Implementasi *Voice Recognition* Menggunakan Metode *Mel-Frequency Ceptral Coefficients* dan *Convolutional Neural Network* Pada *Smart Dorm Key* Telkom *University*

1st Muhamad Hazbi Ashiddiqi
School of Electrical Engineering
Telkom University
Bandung, Indonesia
hazbiashiddiqi@student.telkomunivesity.ac.id

2nd 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 Telkom *University* saat ini masih mengandalkan sistem konvensional seperti kunci manual dan *logbook*, yang dinilai kurang memadai, rentan, dan kurang aman. Keterbatasan ini memungkinkan akses yang tidak sah dan meningkatkan risiko kehilangan barang berharga mahasiswa. Untuk mengatasi permasalahan tersebut, dirancanglah sebuah "*Smart Dorm Key*" berbasis pengenalan suara (*voice recognition*) menggunakan *machine learning* dengan metode *Mel-Frequency Cepstral Coefficients* (MFCC) untuk pemrosesan ekstraksi suara dan menggunakan model *Convolutional Neural Networks* (CNN) untuk pengenalan suara. Pengujian sistem dilakukan dengan melibatkan Hazbi berjumlah 475 *dataset*, Ito 712 *dataset*, Faiq 477 *dataset*, dan *Unknown* 988 *dataset*. Terdapat tiga macam kondisi pengujian yaitu dalam keadaan normal, berisik, dan serak. Hasil pengujian menunjukkan bahwa sistem ini dapat mengenali suara dengan akurasi dalam keadaan normal 91% untuk *dataset* suara terdaftar dan 88% untuk *dataset* suara tidak terdaftar, namun dalam keadaan berisik dan serak akurasi berkurang menjadi 68% (terdaftar) dan 62% (tidak terdaftar) untuk keadaan berisik, 73% (terdaftar) dan 66% (tidak terdaftar) dalam keadaan serak.**

***Kata kunci*— CNN, Keamanan Asrama, MFCC, *Smart Dorm Key*, *Voice Recognition*.**

I. PENDAHULUAN

Asrama Telkom University merupakan salah satu sarana kampus yang dibangun sebagai tempat tinggal bagi mahasiswa/i baru pada satu tahun pertama mereka di Telkom University. Asrama merupakan tempat yang sempurna bagi mahasiswa/i baru untuk belajar banyak hal seperti, toleransi dengan sesama, kerjasama dengan tim, kekeluargaan dan banyak hal bermanfaat lainnya. Semua kegiatan yang dilakukan selama berada di asrama akan dibimbing oleh kakak asrama yang disebut dengan *Senior Resident* serta *Helpdesk* yang berada di setiap gedung asrama untuk menjaga rasa keamanan serta kenyamanan bagi seluruh mahasiswa/i yang berada di gedung asrama [1]. Oleh karena itu, penelitian ini melakukan perancangan dengan memanfaatkan teknologi pengenalan suara dengan membangun sistem menggunakan *Mel-Frequency Cepstral Coefficients (MFCC)* untuk ekstraksi fitur dan *Convolutional Neural Network (CNN)* untuk proses klasifikasi memungkinkan sistem mengidentifikasi pengguna secara tepat melalui karakteristik suara mereka.

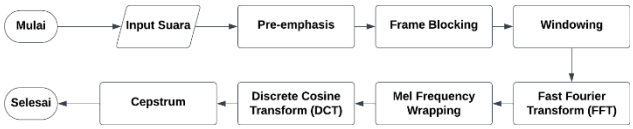
II. KAJIAN TEORI

A. *Machine Learning*

Machine learning adalah teknologi fundamental yang menjadi mesin penggerak di balik sistem *voice recognition* (pengenalan suara). Peran dari *machine learning* ini adalah untuk melatih sistem agar mampu mengenali pola – pola yang sangat kompleks dan bervariasi dalam sinyal suara manusia, kemudia menerjemahkanya menjadi perintah yang dapat dieksekusi. Proses ini dimulai dengan metode seperti *Mel-Frequency Cepstral Coefficients* (MFCC), dengan mengubah rekaman suara analog menjadi data digital. Data ini kemudian diekstraksi, yaitu proses mengambil karakteristik unik yang menjadi pembeda setiap suara. Kumpulan karakteristik inilah yang pada akhirnya dianalisis oleh model *machine learning* untuk dilatih dan mengidentifikasi atau mengklasifikasikan suara tersebut dengan menggunakan kombinasi model *machine learning Convolutional Neural Networks* (CNN) yang efektif dalam mengenali pola lokal pada spektogram, seperti bentuk vokal dan konsonan [2].

B. *Mel-Frequency Cepstral Coefficients* (MFCC)

Mel-Frequency Cepstral Coefficients (MFCC) adalah sebuah teknik ekstraksi fitur yang esensial di bidang *voice recognition*, dirancang untuk mentransformasi sinyal suara menjadi vektor fitur yang ringkas dan informatif. Teknik ini didasarkan pada skala frekuensi *Mel*, sebuah model persepsi pendengaran manusia yang non-linear, di mana sensitivitas lebih tinggi diberikan pada frekuensi rendah. Secara prosedural, sinyal audio disegmentasi, dianalisis dalam domain frekuensi, lalu dilewatkan melalui *filterbank Mel*. Koefisien yang dihasilkan dari transformasi matematis akhir (DCT) secara efektif mengisolasi komponen linguistik dari sinyal (fonem, intonasi) dari variasi non-linguistik seperti amplitudo dan derau [2].



