fitur (koefisien MFCC) yang secara efektif merepresentasikan karakteristik akustik dari suara, yang siap digunakan untuk melatih model klasifikasi.

2. Klasifikasi: *Convolutional Neural Network* (CNN)

Convolutional Neural Network (CNN) adalah arsitektur *deep learning* yang sangat efektif dalam tugas pengenalan pola, awalnya dipopulerkan untuk analisis gambar [1]. Keunggulan CNN terletak pada kemampuannya untuk secara otomatis belajar fitur-fitur hierarkis dari data input. Dalam konteks pengenalan suara, fitur MFCC yang diekstraksi dari sinyal suara dapat diperlakukan sebagai "gambar" 1D atau

2D, di mana satu sumbu merepresentasikan waktu dan sumbu lainnya merepresentasikan koefisien MFCC.

CNN memproses "gambar" spektrogram ini melalui beberapa lapisan utama:

- Lapisan Konvolusi (*Convolution Layer*): Lapisan ini menerapkan serangkaian filter (kernel) yang dapat dipelajari ke data input. Setiap filter bertugas mendeteksi pola-pola spesifik, seperti bentuk vokal atau konsonan dalam spektrogram.
- Lapisan Pengumpulan (*Pooling Layer*): Lapisan ini berfungsi untuk mengurangi dimensi spasial dari peta fitur (feature map), yang membuat model lebih efisien secara komputasi dan lebih tahan terhadap variasi kecil pada posisi fitur (invarian).
- Lapisan Terhubung Penuh (*Fully Connected Layer*): Setelah fitur-fitur penting diekstraksi dan diringkas, lapisan ini bertugas untuk melakukan klasifikasi akhir berdasarkan fitur-fitur tersebut dan menghasilkan probabilitas untuk setiap kelas suara.

Kombinasi MFCC dan CNN telah terbukti sangat efektif dalam berbagai tugas pengenalan suara karena memanfaatkan representasi fitur yang relevan secara perseptual (MFCC) dan kemampuan klasifikasi pola yang kuat (CNN) [1] [2].

C. Arsitektur Backend dan Basis Data

Untuk menghubungkan aplikasi seluler frontend dengan model machine learning dan mengelola data pengguna, diperlukan arsitektur backend yang solid. Arsitektur ini umumnya terdiri dari REST API (Representational State Transfer Application Programming Interface) dan basis data. REST API bertindak sebagai perantara yang menangani permintaan (request) dari aplikasi, seperti permintaan login atau verifikasi suara, dan memberikan respons (response) yang sesuai. Dalam proyek ini, backend REST API dikembangkan menggunakan *Express.js*, sebuah kerangka kerja *Node.js* yang ringan dan fleksibel. Untuk penyimpanan data, basis data real-time seperti *Firebase* dari *Google Cloud Platform* menjadi pilihan yang sangat cocok. *Firebase* memungkinkan sinkronisasi data secara instan antara klien (aplikasi seluler) dan server, yang sangat penting untuk fitur seperti pemantauan log aktivitas secara real-time.

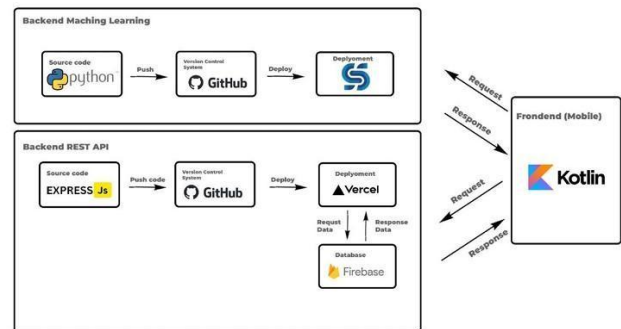
III. METODE

Metodologi yang digunakan dalam pengembangan aplikasi *Smart Dorm Key* adalah pendekatan Desain dan Rekayasa Sistem Perangkat Lunak. Pendekatan ini mencakup beberapa tahapan sistematis, mulai dari perancangan arsitektur, implementasi kode, hingga pengujian fungsional dan performa.

