SKANIKA: Sistem Komputer dan Teknik Informatika
Volume 9, Nomor 1, Januari 2026, Halaman 1-12 E-ISSN: 2721-4788

Implementasi *Mel-Frequency Cepstral Coefficients* dan *Convolutional Neural Network* untuk Pengenalan Huruf Hiragana Berbasis Suara dalam Pembelajaran Bahasa Jepang

Muhammad Yusuf Ibrahim Ramadhani[1*], Saiful Nur Budiman[2], Udkhiati Mawaddah[3]
[1,2,3]Fakultas Teknik dan Informatika, Teknik Informatika, Universitas Islam Balitar, Blitar, Indonesia
E-mail: [1*]yusufibrahimramadhani@gmail.com, [2]sync.saifulnb@gmail.com,
[3]udkhiati.mawaddah@gmail.com
(* : corresponding author)

Abstrak

Bahasa Jepang semakin diminati di Indonesia, namun pembelajar sering mengalami kesulitan dalam menguasai huruf Hiragana karena jumlahnya yang cukup banyak serta adanya kemiripan bunyi antar huruf. Teknologi pengenalan suara dapat dimanfaatkan sebagai media pendukung pembelajaran, khususnya untuk melatih pengucapan dan meningkatkan pemahaman huruf Hiragana. Penelitian ini bertujuan mengembangkan sistem pengenalan suara huruf Hiragana berbasis *Mel-Frequency Cepstral Coefficients* (MFCC) dan *Convolutional Neural Network* (CNN). Dataset yang digunakan terdiri atas 46 huruf Hiragana, di mana setiap huruf direkam sebanyak 20 kali oleh empat narasumber, sehingga diperoleh total 3.680 data audio. Tahapan penelitian meliputi *preprocessing* sinyal audio, ekstraksi fitur MFCC, augmentasi data, pelatihan model CNN, serta evaluasi performa menggunakan metrik klasifikasi. Hasil pengujian menunjukkan bahwa model mampu mencapai akurasi sebesar 95% pada data uji, dengan sebagian besar huruf berhasil dikenali secara tepat. Kesalahan klasifikasi umumnya terjadi pada huruf-huruf yang memiliki kemiripan fonetik. Hasil ini menunjukkan bahwa pendekatan CNN berbasis MFCC efektif untuk pengenalan suara huruf Hiragana dan berpotensi diterapkan sebagai media pembelajaran digital interaktif dalam pembelajaran bahasa Jepang.

Kata kunci: CNN, Hiragana, MFCC, Pengenalan Suara

Abstract

Japanese language learning has gained increasing interest in Indonesia; however, learners often experience difficulties in mastering Hiragana characters due to their large number and phonetic similarities. Speech recognition technology can be utilized as a supportive learning medium, particularly for improving pronunciation and enhancing learners' understanding of Hiragana characters. This study aims to develop a Hiragana speech recognition system based on Mel-Frequency Cepstral Coefficients (MFCC) for feature extraction and Convolutional Neural Networks (CNN) for classification. The dataset consists of 46 Hiragana characters, with each character recorded 20 times by four speakers, resulting in a total of 3,680 audio samples. The research stages include audio signal preprocessing, MFCC feature extraction, data augmentation, CNN model training, and performance evaluation using classification metrics. Experimental results indicate that the proposed model achieves an accuracy of 95% on the test data, with most Hiragana characters being correctly recognized. Misclassifications mainly occur among characters with similar phonetic characteristics. These results demonstrate that the MFCC-based CNN approach is effective for Hiragana speech recognition and has potential to be applied as an interactive digital learning medium for Japanese language education.

***Keywords**: CNN, Hiragana, MFCC, Speech Recognition*

1. PENDAHULUAN

Bahasa Jepang merupakan salah satu bahasa asing yang perkembangannya cukup pesat di berbagai negara, termasuk di Indonesia. Hal ini ditunjukkan oleh survei yang dilakukan oleh *The Japan Foundation* yang berpusat di Tokyo [1]. Dalam sistem penulisan bahasa Jepang, Hiragana merupakan huruf dasar yang pertama kali harus dipelajari dan dikuasai oleh pembelajar karena digunakan secara luas dalam penulisan kata-kata sehari-hari [2]. Oleh karena itu, penguasaan Hiragana menjadi fondasi penting dalam pembelajaran bahasa Jepang dan tidak dapat diabaikan.