Tujuan dari metode *mel-frequency cepstral coefficients* (MFCC) ini adalah untuk melakukan ekstraksi pada filter dan menghapus bagian sumber data suara. Berikut adalah tahapan metode MFCC :

a. *Pre-Emphasis*

Pre-Emphasis yaitu proses untuk memperkuat komponen frekuensi tinggi pada sinyal suara yang di*input*. Hal ini dilakukan karena pada sinyal suara manusia, energi pada frekuensi tinggi cenderung lebih lemah akibat sifat alami pita suara dan resonansi rongga mulut. Informasi penting untuk mengenali karakter suara banyak terkandung pada frekuensi tinggi [3]. Proses pre-emphasis dilakukan dengan menerapkan filter high-pass sederhana pada sinyal input seperti yang terlihat pada persamaan (2.1) [3].

$$S'_n = S_n - \alpha S_{n-1} \quad (2.1)$$

Dimana :
S_n : nilai sampel ke-n
α : konstanta *pre-emphasis*, $0.9 \leq \alpha \leq 1.0$

b. *Frame Blocking*

Pada bagian *framing* sinyal suara dibagi menjadi beberapa segmen kecil yang disebut *frame* karena sinyal suara manusia bersifat *non-stationary* (karakteristiknya berubah terhadap waktu), sedangkan banyak metode analisis sinyal memerlukan sinyal yang stationer (karakteristiknya konstan pada periode tertentu). Dengan membagi sinyal menjadi *frame-frame* pendek berdurasi sekitar 20-30 milidetik. Selain itu, untuk memastikan informasi pada batas *frame* tidak hilang, *frame-frame* ini biasanya saling tumpeng tindih (*overlap*) sekitar 10-15 milidetik. Hasil dari tahap *framing* adalah serangkaian *frame* yang siap untuk diproses pada domain frekuensi [3].

c. *Windowing*

Setiap *frame* kemudian diberikan *window* untuk meredam efek diskontinuitas pada tiap ujung *frame* yang dapat menyebabkan distorsi pada spektrum frekuensi. Tanpa *windowing*, perbedaan nilai tiba-tiba anatara *frame* dapat menghasilkan efek kebocoran (*spectral leakage*) pada hasil transformasi frekuensi. Untuk itu digunakan fungsi *window function*, seperti *hamming window*, yang mereduksi amplitude sinyal di tepi *frame* menjadi nol secara halus. *Windowing* menhasilkan sinyal *frame* yang lebih mulus, sehingga Ketika dianalisis di domain frekuensi hasilnya menjadi lebih bersih dan akurat [3]. Proses *hamming window* tersebut dapat dituliskan dalam persamaan sebagai berikut .seperti pada persamaan (2.2).

$$w(n) = 0{,}54 - 0{,}46\ cos = \frac{2\pi n}{N-1} \quad (2.2)$$

Dimana :
N : Jumlah sampel pada masing-masing *frame*.
n : 0, 1, 2, 3, …, $N-1$

d. *Fast Fourier Transform* (FFT)

Pada masing-masing *frame* yang sudah di-*windowing* diubah dari domain waktu ke domain frekuensi menggunakan *Fast Fourier Transform* (FFT). Perubahan ke domain frekuensi dilakukan karena ciri khas suara manusia lebih jelas terlihat pada pola distribusi frekuensinya daripada pada bentuk gelombang waktu. FFT menghitung komponen-komponen sinyal pada berbagai frekuensi dan menghasilkan spektrum magnitudo yang merepresentasikan seberapa kuat energi pada setiap frekuensi dalam *frame* tersebut. Hasil dari tahap ini adalah spektrum frekuensi dari masing-masing *frame* [3]. Proses FFT ini dapat ditulis pada persamaan (2.3).

$$f(n) = \sum_{K=0}^{N-1} (Y_k)\, e^{-\frac{2\pi jkn}{N}}, n = 0, 1, 2, \dots, N-1 \quad (2.3)$$

Dimana :
$f(n)$: Frekuensi.
k : 0, 1, 2, …, $(N-1)$
N : Jumlah sampel pada masing-masing *frame*.
j : Bilangan imajiner.
n : 0, 1, 2, 3, …, $(N-1)$

e. *Mel-Frequency Wrapping*

Spektrum frekuensi hasil FFT kemudian diubah ke skala *Mel* untuk menyesuaikan dengan cara telinga manusia mendengar suara. Penelitian psikofisik menunjukkan bahwa telinga manusia lebih sensitif terhadap perbedaan frekuensi di daerah rendah dibandingkan daerah tinggi, sehingga skala *Mel* digunakan untuk merefleksikan persepsi ini. Caranya, spektrum frekuensi dilewatkan melalui sekumpulan filter segitiga yang tersusun dalam skala *Mel*, yang disebut *Mel* filter bank. Setiap filter mengakumulasi energi pada rentang frekuensi tertentu dalam skala *Mel*. Hasil tahap ini berupa vector energi dari masing-masing filter *Mel* yang menggambarkan intensitas suara pada rentang frekuensi yang relevan bagi pendengaran manusia [3]. Untuk menghitung skala *Mel* pada frekuensi dalam Hz ditulis dalam persamaan (2.4).

$$Mel\ f = 2595 \times \log_{10}\left(1 + \frac{f}{700}\right)$$
$$f = 700\ (10^{\frac{m}{2595}} - 1) \quad (2.4).$$

Dimana :
$Mel\ f$: Nilai frekuensi *Mel* dari f.

f. *Discrete Cosine Transform* (DCT)

Tahap akhir dari MFCC adalah DCT pada vektor *log* energi filter *Mel*. Tujuan dari DCT adalah memadatkan informasi ke dalam beberapa koefisien pertama dengan meminimalkan korelasi antar komponen. Koefisien pertama umumnya mewakili energi rata-rata, sementara koefisien-koefisien berikutnya merepresentasikan pola distribusi spektral yang menjadi ciri khas suara. Hasil akhir dari tahap ini adalah sekumpulan koefisien MFCC untuk masing-masing *frame* suara yang kemudian digunakan sebagai fitur dalam proses klasifikasi atau pengenalan suara [3]. Persamaan DCT ini dituliskan seperi pada (2.5).