A. Arsitektur Perangkat Lunak

Perancangan *smart dorm key* ini memerlukan rangkaian arsitektur dari perangkat lunak untuk membantu optimalisasi aplikasi yang akan digunakan. Terdapat beberapa bagian dalam perancangan perangkat lunak ini, yaitu *backend* dari

machine learning, backend REST API dan frontend untuk mobile. Dalam perancangan bagian tersebut digunakan beberapa software pendukung yang akan ditampilkan pada gambar di bawah.



Gambar 1 Arsitektur Perangkat Lunak

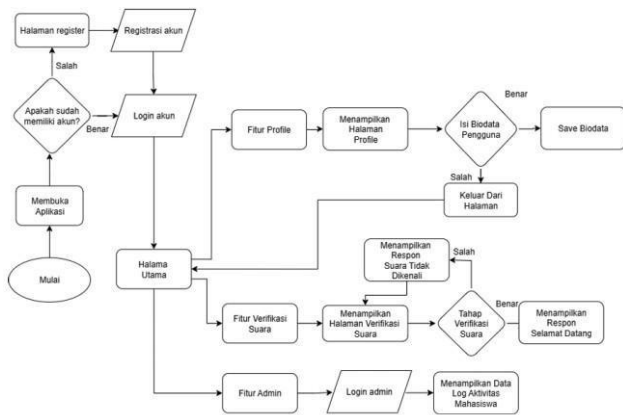
Arsitektur perangkat lunak dirancang dengan pendekatan client-server yang modular, memisahkan antara antarmuka pengguna, logika bisnis, dan pemrosesan data. Seperti yang diilustrasikan pada Gambar 1, arsitektur ini terdiri dari tiga komponen utama:

- Frontend (Aplikasi Seluler)** : Dikembangkan sebagai aplikasi Android native menggunakan bahasa pemrograman Kotlin. Komponen ini bertanggung jawab penuh atas *User Interface* (UI) dan *User Experience* (UX), menangani input dari pengguna (termasuk perekaman suara), menampilkan data, dan berkomunikasi dengan backend melalui permintaan HTTP ke REST API. Struktur proyeknya diorganisir ke dalam paket-paket logis seperti *login*, *register*, *session*, *UI*, dan verifikasi untuk kemudahan pengelolaan.
- Backend (REST API)**: Dibangun menggunakan *Express.js* dan berfungsi sebagai otak dari sisi server. Tanggung jawabnya meliputi:
 - Mengelola autentikasi dan sesi pengguna (*registrasi*, *login*, *logout*).
 - Menyediakan endpoint untuk diakses oleh aplikasi seluler.
 - Berinteraksi dengan basis data *Firebase* untuk menyimpan dan mengambil data pengguna serta log aktivitas.
 - Bertindak sebagai perantara (*proxy*) yang meneruskan data audio dari aplikasi ke layanan machine learning untuk diproses.

3. **Layanan Machine Learning**: Model CNN yang telah dilatih untuk pengenalan suara di-deploy sebagai layanan mandiri menggunakan *Python* dan *TensorFlow/Keras*. Layanan ini diekspos sebagai endpoint API terpisah. Ketika menerima file audio, layanan ini akan melakukan ekstraksi fitur MFCC, menjalankan prediksi menggunakan model CNN yang telah dimuat, dan mengembalikan hasil prediksi (label pengguna dan skor keyakinan) dalam format *JSON*.

Seluruh kode, baik untuk backend maupun layanan machine learning, dikelola menggunakan sistem kontrol versi *GitHub* dan di-deploy ke platform cloud (misalnya, *Vercel* atau layanan serupa) untuk memastikan ketersediaan dan skalabilitas.

Gambar 2 merupakan *flowchart* dari aplikasi *Smart Dorm Key*. *Flowchart* ini membantu dalam memberikan gambaran singkat dari aplikasi yang akan dirancang.