Meskipun demikian, pembelajaran Hiragana masih menghadapi berbagai tantangan. Jumlah huruf yang relatif banyak serta kemiripan bentuk dan bunyi antarhuruf menyebabkan pembelajar

sering mengalami kesulitan dalam membaca, mengucapkan, mengingat, maupun membedakan huruf-huruf tersebut [1]. Beberapa pendekatan pembelajaran telah dikembangkan, termasuk penggunaan media digital dan aplikasi pembelajaran bahasa. Namun, sebagian besar sistem yang ada masih berfokus pada aspek visual dan latihan membaca, sementara dukungan terhadap latihan pengucapan dan evaluasi suara pembelajar masih terbatas. Teknologi pengenalan suara (*speech recognition*), yang mampu mengenali dan mengubah ucapan manusia menjadi teks, berpotensi menjadi solusi untuk mendukung keterampilan berbicara dan mendengarkan dalam pembelajaran Hiragana [3]. Akan tetapi, penerapan *speech recognition* untuk pengenalan huruf Jepang masih menghadapi keterbatasan, terutama dalam menangani variasi pelafalan dan kemiripan fonetik antarhuruf.

Convolutional Neural Network (CNN) merupakan salah satu algoritma yang banyak digunakan dalam pengembangan sistem speech recognition karena kemampuannya mengolah data berstruktur grid, termasuk sinyal suara [4]. CNN unggul dalam menangani data berukuran besar dan kompleks serta mampu mengekstraksi fitur secara otomatis tanpa memerlukan proses ekstraksi manual yang rumit [5]. Dalam pengenalan suara, CNN umumnya dikombinasikan dengan fitur *Mel-Frequency Cepstral Coefficients* (MFCC) yang merepresentasikan karakteristik akustik suara secara efektif dan telah banyak digunakan untuk membedakan pola pengucapan [4]. Namun, pemanfaatan kombinasi MFCC dan CNN secara spesifik untuk pengenalan huruf Hiragana sebagai media pendukung pembelajaran bahasa Jepang masih relatif terbatas dan perlu dikaji lebih lanjut dari sisi performa dan aplikasinya.

Berdasarkan celah penelitian tersebut, penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem pengenalan suara huruf Hiragana menggunakan fitur MFCC dan metode CNN. Kontribusi utama penelitian ini terletak pada pengembangan model pengenalan suara yang mampu mengenali pengucapan huruf Hiragana secara akurat serta evaluasi performanya menggunakan metrik akurasi, presisi, *recall*, dan *F1-score*. Hasil penelitian diharapkan dapat menjadi dasar pengembangan media pembelajaran bahasa Jepang berbasis teknologi suara yang lebih interaktif dan efektif, khususnya dalam meningkatkan keterampilan berbicara dan mendengarkan pembelajar.

2. METODE PENELITIAN

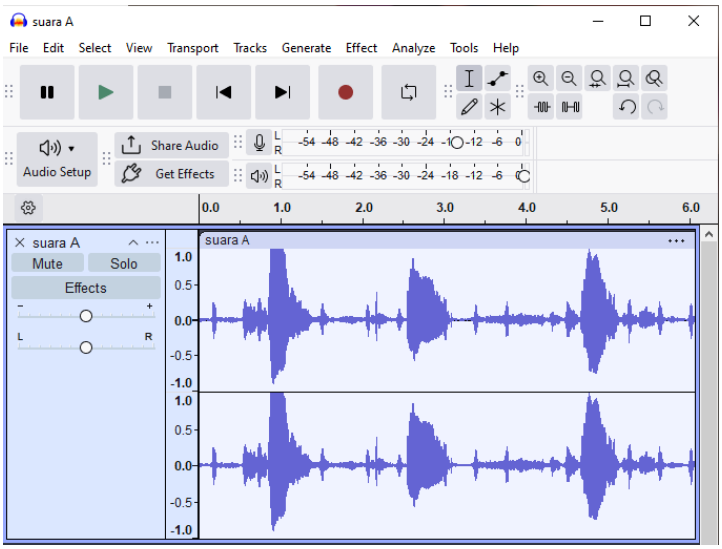
Penelitian ini termasuk ke dalam jenis penelitian terapan (*applied research*), karena berfokus pada pemanfaatan teori dan metode dalam bidang pengenalan suara serta kecerdasan buatan untuk menyelesaikan masalah praktis, yaitu mengenali suara huruf Hiragana. Tujuan dari penelitian ini bukan sekadar untuk menggali konsep-konsep teoretis, melainkan untuk mengembangkan sebuah sistem yang dapat diimplementasikan secara langsung dan memberi manfaat nyata, khususnya dalam mendukung proses pembelajaran bahasa Jepang. Terdapat 3 tahapan utama pada *speech recognition* yang dilakukan untuk mendapatkan hasil *text* dari masukan data suara. Tahapan tersebut meliputi *preprocessing*, *training*, dan *testing*. Pada tahap *preprocessing*, data suara dan label diproses menjadi array numerik, kemudian diambil nilai fitur MFCC dan nilai labelnya. Selanjutnya, pada tahap *training*, nilai fitur MFCC dan label digunakan untuk melatih model dengan metode *deep learning*, yaitu CNN. Pada tahap *testing*, model diuji dengan data uji pada tahap *testing* sistem untuk mengukur seberapa baik model mengenali suara Hiragana.