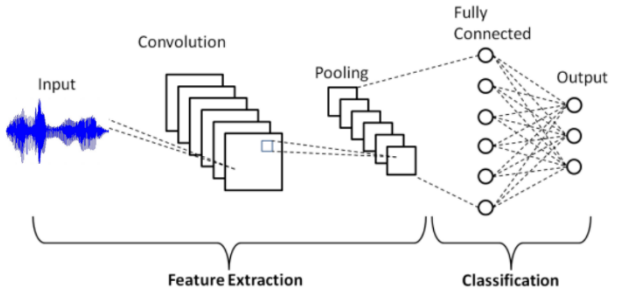
$$C_n = \sum_{K=1}^{K} (logS_k) \cos\left[n\left(k-\frac{1}{2}\right)\frac{\pi}{k}\right], \quad n = 1, 2, \dots, k \qquad (2.5)$$

Dimana :

C_n : Koefisien *cepstrum mel-frequency*
S_k : *Mel Frequency*
n : Bilangan bulat dari 1, …,N (jumlah total sampel).
k : Jumlah koefisien.r

C. *Convolutional Neural Network* (CNN)

Convolutional Neural Network adalah arsitektur *deep learning* yang andal untuk mengenali suara karena keahliannya dalam mengidentifikasi pola visual. Agar bisa memproses suara, CNN mengubah sinyal audio menjadi "gambar" yang disebut spektogram. Sama seperti saat mengenali objek dalam foto, CNN memindai spektogram untuk menemukan ciri-ciri khas ucapan. Jaringan ini belajar secara mandiri untuk mendeteksi komponen-komponen dasar (seperti vokal) dan kemudian menggabungkannya untuk memahami struktur yang lebih kompleks (seperti kata). Keunggulan utamanya adalah CNN dapat menangkap esensi sebuah kata tanpa terganggu oleh perbedaan kecil, misalnya jika diucapkan sedikit lebih cepat atau lebih tinggi [2].



Gambar 2.1 Arsitektur CNN

Berikut adalah penjelasan Gambar 2.1 dengan sumber referensi [8] :

1. Input Sinyal Suara

Input awal adalah sinyal suara mentah dalam bentuk gelombang audio (*waveform*), seperti yang ditunjukkan di paling kiri [4]. Namun, CNN tidak bisa langsung memproses gelombang 1D ini. Oleh karena itu, sinyal ini pertama-tama diubah menjadi representasi visual 2D yang disebut spektogram (contohnya menggunakan MFCC). Spektogram ini adalah "gambar" dari suara, yang memetakan frekuensi terhadap waktu, sehingga bisa dianalisis oleh CNN [4].

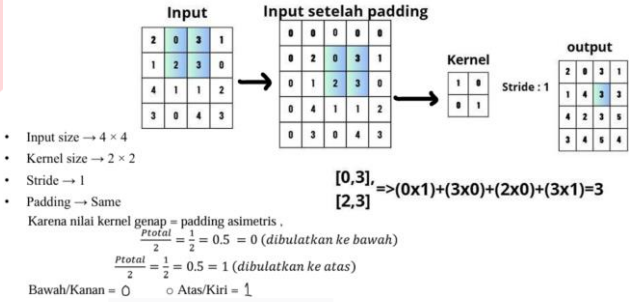
2. Tahap Ekstraksi Fitur

Ini adalah bagian inti di mana CNN menggunakan keahliannya untuk "melihat" pola dalam spektogram. Tahap ini terdiri dari dua lapisan utama yang diulang beberapa kali:

a. *Convolution* (Konvolusi)

Lapisan ini bekerja seperti kaca pembesar yang digerakkan di seluruh "gambar" spektogram untuk menemukan pola-pola kecil. Filter-filter pada lapisan ini secara otomatis belajar untuk mendeteksi fitur-fitur akustik dasar, seperti bentuk vokal, konsonan letup (seperti 'p' atau 't'), atau suara desis (seperti 's') [4]. Hasil dari setiap filter adalah sebuah *Feature Map*, yaitu sebuah peta yang menyorot di mana saja fitur tersebut ditemukan dalam suara [4].

Konvolusi pada arsitektur CNN adalah operasi matematis fundamental yang berfungsi untuk mengekstraksi fitur dari data masukan, khususnya gambar. Proses ini melibatkan sebuah filter atau kernel, yaitu matriks kecil berisi bobot (nilai-nilai yang dipelajari selama pelatihan), yang digeser secara sistematis ke seluruh area gambar. Pada setiap posisi, dilakukan operasi perkalian *element-wise* antara nilai pada kernel dengan bagian gambar yang ditutupinya, lalu semua hasilnya dijumlahkan untuk menghasilkan satu nilai Tunggal seperti pada Gambar 2.2. Kumpulan dari semua nilai tunggal ini membentuk sebuah matriks baru yang disebut *feature map* atau peta fitur.



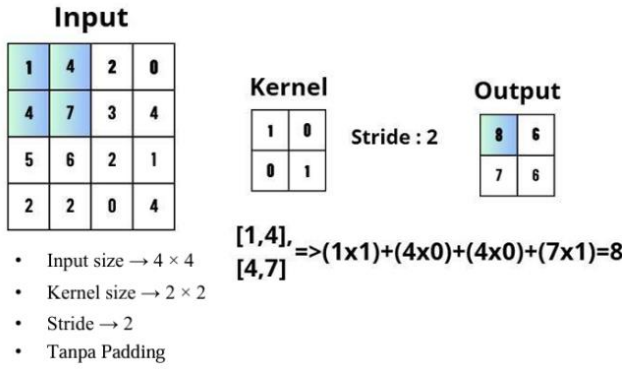
Gambar 2.2 Proses Konvolusi Manual

Tujuan utama dari konvolusi adalah untuk mengidentifikasi pola-pola spesifik seperti tepi, sudut, tekstur, atau bentuk-bentuk yang lebih kompleks pada lapisan yang lebih dalam, sehingga memungkinkan jaringan untuk mengenali objek secara efektif.

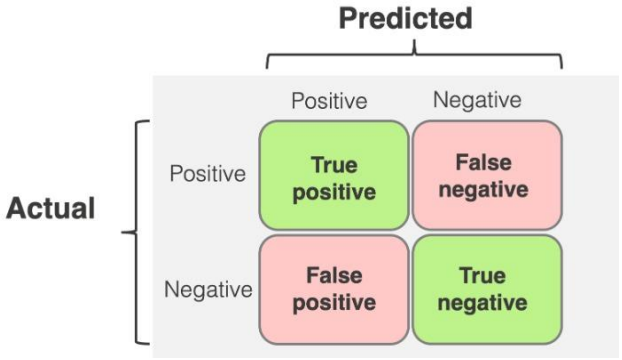
b. *Pooling* (Pengumpulan)

Setelah menemukan banyak fitur, lapisan *pooling* bertugas untuk meringkas dan mereduksi ukuran *feature map*. Cara kerjanya adalah dengan mengambil nilai terpenting (misalnya nilai maksimum atau rata-rata) dari sebuah area kecil [4]. Tujuannya ada dua: (1) Membuat komputasi lebih efisien, dan (2) Membuat model lebih tahan terhadap variasi kecil. Misalnya, jika sebuah kata diucapkan sedikit lebih cepat atau lambat, fitur suaranya mungkin sedikit bergeser di spektogram. *Pooling* membantu model tetap mengenali fitur tersebut meskipun posisinya tidak persis sama [4].