Gambar 2 Flowchart Aplikasi Smart Dorm Key

B. Alur Kerja Verifikasi Suara

Alur kerja sistem saat pengguna melakukan verifikasi suara melalui aplikasi adalah sebagai berikut:

1. Pengguna menekan tombol mikrofon pada antarmuka aplikasi *Smart Dorm Key*.
2. Aplikasi merekam frasa kunci yang diucapkan pengguna.
3. File audio yang telah direkam dikirim melalui permintaan HTTP POST ke *endpoint* REST API di *backend*.
4. REST API menerima file audio dan meneruskannya ke *endpoint* layanan *machine learning*.
5. Layanan *machine learning* memproses audio: mengekstraksi fitur MFCC dan melakukan prediksi dengan model CNN.
6. Hasil prediksi (misalnya, {"prediction": "Hazbi", "confidence": 0.95}) dikembalikan ke REST API.
7. REST API mencatat hasil verifikasi (berhasil atau gagal, identitas, waktu) ke dalam basis data Firebase sebagai log aktivitas.
8. REST API kemudian mengirimkan respons konfirmasi (sukses atau gagal) kembali ke aplikasi seluler.
9. Aplikasi menerima respons dan menampilkan pesan yang sesuai kepada pengguna (misalnya, "Verifikasi Berhasil" atau "Verifikasi Gagal").

C. Pengumpulan Dataset dan Pelatihan Model

Pengembangan model *machine learning* yang akurat dimulai dengan pengumpulan dataset yang representatif.

1. Pengumpulan Data: Dataset suara dikumpulkan dari tiga pengguna terdaftar (Hazbi, Ito, Faiq) dan beberapa pengguna tidak terdaftar (diklasifikasikan sebagai Unknown). Setiap pengguna merekam frasa kunci ("Pin") berulang kali dalam berbagai kondisi untuk menangkap variabilitas alami. Total dataset untuk pengguna terdaftar adalah 1664 sampel, dan untuk pengguna tidak terdaftar adalah 988 sampel

2. Pelatihan Model: Dataset yang telah di-label kemudian digunakan untuk melatih model CNN. Data dibagi menjadi set pelatihan dan set pengujian. Model dilatih selama beberapa *epoch* hingga mencapai tingkat akurasi yang memuaskan pada data validasi. Hasil akhir pelatihan menunjukkan akurasi tes final sebesar 92.75%

D. Prosedur Pengujian Aplikasi

Evaluasi aplikasi dilakukan melalui beberapa skenario pengujian yang komprehensif:

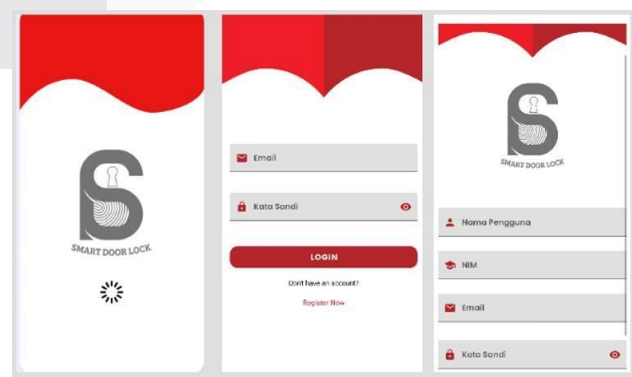
- Pengujian Fungsional: Memastikan semua fitur aplikasi, seperti registrasi, login, pembaruan profil, verifikasi suara, dan tampilan log admin, berfungsi sesuai alur yang dirancang pada *flowchart* (Gambar 2).
- Pengujian Akurasi Verifikasi Suara: Menguji kemampuan aplikasi dalam mengenali suara pengguna terdaftar dan menolak pengguna tidak terdaftar dalam tiga kondisi berbeda: normal, berisik, dan serak. Setiap skenario diuji sebanyak 15 kali per partisipan untuk pengumpulan data kuantitatif.
- Pengujian Performa Jaringan (QoS): Menggunakan perangkat lunak Wireshark untuk menangkap paket data yang ditransmisikan antara aplikasi dan server selama 5 menit operasi. Parameter yang diukur adalah delay (latensi) dan jitter (variasi delay) untuk menilai responsivitas dan stabilitas koneksi jaringan. Analisis dilakukan berdasarkan standar TIPHON.
- Evaluasi Model *Machine Learning*: Menganalisis hasil prediksi model pada data uji menggunakan *Confusion Matrix* dan *Classification Report* untuk mengukur metrik seperti *Precision*, *Recall*, dan *F1-Score* per kelas.