2.1. Dataset Suara Huruf Hiragana

Data yang diambil berupa rekaman suara untuk masing-masing huruf hiragana. Rekaman suara ini menjadi bahan dasar untuk melatih sistem dalam mengenali berbagai bunyi yang terkait dengan setiap huruf. Proses pengambilan data suara ini harus dilakukan dengan hati-hati, karena kualitas data sangat menentukan keberhasilan model dalam mengenali suara dengan akurat. Dataset terdiri atas 46 huruf Hiragana, sesuai dengan karakter dasar dalam sistem penulisan Jepang. Setiap huruf direkam sebanyak 20 kali, sehingga total jumlah rekaman yang diperoleh adalah 920 data audio per narasumber. Jumlah narasumber yang digunakan dalam proses

perekaman berjumlah 4 orang, sehingga total data rekaman suara keseluruhan adalah 3.680 dataset.

Seluruh rekaman disimpan dalam format .wav, yang dipilih karena memiliki kualitas suara tinggi dan tidak terkompresi, sehingga sesuai untuk kebutuhan ekstraksi fitur audio. Proses perekaman dilakukan dalam lingkungan yang terkontrol untuk meminimalkan gangguan suara latar (*background noise*), yang terlihat pada Gambar 1.



Gambar 1. Contoh Pemotongan File Suara Menggunakan Audacity

Dalam prosesnya, digunakan sebuah aplikasi tambahan yang akan mempermudah dalam proses pemotongan dan pelabelan data suara, sebelum melalui tahap ekstraksi fitur MFCC. Data suara akan dipotong menjadi ukuran 2 detik per huruf, agar tidak terlalu banyak ruang kosong didalam data suara, yang akan mempersulit sistem dalam melatih model.

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

MFCC adalah metode yang umum digunakan untuk mengubah suara menjadi representasi numerik yang dapat dipahami oleh komputer [5]. *Mel-Frequency Cepstral Coefficients* (MFCC) digunakan pada penelitian ini sebagai metode ekstraksi fitur untuk merepresentasikan sinyal suara huruf Hiragana ke dalam bentuk numerik yang dapat diproses oleh model *Convolutional Neural Network* (CNN). MFCC dipilih karena mampu menangkap karakteristik penting dari sinyal suara, khususnya pola spektral yang berkaitan dengan persepsi pendengaran manusia [6].

Pada tahap ekstraksi fitur, sinyal suara hasil perekaman diproses melalui beberapa langkah utama, yaitu framing, transformasi Fourier, pemetaan ke skala mel, dan transformasi kosinus diskrit (DCT). Proses ini menghasilkan koefisien MFCC yang merepresentasikan informasi timbre dan pola frekuensi suara secara ringkas. Dalam penelitian ini digunakan 13 koefisien MFCC, karena jumlah tersebut umum digunakan dan telah terbukti efektif dalam berbagai sistem pengenalan suara [7].

Setiap data suara dipotong dengan durasi yang sama dan diekstraksi menjadi matriks MFCC berukuran 13 × 87, sehingga memiliki dimensi yang konsisten untuk seluruh data. Untuk meningkatkan kestabilan pelatihan model, fitur MFCC kemudian dinormalisasi menggunakan *StandardScaler* yang dihitung dari data latih dan disimpan untuk digunakan kembali pada tahap pengujian dan prediksi suara dari mikrofon.