Pooling dalam arsitektur CNN adalah sebuah operasi *down-sampling* yang bertujuan untuk mengurangi dimensi spasial (lebar dan tinggi) dari *feature map* yang dihasilkan oleh lapisan konvolusi. Proses ini bekerja dengan cara meringkas informasi dari sekelompok kecil piksel menjadi satu nilai tunggal. Jenis yang paling umum adalah *Max Pooling*, yang mengambil nilai piksel maksimum dari setiap area (misalnya, area 2x2). Tujuan utama dari pooling adalah untuk mengurangi jumlah parameter dan beban komputasi jaringan, mengontrol *overfitting*, serta membuat representasi fitur lebih tahan (invarian) terhadap pergeseran atau distorsi kecil pada gambar.



Gambar 2.3 Proses *Pooling* pada *Feature Map*



Gambar 2.4 Struktur *Confusion Matrix*

Sebagai contoh, pada Gambar 2.3 proses *Max pooling* dengan *filter* 4x4 dan *stride* 2 akan mengambil sebuah blok nilai, misalnya [1, 4, 4, 7], lalu menghitung hasilnya menjadi 8. Nilai tunggal ini kemudian menjadi bagian dari peta fitur hasil *pooling* (*pooled feature map*) yang baru. Ketika operasi ini diterapkan ke seluruh peta fitur, hasilnya adalah sebuah peta fitur dengan ukuran yang lebih kecil, sehingga lebih efisien untuk diolah oleh lapisan jaringan berikutnya. Dengan demikian, *Max pooling* bekerja dengan cara mengambil nilai piksel maksimum dari setiap blok atau area pada *feature map* serta mampu mengurangi kompleksitas komputasi dan membuat sistem lebih tahan terhadap pergeseran kecil pada gambar, sambil tetap mempertahankan informasi suara dari setiap fitur.

3. Tahap Klasifikasi

Setelah fitur-fitur penting dari suara diekstrak, tahap selanjutnya adalah mengklasifikasikan suara tersebut.

a. *Fully Connected* (Terhubung Penuh)

Data yang sudah ringkas dari tahap *pooling* kemudian "diratakan" menjadi satu baris data panjang dan dimasukkan ke lapisan ini [4].

b. *Output* (Keluaran)

Ini adalah lapisan terakhir yang memberikan hasil akhir. Jumlah neuron (lingkaran) di lapisan ini sesuai dengan jumlah kelas yang ingin dikenali [4]. Misalnya, jika tujuannya adalah mengenali empat perintah ("maju", "mundur", "kiri", "kanan"), maka akan ada empat neuron di lapisan *output*. Neuron dengan nilai probabilitas tertinggi akan menjadi jawaban akhir dari system [4].

D. Model *Performance Evaluate Confusion Matrix*

Confusion Matrix merupakan sebuah matriks tabulasi yang berfungsi sebagai alat evaluasi komprehensif untuk mengukur performa sebuah model klasifikasi. Fungsinya adalah untuk memvisualisasikan kinerja model dengan cara membandingkan setiap hasil prediksi dengan kelas aktualnya. Tabel ini mengkuantifikasi prediksi ke dalam empat kategori: jumlah data yang diprediksi benar (*True Positive* dan *True Negative*) dan jumlah data yang diprediksi salah (*False Positive* dan *False Negative*). Dengan demikian, matriks ini tidak hanya menyimpulkan performa secara umum, tetapi juga memberikan wawasan detail mengenai jenis-jenis kesalahan yang sering dibuat oleh model [5].

Diperlukan metrik evaluasi seperti pada Gambar 2.4 untuk menilai performa klasifikasi dari model yang sudah dilatih. Metrik yang paling sering dipakai dalam tugas klasifikasi adalah *accuracy*. Namun, untuk memperoleh penilaian yang lebih menyeluruh, perlu juga digunakan metrik lain seperti *precision*, *recall*, dan *F1-score* [6].

a. *Recall* (*Sensitivity*)

Recall (*sensitivity*) adalah metrik yang mengukur kemampuan model dalam mendeteksi seluruh kasus positif yang sebenarnya, dengan membandingkan jumlah prediksi positif yang benar terhadap total kasus positif aktual [6]. Rumus untuk menghitung *recall* dapar dilihat pada (2.6).

$$Recall = \frac{TP}{TP + FN} \tag{2.6}$$

b. *Precision*

Precision adalah metrik yang menilai seberapa tepat prediksi positif yang dibuat oleh model, dengan membandingkan jumlah prediksi positif yang benar terhadap seluruh prediksi positif yang dihasilkan [6]. Rumus untuk menghitung *precision* dapat dilihat pada (2.7).

$$Precision = \frac{TP}{TP + FP} \tag{2.7}$$

c. *F1 Score*

F1 Score adalah rata-rata harmonis (*harmonic mean*) dari *precision* dan *recall*. Metrik ini digunakan untuk memberikan keseimbangan antara keduanya, khususnya ketika terjadi ketidakseimbangan jumlah data pada masing-masing kelas [6]. Adapun rumus untuk menghitung *F1 Score* seperti pada (2.8).

$$F1_{score} = 2 \times \frac{presisi\ x\ recall}{presisi + recall} \tag{2.8}$$

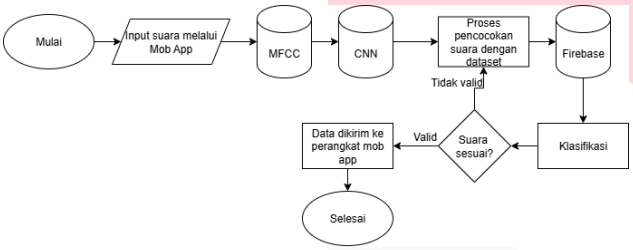
Keterangan :
TP: Prediksi positif, label asli positif.
TN: Prediksi negatif, label asli negatif.
FP: Prediksi positif, label asli negatif.
FN: Prediksi negatif, label asli positif.