IV. HASIL DAN PEMBAHASAN

Bagian ini menyajikan dan menganalisis temuan utama dari implementasi dan pengujian aplikasi *Smart Dorm Key*.

A. Desain Antarmuka dan Pengalaman Pengguna

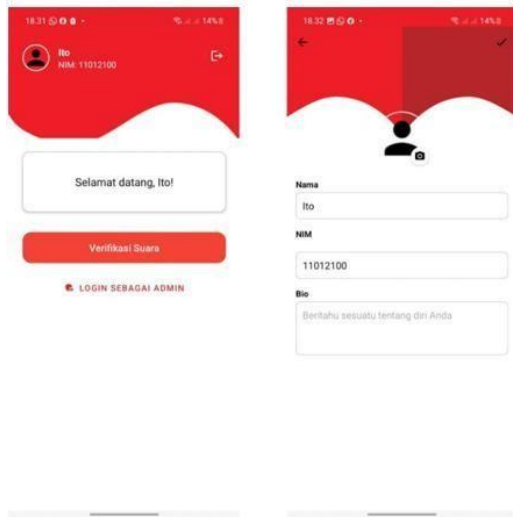
Aplikasi *Smart Dorm Key* dirancang dengan antarmuka yang bersih dan alur navigasi yang intuitif.



Gambar 3 Tampilan Awal Aplikasi

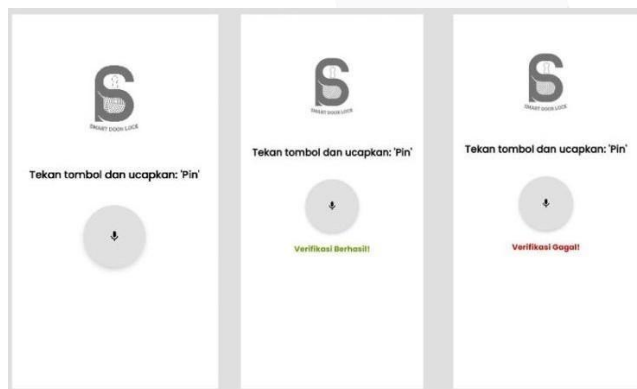
Pada gambar 3 merupakan tampilan awal aplikasi yang ditunjukkan kepada pengguna sebelum ke halaman Utama. Pada gambar sebelah kiri merupakan logo aplikasi

yang akan muncul ketika pengguna membuka aplikasi. Pada gambar yang ada ditengah menampilkan halaman selanjutnya setelah animasi logo selesai, yaitu halaman bagi pengguna untuk melakukan login ketika sudah mempunyai akun, akan tetapi jika pengguna belum memiliki akun, maka pengguna dapat melakukan registrasi seperti yang ditampilkan pada gambar sebelah kiri dengan mengisi biodata diri.



Gambar 4 Tampilan Dalam Aplikasi

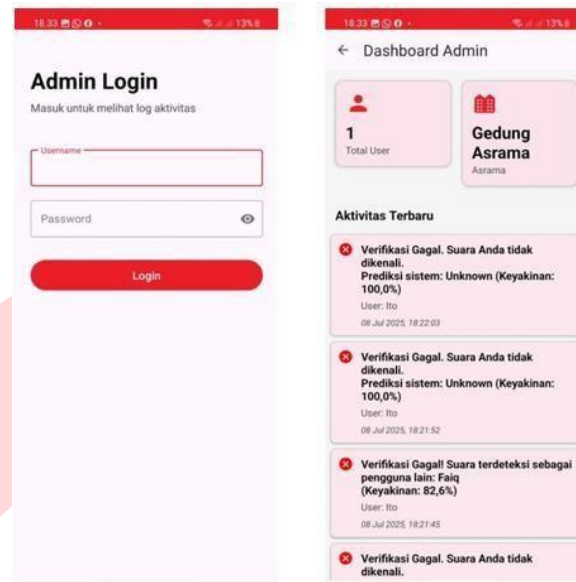
Pada Gambar 4 terdapat dua halaman gambar. Pada gambar sebelah kiri menunjukkan tampilan halaman utama dari aplikasi setelah berhasil login dan jika menekan profil yang ada di kiri atas, maka akan beralih ke halaman profil seperti gambar yang ada di sebelah kanan, yang di mana halaman profil berfungsi untuk melengkapi profil. Halaman utama berfungsi untuk meneruskan tahap verifikasi suara yang akan dilakukan nanti dan juga pada halaman utama terdapat button untuk melakukan login sebagai admin, dimana untuk akun dari admin sendiri sudah didaftarkan melalui local database.