Penggunaan MFCC pada penelitian ini bertujuan untuk memastikan bahwa perbedaan antar huruf Hiragana dapat direpresentasikan secara optimal, sekaligus mengurangi pengaruh noise dan variasi intensitas suara. Dengan representasi fitur ini, model CNN dapat mempelajari pola akustik

tiap huruf secara lebih efektif, baik pada data uji maupun pada pengujian sistem menggunakan input suara langsung dari pengguna.

2.3. *Preprocessing*

Preprocessing adalah tahap di mana data dipersiapkan dan diolah agar sesuai dengan format yang diperlukan untuk diproses lebih lanjut[8]. *preprocessing* adalah langkah penting dalam mempersiapkan data sebelum dilakukan analisis lebih lanjut. Dalam bidang biologi sistem, *preprocessing* mencakup berbagai teknik untuk mengatasi data yang tidak lengkap, membersihkan *noise*, dan menangani nilai yang hilang. Proses ini melibatkan tahapan seperti pembersihan data, normalisasi, dan ekstraksi fitur MFCC[9]. Semua langkah tersebut dilakukan untuk memastikan bahwa data sudah dalam kondisi optimal dan siap digunakan untuk analisis berikutnya.

Tahap *preprocessing* dilakukan untuk mempersiapkan data suara sebelum masuk ke proses pelatihan dan pengujian model. Pada tahap awal, sistem membaca path data yang berisi file suara beserta labelnya, kemudian dilakukan penyelarasan antara ID label dengan ID file suara. File audio yang dimuat terlebih dahulu dibersihkan dari noise menggunakan pustaka Python, seperti *noisereduce*, agar kualitas sinyal lebih optimal. Selanjutnya, dilakukan proses augmentasi, misalnya *pitch shifting*, *time stretching*, dan penambahan *synthetic noise*, dengan tujuan memperbanyak variasi data dan meningkatkan ketahanan model terhadap perubahan input.

Setelah itu, dilakukan ekstraksi fitur menggunakan *Mel-Frequency Cepstral Coefficients* (MFCC), yang merepresentasikan karakteristik spektral suara sesuai cara manusia mendengarnya. Proses MFCC dimulai dari *pre-emphasis* untuk memperkuat frekuensi tinggi, dilanjutkan dengan *framing* dan *windowing* menggunakan *Hamming Window* agar transisi sinyal lebih halus. Kemudian, sinyal dipindahkan dari domain waktu ke domain frekuensi dengan *Fast Fourier Transform* (FFT), disaring dengan *Mel Filterbank*, dan diterapkan log energi agar lebih sesuai dengan persepsi manusia. Tahap akhir menggunakan *Discrete Cosine Transform* (DCT) untuk mereduksi informasi sehingga diperoleh 13 koefisien utama yang cukup representatif dalam membedakan pola suara.

Fitur MFCC yang dihasilkan kemudian dinormalisasi menggunakan *StandardScaler* agar berada pada skala seragam dan meminimalkan bias akibat perbedaan nilai ekstrem. Data yang sudah dinormalisasi selanjutnya di-*reshape* agar sesuai dengan format input CNN, sementara label dikonversi ke indeks numerik untuk keperluan klasifikasi. Seluruh hasil *preprocessing* disimpan dalam format *.h5* sehingga dapat digunakan secara konsisten dalam tahap pelatihan model. Dengan adanya tahapan ini, data suara diubah menjadi representasi numerik yang optimal, sehingga model CNN mampu mengenali pola suara Hiragana dengan lebih efektif dan akurat.

2.4. *Concolutional Neural Network* (CNN)

CNN adalah jenis jaringan syaraf tiruan yang dirancang khusus untuk mengolah data dengan struktur grid, seperti gambar atau suara[4]. CNN memiliki keunggulan utama yaitu kemampuannya dalam mengurangi dampak *noise* (gangguan), berbagi bobot di berbagai bagian input, serta mencegah model dari *overfitting* (memahami data dengan terlalu mendalam)[10].