III. METODE

A. Perancangan Sistem

sistem yang dirancang terdiri dari dua komponen utama, yaitu perangkat "*Smart Dorm Key*" yang berbasis IoT dan aplikasi berbasis *mobile* yang berfungsi sebagai antarmuka pengguna. Pada aplikasi *mobile* berfungsi sebagai media bagi pengguna untuk mendaftarkan data pengguna, serta untuk melakukan tahap pertama verifikasi yaitu pemindaian suara. Sistem ini didukung oleh machine learning yang menggunakan metode *Mel-Frequency Cepstral Coefficients (MFCC)* untuk ekstraksi fitur suara dan model *Convolutional Neural Networks (CNN)* untuk proses pengenalan suara. Sementara itu, perangkat keras "*Smart Dorm Key*" menggunakan mikrokontroler ESP32 sebagai unit kendali pusat.

Pada Gambar 3.1 yang disajikan dibawah ini merupakan diagram alir dari machine learning *Smart Dorm Key* yang akan dirancang.



Gambar 3.1 Diagram *Alir Machine Learning*

Proses ini dimulai dengan pengambilan *dataset* suara yang menjadi input utama sistem dan kemudian dataset suara akan diproses menggunakan metode MFCC. Setelah data selesai diproses oleh kemudian akan dilakukan prosesan lebih lanjut menggunakan CNN. Kemudian akan dilakukan pencocokan suara, di mana *input* suara akan dibandingkan dengan *dataset* yang tersedia. Hasil dari proses pencocokan suara akan disimpan dalam *Server*. Selanjutnya, sistem melakukan klasifikasi suara berdasarkan hasil pencocokan untuk menentukan identifikasi suara sesuai dengan *dataset* yang ada. Kemudian ketika pengguna menginputkan suara, sistem akan memverifikasi apakah suara tersebut *valid* atau tidak. Jika suara *valid* maka data akan diteruskan ke perangkat *mobile application* tetapi jika suara tidak *valid* sistem akan kembali ke tahap pencocokan suara untuk melakukan verifikasi ulang.

B. Implementasi Sistem

Kombinasi CNN dengan MFCC (*Mel-Frequency Cepstral Coefficients*) sering digunakan dalam aplikasi pemrosesan sinyal *audio*, seperti pengenalan ucapan, identifikasi pembicara, atau klasifikasi suara. MFCC adalah representasi fitur standar dalam pemrosesan *audio* yang menggambarkan spektrum daya logaritmik sinyal pada skala frekuensi *Mel*. Fitur MFCC mengekstraksi karakteristik penting dari *timbre* suara manusia, sehingga sangat cocok untuk analisis suara. Ketika MFCC dikombinasikan dengan CNN, fitur MFCC yang biasanya berbentuk vektor atau matriks waktu-frekuensi diperlakukan sebagai "gambar" masukan 1D atau 2D untuk CNN. CNN kemudian dapat mempelajari pola spasial dan temporal dalam fitur MFCC tersebut, seperti perubahan frekuensi atau durasi yang penting untuk mengidentifikasi kata, emosi, atau pembicara. Pendekatan ini memanfaatkan kemampuan MFCC untuk merepresentasikan fitur *audio* yang relevan dengan persepsi manusia, sekaligus memanfaatkan kemampuan CNN untuk belajar fitur kompleks dari representasi tersebut secara otomatis [2], [7].

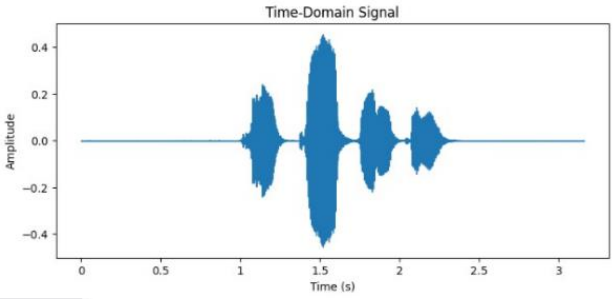
Proses awal dalam pengembangan sistem *machine learning* untuk verifikasi suara pada "*Smart Dorm Key*" adalah pengumpulan dataset suara. *Dataset* ini harus mencakup beragam sampel suara dari pengguna yang sah dan tidak sah, dengan mempertimbangkan variasi seperti intonasi, kecepatan bicara, dan kondisi lingkungan yang berbeda (misalnya, adanya *noise* latar belakang).

Setelah pengumpulan, setiap sampel suara dalam *dataset* akan melalui proses anotasi data. Anotasi ini melibatkan pemberian label yang akurat untuk setiap rekaman suara, mengidentifikasi apakah suara tersebut berasal dari pengguna yang terdaftar ("sah") atau tidak terdaftar (tidak sah). Selain itu, anotasi juga dapat mencakup informasi meta data lainnya seperti identitas pembicara, waktu rekaman, dan kondisi lingkungan saat rekaman dibuat.

Tabel 3.1 *Class* Berdasarkan Jenis Suara

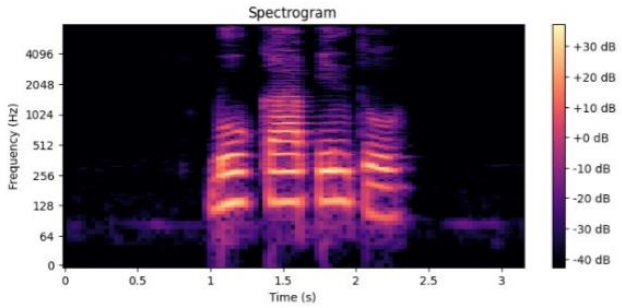
No.	Class	Jumlah Dataset
1	Hazbi	475
2	Ito	712
3	Faiq	477
4	*Unknown*	988

Tabel 3.1 merupakan daftar dari *class* suara yang telah didaftarkan berdasarkan jenis suara pengguna dan juga total dari dataset pada setiap label.



Gambar 3.2 Dataset Suara Mentah

Implementasi pelatihan dataset suara pada Gambar 3.2 menggunakan metode MFCC dimulai dari data suara mentah, yang merupakan sinyal domain waktu, yang menunjukkan amplitudo suara seiring waktu.