Gambar 5 Tampilan Verifikasi Suara

Pada gambar 5 adalah, halaman verifikasi suara aplikasi *Smart Door Lock*, yang merupakan fitur ini untuk mengotentikasi pengguna melalui perintah suara. Kondisi awal halaman verifikasi (kiri) menginstruksikan pengguna dengan tulisan Tekan tombol dan ucapkan: Pin di bawah logo aplikasi, disertai tombol mikrofon besar di tengah layar yang siap menerima input suara. Selanjutnya, skenario ketika verifikasi suara berhasil (tengah) digambarkan dengan munculnya teks Verifikasi Berhasil berwarna hijau di bawah tombol mikrofon setelah pengguna mengucapkan kata kunci

yang benar, menandakan akses diberikan atau pintu telah terbuka. Sebaliknya, jika verifikasi suara gagal karena pengguna mengucapkan kata kunci yang salah atau suara tidak dikenali (kanan), akan tampil teks Verifikasi Gagal berwarna merah di bawah tombol mikrofon, yang mengindikasikan bahwa akses ditolak.



Gambar 6 Tampilan Logbook Admin

Pada Gambar 4.26 merupakan tampilan ketika melakukan login sebagai admin yang ada pada halaman utama aplikasi. Pada gambar sebelah kiri merupakan halaman untuk memasukkan akun yang berupa username dan password yang telah didaftarkan. Pada gambar sebelah kanan merupakan tampilan setelah admin berhasil melakukan login, maka akan muncul halaman yang menampilkan total user yang terdaftar dan gedung asrama serta inti dari halaman ini, yaitu tampilan log aktivitas mahasiswa ketika memasukkan verifikasi suara. Tampilan dari log ini sendiri berupa verifikasi suara yang gagal dan berhasil, tanggal ketika mahasiswa melakukan verifikasi suara serta waktu mahasiswa melakukan verifikasi suara. Dengan adanya log aktivitas ini tentunya akan memudahkan penjaga asrama untuk memantau kapan mahasiswa kembali ke kamar asrama.

B. Analisis Performa Verifikasi Suara

Pengujian akurasi verifikasi suara menghasilkan temuan yang signifikan mengenai keandalan sistem dalam berbagai kondisi.

- Kondisi Normal: Dalam lingkungan yang tenang, sistem menunjukkan performa yang sangat baik. Akurasi untuk mengenali suara terdaftar mencapai 91%, dan akurasi untuk mengidentifikasi suara tidak terdaftar (unknown) mencapai 88%. Tingkat akurasi yang tinggi ini membuktikan bahwa model dasar MFCC-CNN sangat efektif dalam membedakan karakteristik vokal dalam kondisi ideal.
- Kondisi Berisik: Seperti yang diantisipasi, kehadiran kebisingan latar belakang (misalnya, musik atau percakapan) secara signifikan menurunkan performa. Akurasi untuk suara terdaftar turun menjadi 68%, dan untuk suara tidak terdaftar turun menjadi 62%. Penurunan ini terjadi

karena noise mengganggu fitur MFCC yang diekstraksi dari sinyal suara, sehingga menyulitkan model CNN untuk mengidentifikasi pola yang relevan.

- Kondisi Suara Serak: Variasi pada suara pengguna, seperti suara serak akibat kelelahan atau sakit, juga mempengaruhi akurasi. Akurasi untuk suara terdaftar dalam kondisi ini adalah 73%, dan untuk suara tidak terdaftar adalah 66%. Meskipun terjadi penurunan, performanya sedikit lebih baik daripada dalam kondisi bising, menunjukkan bahwa model memiliki tingkat toleransi tertentu terhadap perubahan kualitas vokal alami.