CNN digunakan pada penelitian ini sebagai model klasifikasi untuk mengenali huruf Hiragana berdasarkan fitur MFCC yang diekstraksi dari sinyal suara. CNN dipilih karena kemampuannya dalam mempelajari pola spasial pada data dua dimensi, yang dalam penelitian ini direpresentasikan sebagai matriks MFCC. Arsitektur CNN yang digunakan terdiri dari beberapa lapisan konvolusi dengan fungsi aktivasi ReLU, diikuti oleh lapisan pooling untuk mereduksi dimensi fitur dan menangkap pola yang lebih abstrak [5]. Lapisan dropout diterapkan pada beberapa bagian jaringan untuk mengurangi risiko overfitting, terutama karena jumlah kelas yang relatif banyak dan variasi data suara yang tinggi. Pada bagian akhir, lapisan *fully connected* digunakan untuk melakukan klasifikasi ke dalam 46 kelas huruf Hiragana. Input ke dalam model berupa matriks MFCC berukuran 13×87 yang telah dinormalisasi dan ditambahkan dimensi kanal agar sesuai dengan format masukan CNN. Model dilatih menggunakan fungsi *loss*

categorical cross-entropy dan optimizer Adam, yang dipilih karena stabil dan efektif untuk permasalahan klasifikasi multi-kelas [11].

Penggunaan CNN pada penelitian ini memungkinkan model untuk secara otomatis mempelajari perbedaan pola akustik antar huruf Hiragana, termasuk perbedaan yang bersifat halus, sehingga dapat meningkatkan akurasi pengenalan baik pada data uji maupun pada pengujian sistem menggunakan suara langsung dari mikrofon [12].

2.5. *Training*

Tahap *training* pada penelitian ini merupakan proses pembelajaran model CNN untuk mengenali pola akustik huruf Hiragana berdasarkan fitur MFCC. Proses pelatihan dilakukan setelah seluruh data suara melalui tahap *preprocessing* dan ekstraksi fitur, sehingga data berada dalam kondisi yang siap digunakan oleh model. Proses ini sangat penting untuk memastikan model dapat berfungsi dengan baik dalam mengenali dan memahami data yang diberikan [13].

Fitur MFCC yang telah dinormalisasi digunakan sebagai input model, sedangkan label kelas merepresentasikan 46 huruf Hiragana. Dataset kemudian dibagi ke dalam subset *training* dan *validation* untuk mengevaluasi kemampuan generalisasi model selama pelatihan. Model dilatih menggunakan fungsi *loss categorical cross-entropy* dan *optimizer* Adam, yang dipilih karena stabilitasnya dalam menangani klasifikasi multi-kelas dan konvergensi yang efisien pada data audio berdimensi tinggi [14][15].

Arsitektur CNN yang digunakan terdiri dari 13 lapisan dengan total 262.574 parameter terlatih. Lapisan awal berupa *Conv2D* dengan 32 filter berukuran 3×3 berfungsi untuk mengekstraksi pola dasar dari matriks MFCC. Lapisan ini diikuti oleh *MaxPooling* untuk mereduksi dimensi fitur dan *Dropout* guna mengurangi risiko overfitting. Proses ekstraksi fitur dilanjutkan dengan dua lapisan konvolusi bertingkat masing-masing menggunakan 64 dan 128 filter, yang memungkinkan model mempelajari pola akustik yang lebih kompleks.

Setelah proses konvolusi, fitur diratakan menggunakan lapisan *Flatten* dan diproses oleh lapisan *Dense* dengan 128 neuron dan fungsi aktivasi ReLU. Lapisan *Dropout* dengan nilai 0.5 diterapkan pada tahap ini untuk meningkatkan kemampuan generalisasi model. Lapisan output menggunakan fungsi aktivasi *Softmax* dengan 46 unit, sesuai dengan jumlah kelas huruf Hiragana yang dikenali oleh sistem (Tabel 1).

Tabel 1. Arsitektur CNN

No.	Layer (Tipe)	Output Shape	Jumlah Parameter
1	Conv2D (32 filters, 3×3)	(None, 13, 87, 32)	320
2	MaxPooling2D (2×2)	(None, 6, 43, 32)	0
3	Dropout (rate=0.25)	(None, 6, 43, 32)	0
4	Conv2D (64 filters, 3×3)	(None, 6, 43, 64)	18,496
5	MaxPooling2D (2×2)	(None, 3, 21, 64)	0
6	Dropout (rate=0.25)	(None, 3, 21, 64)	0
7	Conv2D (128 filters, 3×3)	(None, 3, 21, 128)	73,856
8	MaxPooling2D (2×2)	(None, 1, 10, 128)	0
9	Dropout (rate=0.25)	(None, 1, 10, 128)	0
10	Flatten	(None, 1280)	0
11	Dense (ReLU, 128 units)	(None, 128)	163,968
12	Dropout (rate=0.5)	(None, 128)	0
13	Dense (Softmax, 46 units)	(None, 46)	5,934
	Total Parameter		**262,574**