Gambar 3.3 Spektogram

Langkah pertama adalah mengubah sinyal ini menjadi representasi frekuensi-waktu. Hal ini dilakukan dengan proses *framing*, di mana sinyal dibagi menjadi bingkai-bingkai pendek yang tumpang tindih, lalu pada setiap bingkai diterapkan transformasi *Fast Fourier Transform* (FFT) untuk menganalisis kandungan frekuensinya. Hasil dari seluruh bingkai ini kemudian digabungkan untuk membentuk spectrogram Gambar 3.3, yang memvisualisasikan energi pada berbagai frekuensi dari waktu ke waktu. Spektrogram ini kemudian diolah lebih lanjut untuk mengekstrak fitur yang lebih relevan dengan persepsi pendengaran manusia. Kumpulan fitur MFCC dari seluruh *dataset* inilah yang kemudian digunakan sebagai *input* untuk melatih model *Convolutional Neural Network.*

IV. HASIL DAN PEMBAHASAN

Skenario pengujian untuk *smart dorm key* berbasis *voice recognition* akan dirancang untuk mengevaluasi akurasi sistem dalam berbagai kondisi lingkungan dan penggunaan. Pengujian akan melibatkan sejumlah partisipan yang akan berperan sebagai pengguna, dengan setiap partisipan menjalani serangkaian pengujian dalam beberapa skenario utama, masing-masing diulang sebanyak 15 kali. Melakukan 15 kali pengujian untuk setiap partisipan di setiap skenario memungkinkan kami mengumpulkan data kuantitatif yang kuat. Ini akan menunjukkan seberapa akurat sistem *Smart Dorm Key* kami dalam berbagai kondisi, sekaligus menjadi tolak ukur ketangguhan dan keandalan perangkat yang telah dikembangkan. Berikut adalah detail skenario pengujian yang akan dilakukan:

A. Pengujian *Voice Recognition*

Pada pengujian *voice recognition* ini dilakukan secara dua kali, dimana pada pengujian pertama dilakukan dengan menggunakan suara dari tiga orang sebagai subjek yang terdaftar dan tiga orang sebagai objek yang tidak terdaftar (*unknown*). Pengujian *voice recognition* ini dilakukan sebanyak 3 kali pengujian dimana pada setiap pengujian dilakukan pengambilan suara untuk dilakukan percobaan sebanyak 5 kali, sehingga untuk total percobaan dalam *voice recognition* ini mencapai 90 kali percobaan.

1. *Dataset* Keadaan Normal

Pengujian suara dalam keadaan normal merupakan bagian dari evaluasi sistem *voice recognition* yang dilakukan untuk mengukur akurasi pengenalan suara dalam kondisi lingkungan ideal.

Tabel 4.1 Hasil Pengujian Suara Terdaftar Keadaan Normal

	Pengujian ke-1		Pengujian ke-2		Pengujian ke-3	
	Berhasil	Gagal	Berhasil	Gagal	Berhasil	Gagal
Hazbi	✓		✓		✓	
		✓	✓		✓	
	✓		✓		✓	
	✓		✓			✓
	✓		✓		✓	
Trisucipto	✓		✓		✓	
	✓		✓		✓	
		✓	✓		✓	
	✓		✓		✓	
	✓		✓		✓	
Faiq	✓		✓		✓	
	✓		✓		✓	
	✓			✓	✓	
	✓		✓		✓	
	✓		✓		✓	
Hasil suara yang terverifikasi berhasil						41
Hasil suara yang gagal terverifikasi						4
Akurasi Keakuratan (%)						91%

Pada Tabel 4.1 setiap partisipan melakukan tiga kali pengujian, dengan masing-masing pengujian terdiri dari lima percobaan. Berdasarkan data, dari total 45 percobaan, sebanyak 41 suara berhasil diverifikasi, sementara 4 suara gagal diverifikasi. Hal ini menghasilkan akurasi keakuratan sistem sebesar 91% dalam kondisi suara dan lingkungan normal.

Tabel 4.2 Hasil Pengujian Suara Tidak Terdaftar Keadaan Normal

	Pengujian ke-1		Pengujian ke-2		Pengujian ke-3	
	Berhasil	Gagal	Berhasil	Gagal	Berhasil	Gagal
Ikratul	✓		✓		✓	
		✓	✓		✓	
	✓			✓		✓
	✓		✓		✓	
	✓		✓		✓	
Naufal	✓		✓		✓	
	✓		✓		✓	
	✓		✓		✓	
	✓		✓			✓
	✓		✓		✓	
Rayzi	✓		✓		✓	
	✓		✓		✓	
	✓			✓	✓	
	✓		✓		✓	
	✓		✓		✓	
Hasil suara yang terverifikasi unknown						40
Hasil suara yang tidak terverifikasi unknown						5
Akurasi Keakuratan (%)						88%

Berdasarkan data pada Tabel 4.2, dari total 45 percobaan, sebanyak 40 suara berhasil terverifikasi sebagai *unknown*, sementara 5 suara gagal diverifikasi sebagai *unknown*. Hal ini menghasilkan tingkat keakuratan sistem sebesar 88% dalam kondisi suara dan lingkungan normal.

2. *Dataset* Keadaan Berisik

Pengujian suara dalam keadaan berisik bertujuan untuk mengevaluasi akurasi sistem *voice recognition* saat dihadapkan pada gangguan suara dari lingkungan sekitar, seperti di lingkungan asrama yang ramai dengan percakapan atau musik.

Tabel 4.3 Hasil Pengujian Suara Normal Terdaftar dalam Keadaan Berisik

	Pengujian ke-1		Pengujian ke-2		Pengujian ke-3	
	Berhasil	Gagal	Berhasil	Gagal	Berhasil	Gagal
Hazbi		✓	✓			✓
	✓		✓		✓	
	✓			✓	✓	
		✓	✓		✓	
	✓		✓			✓
Trisucipto		✓	✓		✓	
	✓			✓		✓
	✓			✓	✓	
	✓		✓		✓	
		✓	✓		✓	
Faiq	✓			✓	✓	
	✓		✓		✓	
		✓	✓		✓	
		✓	✓			✓
	✓		✓		✓	
Hasil suara yang terverifikasi berhasil						31
Hasil suara yang gagal terverifikasi						14
Akurasi Keakuratan (%)						68%

Pada Tabel 4.3 dari total 45 percobaan, sebanyak 31 suara berhasil terverifikasi sebagai *unknown*, sementara 14 suara gagal diverifikasi. Hal ini menghasilkan akurasi sistem *voice recognition* sebesar 68% dalam kondisi berisik. Kesimpulannya, akurasi sistem *voice recognition* mengalami penurunan yang signifikan ketika dihadapkan pada kebisingan latar belakang, meskipun masih menunjukkan tingkat keberhasilan yang moderat.