C. Analisis Kualitas Layanan (QoS) Aplikasi

Hasil perhitungan QoS ditampilkan pada tabel 1:

Tabel 1 Hasil Perhitungan Nilai QoS Aplikasi Smart Door Lock

Parameter	Delay	Jitter
Hasil	65.84 ms	0.017 ms
Kategori Tiphon	Sangat Bagus	Bagus
Indeks	4	3

Dari hasil pengujian QoS aplikasi *smart dorm key* menggunakan *Wireshark* seperti yang ada pada tabel 5.12 menampilkan bahwa aplikasi memiliki performa sistem yang unggul dikarenakan nilai dari parameter delay ini sendiri mencapai hasil sebesar 65.84 ms, dimana nilai dari *delay* tersebut berada pada indeks 4 yang berarti sangat bagus pada standar TIPHON. Nilai ini menandakan responsivitas sistem yang tinggi dalam memproses kode IoT dan *machine learning*, serta efisiensi pengiriman data dengan waktu tunggu yang minim. Di sisi lain, parameter *jitter* juga menunjukkan hasil yang cukup bagus yaitu di angka 0,017 ms dan nilai tersebut berada pada indeks 3 yang berarti termasuk kategori bagus pada standar TIPHON yang berarti transmisi data berjalan stabil karena variasi waktu antar paket data yang tidak signifikan.

Pengujian QoS sangat penting untuk memastikan aplikasi tidak hanya akurat tetapi juga responsif. Hasil pengujian menggunakan *Wireshark* (Tabel 1) menunjukkan performa jaringan yang sangat memuaskan:

- Delay (Latensi): Nilai delay rata-rata yang terukur adalah 65.84 ms. Menurut standar TIPHON, nilai di bawah 150 ms dikategorikan sebagai "Sangat Bagus". Delay yang rendah ini memastikan bahwa pengguna mendapatkan umpan balik verifikasi dengan sangat cepat, menciptakan pengalaman pengguna yang mulus tanpa waktu tunggu yang terasa.
- Jitter: Nilai *jitter* yang terukur adalah 0.017 ms. Nilai ini berada dalam kategori "Bagus" menurut standar TIPHON, yang menunjukkan bahwa variasi waktu kedatangan paket data sangat minim. *Jitter* yang rendah menjamin stabilitas dan konsistensi komunikasi *real-time* antara aplikasi dan server.

V. KESIMPULAN

Penelitian ini berhasil mendemonstrasikan pengembangan dan implementasi aplikasi seluler *Smart Dorm Key* sebagai solusi keamanan asrama yang modern dan efektif. Berdasarkan hasil analisis dan pembahasan, dapat ditarik beberapa kesimpulan utama:

- Aplikasi Fungsional dan Intuitif: Aplikasi *Smart Dorm Key* berhasil dirancang dan diimplementasikan pada platform Android menggunakan Kotlin. Aplikasi ini terbukti fungsional dalam menjalankan peran-peran utamanya, termasuk autentikasi pengguna, antarmuka verifikasi suara, dan penyediaan dasbor pemantauan log aktivitas secara real-time untuk admin. Desain antarmuka yang berorientasi pada pengguna memastikan kemudahan dalam pengoperasian.
- Performa Verifikasi Suara yang Andal dalam Kondisi Ideal: Model *machine learning* yang mengkombinasikan MFCC dan CNN menunjukkan akurasi yang tinggi (91% untuk suara terdaftar dan 88% untuk suara tidak terdaftar) dalam kondisi lingkungan normal. Ini membuktikan kelayakan teknologi pengenalan suara sebagai mekanisme verifikasi yang valid.
- Tantangan pada Kondisi Dunia Nyata: Performa aplikasi menurun secara signifikan di lingkungan yang bising dan ketika kualitas vokal pengguna berubah (serak). Hal ini menyoroti keterbatasan model saat ini dan pentingnya augmentasi dataset dengan sampel suara yang lebih beragam untuk meningkatkan robustitasnya di dunia nyata.
- Model yang Berorientasi pada Keamanan: Analisis mendalam terhadap model menunjukkan bahwa sistem dirancang dengan bias ke arah keamanan (*security-first approach*). Model ini sangat efektif dalam menolak akses tidak sah (tidak ada prediksi positif palsu untuk kelas *unknown*), meskipun kadang-kadang mengorbankan kenyamanan dengan salah menolak pengguna yang sah.
- Kualitas Layanan Jaringan yang Unggul: Aplikasi menunjukkan performa transmisi data yang sangat responsif dan stabil, dengan nilai *delay* dan *jitter* yang masuk dalam kategori "Sangat Bagus" dan "Bagus" menurut standar industri. Hal ini memastikan pengalaman pengguna yang lancar dan mendukung keandalan sistem keamanan secara keseluruhan.