Arsitektur yang dirancang memiliki tingkat kompleksitas yang seimbang antara kemampuan representasi fitur dan efisiensi komputasi. Penggunaan *Dropout* pada beberapa lapisan terbukti membantu mengurangi overfitting, sehingga model mampu memberikan performa yang stabil baik pada data validasi maupun saat diuji menggunakan suara langsung dari mikrofon. Dengan

konfigurasi tersebut, model CNN dinilai sesuai untuk tugas pengenalan suara huruf Hiragana berbasis fitur MFCC.

2.6. *Testing*

Testing adalah kegiatan yang dilakukan untuk mengevaluasi parameter atau kemampuan suatu program atau sistem, sekaligus memastikan apakah hasilnya sesuai dengan kebutuhan atau harapan[16]. Proses ini melibatkan pengujian untuk mengidentifikasi masalah dalam sistem sehingga dapat diperbaiki sebelum diluncurkan.

Pengujian model dilakukan menggunakan data uji (testing data) yang telah dipisahkan dari data pelatihan pada tahap sebelumnya. Data uji diproses melalui tahapan yang sama seperti data pelatihan, yaitu *preprocessing*, reduksi noise, ekstraksi fitur MFCC, serta normalisasi menggunakan *StandardScaler* yang sama. Selanjutnya, fitur MFCC hasil ekstraksi digunakan sebagai input ke model CNN untuk menghasilkan prediksi kelas huruf Hiragana.

Evaluasi performa model dilakukan dengan mengukur akurasi klasifikasi, baik pada data validasi selama pelatihan maupun pada data uji. Selain akurasi, pengujian juga memperhatikan hasil prediksi kelas untuk melihat kesalahan klasifikasi yang terjadi, khususnya pada huruf-huruf Hiragana yang memiliki kemiripan fonetik. Pendekatan ini umum digunakan dalam penelitian pengenalan suara untuk menganalisis kekuatan dan kelemahan model dalam membedakan pola akustik yang mirip.

Selain pengujian berbasis data uji, penelitian ini juga melakukan pengujian sistem secara langsung (real-time testing) menggunakan input suara dari mikrofon. Pada tahap ini, suara pengguna direkam, diekstraksi fitur MFCC-nya, kemudian diprediksi oleh model CNN yang telah dilatih. Hasil prediksi ditampilkan dalam bentuk huruf Hiragana dan romaji, sehingga memungkinkan evaluasi sistem secara end-to-end. Pengujian real-time ini bertujuan untuk menilai performa sistem dalam kondisi nyata yang mengandung variasi intonasi, kecepatan bicara, dan noise lingkungan.

2.7. Evaluasi Model

Evaluasi model merupakan salah satu tahapan penting dalam siklus pembelajaran mesin (*machine learning*) yang bertujuan untuk menilai sejauh mana model mampu bekerja pada data yang belum pernah ditemui sebelumnya [17]. Tahap ini dilakukan dengan menggunakan berbagai metrik, seperti akurasi, presisi, *recall*, dan *F1-score*, untuk mengukur efektivitas, keandalan, serta kemampuan generalisasi model. Melalui proses evaluasi, peneliti dapat mengetahui apakah model mengalami *overfitting*, *underfitting*, atau justru sudah sesuai dengan tujuan yang diharapkan. Dengan demikian, evaluasi tidak hanya berfungsi sebagai penilaian akhir, tetapi juga sebagai dasar untuk menentukan arah pengembangan model lebih lanjut.