Tabel 4.4 Hasil Pengujian Suara Tidak Terdaftar Keadaan Berisik

	Pengujian ke-1		Pengujian ke-2		Pengujian ke-3	
	Berhasil	Gagal	Berhasil	Gagal	Berhasil	Gagal
Ikratul			✓			✓
	✓		✓		✓	
	✓			✓	✓	
		✓	✓		✓	
	✓		✓			✓
Naufal		✓	✓		✓	
	✓			✓		✓
	✓			✓	✓	
	✓		✓		✓	
		✓	✓		✓	
Rayzi	✓			✓	✓	
	✓		✓		✓	
		✓	✓		✓	
		✓	✓			✓
	✓		✓		✓	
Hasil suara yang terverifikasi unknown						28
Hasil suara yang tidak terverifikasi unknown						17
Akurasi Keakuratan (%)						62%

Pada Tabel 4.4 dari total 45 percobaan, sebanyak 28 suara berhasil terverifikasi sebagai *unknown*, sementara 17 suara gagal terverifikasi sebagai *unknown*. Hal ini menghasilkan akurasi sistem *voice recognition* sebesar 62% dalam kondisi berisik.

3. *Dataset* Keadaan Serak

Pengujian suara dalam keadaan serak dirancang untuk mengevaluasi ketahanan dan akurasi sistem *voice recognition* ketika kualitas suara pengguna tidak optimal. Skenario ini mensimulasikan kondisi di mana pengguna mungkin sedang tidak sehat atau mengalami kelelahan suara, yang dapat mengubah karakteristik vokal.

Tabel 4.5 Hasil Pengujian ketika Suara Terdaftar dalam Keadaan Serak

	Pengujian ke-1		Pengujian ke-2		Pengujian ke-3	
	Berhasil	Gagal	Berhasil	Gagal	Berhasil	Gagal
Hazbi	✓		✓			✓
	✓		✓		✓	
		✓		✓	✓	
	✓			✓	✓	
	✓		✓		✓	
Trisucipto	✓		✓		✓	
		✓	✓		✓	
		✓	✓			✓
	✓		✓			✓
	✓			✓	✓	
Faiq		✓	✓		✓	
	✓		✓			✓
	✓		✓		✓	
	✓			✓	✓	
	✓		✓		✓	
Hasil suara yang terverifikasi berhasil						33
Hasil suara yang gagal terverifikasi						12
Akurasi Keakuratan (%)						73%

Pada Tabel 4.5 dari total 45 percobaan, sebanyak 33 suara berhasil diverifikasi, sementara 12 suara gagal diverifikasi. Hal ini menghasilkan akurasi sistem *voice recognition* sebesar 73% dalam kondisi suara serak. Kesimpulannya, meskipun suara dalam keadaan serak dapat menurunkan akurasi sistem *voice recognition*, sistem masih mampu mengidentifikasi sebagian besar suara dengan benar, menunjukkan tingkat adaptabilitas yang cukup baik terhadap variasi kualitas suara pengguna.

Tabel 4.6 Hasil Pengujian Suara Tidak Terdaftar Keadaan Serak

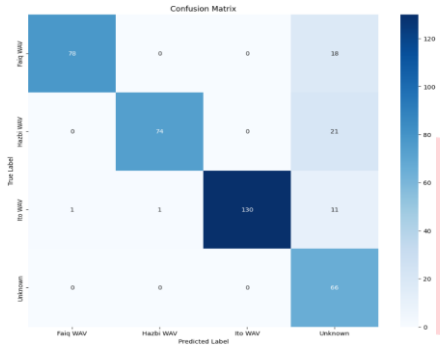
	Pengujian ke-1		Pengujian ke-2		Pengujian ke-3	
	Berhasil	Gagal	Berhasil	Gagal	Berhasil	Gagal
Ikratul	✓		✓			✓
		✓		✓	✓	
	✓			✓		✓
		✓	✓		✓	
	✓		✓		✓	
Naufal	✓			✓	✓	
	✓			✓	✓	
	✓		✓		✓	
	✓		✓		✓	
		✓	✓			✓
Rayzi	✓			✓	✓	
		✓	✓		✓	
		✓	✓		✓	
	✓		✓			✓
	✓		✓		✓	
Hasil suara yang terverifikasi unknown						30
Hasil suara yang tidak terverifikasi unknown						15
Akurasi Keakuratan (%)						66%

Pada Tabel 4.6 dari total 45 percobaan, sebanyak 30 berhasil terverifikasi sebagai *unknown*, sementara 15 suara gagal terverifikasi sebagai *unknown*. Hal ini menghasilkan akurasi sistem *voice recognition* sebesar 66% dalam kondisi suara serak.

B. Pengujian *Machine Learning*

Pengujian untuk *machine learning* ini dilakukan dengan mengumpulkan *dataset* suara dari Hazbi, Trisucipto, Faiq,

dan suara dari orang tidak dikenal sebanyak kurang lebih 300 suara dengan kondisi yang berbeda, seperti pengambilan suara ketika keadaan sedang berisik dan perubahan intonasi pada pengambilan *dataset*. Pengambilan *dataset* yang beragam ini diperlukan agar proses *training* untuk model dari *machine learning* dapat membaca dengan baik perbedaan suara dari setiap subjek. *Dataset* yang telah diperoleh akan *di training* oleh model *machine learning* yang digunakan dan dari hasil *training* tersebut akan dilakukan pengujian menggunakan *confusion matrix* dimana seperti yang terlihat pada gambar 4.1.



Gambar 4.1 Hasil Pengujian Menggunakan *Confusion Matrix Voice Recognition*

Dari gambar 4.1, dapat dilihat hasil *confusion matrix voice recognition* yang merupakan visualisasi dari training model *machine learning*, yaitu dengan membandingkan label sebenarnya yang berada pada posisi vertikal dengan label prediksi yang berada pada sumbu horizontal.