Sebagai implikasi, penelitian ini mengonfirmasi bahwa aplikasi seluler yang terintegrasi dengan teknologi biometrik suara dan IoT merupakan alternatif yang jauh lebih unggul dibandingkan sistem kunci konvensional. Untuk pengembangan di masa depan, disarankan untuk fokus pada peningkatan adaptabilitas model *machine learning* terhadap

noise dan variasi vokal melalui teknik *noise reduction* dan augmentasi dataset yang lebih ekstensif.

REFERENSI

- [1] Adhinata, F. D., Rakhmadani, D. P., & Segara, A. J. T. (2021). Pengenalan Jenis Kelamin Manusia Berbasis Suara Menggunakan MFCC dan GMM. *Jurnal Dinda*, 1(1), 1–6.
- [2] Arunashmi, R., Gouda, S. S., & Agrawal, K. (n.d.). Speech Recognition Using MFCC & CNN. *International Journal of Scientific Research and Engineering Development*, 4.
- [3] Gultom, D., & Susanto, M. F. (2020). Studi Aplikasi Smartlock Pada Pintu Rumah Dengan Arduino Berbasis IoT Dengan Sensor Suara. *Prosiding The 11th Industrial Research Workshop and National Seminar*, 239–245.
- [4] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature*, 521, 436–444. doi: 10.1038/nature14539.
- [5] Nagajyothi, D., & Siddaiah, P. (2018). Speech Recognition Using Convolutional Neural Networks. *International Journal of Engineering & Technology*, 7(4.6), 133–137.
- [6] Nugraha, I. T., Patmasari, R., & Irawan, A. I. (2020). Implementasi Membuka Kunci Pintu Otomatis Menggunakan Face Recognition pada Raspberry Pi Berbasis Internet of Thing. *e-Proceeding of Engineering*, 7(1), 707–715.
- [7] Pramudhita, D. A., Azzahra, F., Arfat, I. K., Magdalena, R., & Saidah, S. (2023). Strawberry Plant Diseases Classification Using CNN Based on MobileNetV3-Large and EfficientNet-B0 Architecture. *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, 9(3), 522–534. doi: 10.26555/jiteki.v9i3.26341.
- [8] Swaminathan, S., & Tantri, B. R. (2024). Confusion Matrix-Based Performance Evaluation Metrics. *African Journal of Biomedical Research*, 27, 4023–4031. doi: 10.53555/AJBR.v27i4S.4345.
- [9] Wulandari, R. (2016). Analisis Qos (Quality of Service) Pada Jaringan Internet (Studi Kasus : Upt Loka Uji Teknik Penambangan Jampang Kulon – Lipi). *Jurnal Teknik Informatika dan Sistem Informasi*, 2(2). doi: 10.28932/jutisi.v2i2.454.
- [10] Yoren, M. I., Purnamasari, R., & Suhartono, E. (2024). Penerapan Metode Histogram Oriented Of Gradients Dan Haar-Cascad Pada Pintu Asrama Pintar Telkom University. *e-Proceeding of Engineering*, 11(6), 6487–6489.
- [11] Trisucipto, M., Ashiddiqi, M. H., & Zariaqwila, D. F. (2025). Perancangan Smart Dorm Key Berbasis Voice Recognition dengan Tambahan Sensor Fingerprint sebagai Verifikasi Dua Langkah untuk Meningkatkan Keamanan. *Fakultas Teknik Elektro, Universitas Telkom*.