Tabel 4.7 Hasil Laporan Klasifikasi Setiap Label

Label	Precision	Recall	F1-Score
Faiq WAV	0.98	0.81	0.89
Hazbi WAV	0.98	0.78	0.87
Trisucipto WAV	0.90	1.00	0.95
Unknown	0.57	1.00	0.72

Pada Tabel 4.7, disajikan laporan klasifikasi yang merinci performa model melalui metrik *Precision*, *Recall*, dan *F1-Score* untuk setiap kelas suara. Hasil ini menunjukkan kinerja model yang memiliki keunggulan spesifik sekaligus area yang memerlukan perhatian. Nilai *Precision* yang sangat tinggi untuk kelas pengguna terdaftar, seperti 'Faiq WAV' (0.98) dan 'Hazbi WAV' (0.98), mengindikasikan bahwa ketika model membuat prediksi identitas, prediksi tersebut sangat akurat dan dapat dipercaya. Di sisi lain, nilai Recall yang sempurna (1.00) untuk kelas 'Trisucipto WAV' dan '*Unknown*' menunjukkan kemampuan superior model dalam dua aspek krusial: tidak pernah gagal mengenali pengguna 'Trisucipto' dan secara konsisten berhasil mengidentifikasi setiap suara asing yang sebenarnya.

Selanjutnya, laporan klasifikasi ini juga memperlihatkan potensi optimasi yang jelas pada model. Nilai *Recall* untuk 'Faiq WAV' (0.81) dan 'Hazbi WAV' (0.78) mengindikasikan bahwa model menerapkan standar kecocokan yang tinggi sebelum mengonfirmasi sebuah identitas, yang merupakan karakteristik dari sistem yang memprioritaskan keamanan. Sifat 'hati-hati' inilah yang membuat beberapa sampel vokal yang sedikit berbeda untuk sementara diklasifikasikan sebagai '*Unknown*', yang menjelaskan nilai *Precision* (0.57) pada kelas tersebut. Hal ini bukanlah sebuah kekurangan fundamental, melainkan sebuah indikasi bahwa dengan menambah variasi data latih untuk kedua pengguna, konsistensi pengenalan dapat dengan mudah disempurnakan tanpa perlu mengorbankan tingkat keamanan yang sudah sangat tinggi.

V. KESIMPULAN

Berdasarkan hasil perancangan, implementasi, dan serangkaian pengujian yang telah dilakukan pada proyek ini dapat ditarik beberapa simpulan sebagai berikut:

1. Performa verifikasi suara dengan sistem pengenalan suara menunjukkan akurasi tinggi, yaitu 91% untuk suara terdaftar dan 88% dalam mendeteksi suara tidak terdaftar (*unknown*). Namun, performanya menurun pada kondisi lingkungan yang lebih menantang: akurasi turun menjadi 68% (suara terdaftar) di lingkungan berisik dan 73% saat suara pengguna serak.

2. Kinerja model *machine learning* yang sangat andal menggunakan metode MFCC untuk ekstraksi fitur dan arsitektur CNN untuk klasifikasi suara menunjukkan kinerja yang sangat tinggi. Berdasarkan laporan klasifikasi, model mencapai nilai Precision yang cukup tinggi (0.98) untuk kelas 'Faiq WAV' dan 'Hazbi WAV', namun nilai Recall lebih rendah (0.81 dan 0.78) untuk kedua kelas tersebut, yang berbanding terbalik dengan kelas Ito WAV' dan Unknown' dimana nilai dari recall yang cukup tinggi yaitu (1.00) dan nilai precision (0.90) untuk ‘Ito WAV’ dan (0.57) untuk ‘Unknown’.

REFERENSI

[1] Telkom University, “Asrama.” Diakses: 11 Juli 2025. [Daring]. Tersedia pada: https://telkomuniversity.ac.id/asrama/

[2] R. Arunashmi, S. S. Gouda, dan K. Agrawal, “Speech Recognition Using MFCC & CNN,” *International Journal of Scientific Research and Engineering Development*, vol. 4.

[3] F. D. Adhinata, D. P. Rakhmadani, and A. J. T. Segara, "Pengenalan Jenis Kelamin Manusia Berbasis Suara Menggunakan MFCC dan GMM," *Jurnal Dinda*, vol. 1, no. 1, pp. 1-6, 2021.

[4] Y. LeCun, Y. Bengio, dan G. Hinton, “Deep Learning,” *Nature*, vol. 521, hlm. 436–444, Mei 2015, doi: 10.1038/nature14539.

[5] S. Swaminathan dan B. R. Tantri, “Confusion Matrix-Based Performance Evaluation Metrics,” *African Journal of Biomedical Research*, vol. 27, hlm. 4023–4031, Nov 2024, doi: 10.53555/AJBR.v27i4S.4345.

[6] D. A. Pramudhita, F. Azzahra, I. K. Arfat, R. Magdalena, dan S. Saidah, “Strawberry Plant Diseases Classification Using CNN Based on MobileNetV3-Large and EfficientNet-B0 Architecture,” *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 9, no. 3, hlm. 522–534, Jul 2023, doi: 10.26555/jiteki.v9i3.26341.

[7] D. Nagajyothi and P. Siddaiah, "Speech Recognition Using Convolutional Neural Networks," *International Journal of Engineering & Technology*, vol. 7, no. 4.6, pp. 133-137, 2018.
[8] V. H. Phung and E. J. Rhee, "A High-Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small Datasets," *Appl. Sci.*, vol. 9, no. 21, p. 4500, Oct. 2019.
[9] I. T. Nugraha, R. Patmasari, and A. I. Irawan, "Implementasi Membuka Kunci Pintu Otomatis Menggunakan Face Recognition pada Raspberry Pi Berbasis Internet of Thing," *e-Proceeding of Engineering*, vol. 7, no. 1, pp. 707-715, Apr. 2020.
[10] M. I. Yoren, R. Purnamasari, and E. Suhartono, "Penerapan Metode Histogram Oriented Of Gradients Dan Haar-Cascad Pada Pintu Asrama Pintar Telkom University," *e-Proceeding of Engineering*, vol. 11, no. 6, pp. 6487-6489, Dec. 2024.
[11] A. Y. Nasirudin, S. A. Wibowo, and R. Purnamasari, "Performance Analysis on Fine-tuned Region-based CNN for Object Recognition," in *Symposium of Future Telecommunication and Technologies (SOFTT)*, no. 2, Dec. 2018.