Dalam konteks penelitian pengenalan suara huruf Hiragana, evaluasi sistem dilakukan untuk mengukur kemampuan model dalam mengenali bunyi huruf secara tepat. Akurasi digunakan untuk melihat perbandingan prediksi benar dengan keseluruhan data uji, presisi menilai ketepatan model dalam mengklasifikasi huruf yang benar, sedangkan recall menunjukkan kemampuan model dalam menangkap seluruh huruf yang sesuai. *F1-score*, yang merupakan gabungan presisi dan *recall*, memberikan gambaran seimbang tentang performa sistem pada tiap kelas. Hasil evaluasi ini menjadi acuan untuk menentukan apakah sistem telah bekerja dengan baik atau masih memerlukan perbaikan, misalnya melalui penyesuaian arsitektur CNN, penambahan data latih, maupun optimalisasi teknik ekstraksi fitur [18]. Dengan evaluasi yang menyeluruh, sistem pengenalan suara yang dikembangkan dapat dipastikan lebih efektif, reliabel, dan siap diterapkan dalam konteks nyata.

3. HASIL DAN PEMBAHASAN

3.1 Hasil *Preprocessing*

Tahap preprocessing diawali dengan ekstraksi fitur menggunakan metode MFCC melalui pustaka *librosa*. MFCC dipilih karena mampu merepresentasikan karakteristik spektral suara secara ringkas dan relevan dengan persepsi pendengaran manusia, sehingga banyak digunakan

dalam sistem pengenalan suara berbasis pembelajaran mendalam.

Dalam penelitian ini digunakan 13 koefisien MFCC, yang merupakan konfigurasi umum pada pengenalan suara karena koefisien tersebut telah terbukti cukup untuk merepresentasikan informasi spektral penting tanpa meningkatkan kompleksitas fitur secara berlebihan. Penggunaan 40 filter Mel bertujuan untuk menangkap distribusi energi spektrum secara lebih halus pada skala mel, sehingga perbedaan karakteristik fonetik antar huruf Hiragana dapat lebih jelas terwakili. Parameter *liftering* sebesar 22 diterapkan untuk menekankan koefisien MFCC awal yang mengandung informasi penting terkait bentuk spektrum, sedangkan *hop_length* sebesar 256 sampel (±11,6 ms pada *sample rate* 22.050 Hz) dipilih untuk memberikan resolusi temporal yang memadai dalam menggambarkan dinamika artikulasi suara. Kombinasi parameter ini selaras dengan konfigurasi yang umum digunakan pada penelitian pengenalan suara berbasis MFCC dan CNN.

Setelah proses ekstraksi, dilakukan *padding* pada hasil MFCC agar seluruh data memiliki dimensi yang seragam. Langkah ini diperlukan karena perbedaan durasi sinyal suara dapat menghasilkan jumlah frame MFCC yang berbeda, sementara arsitektur CNN mensyaratkan ukuran input yang konsisten. *Padding* dilakukan dengan menambahkan nilai nol pada frame yang kurang atau memotong frame yang berlebih, sehingga setiap sampel memiliki dimensi tetap sebesar **13 × 87**. Dengan demikian, seluruh data siap diproses pada tahap pelatihan dan pengujian model CNN.

Sebagai ilustrasi hasil ekstraksi fitur, Tabel 2 menampilkan rata-rata 13 koefisien MFCC dari salah satu sampel huruf Hiragana *'A'*.

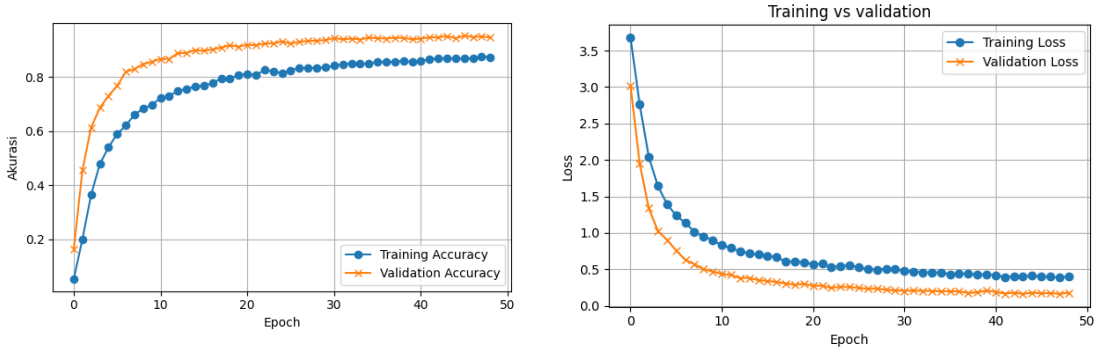
Tabel 2. Contoh Rata-Rata 13 Koefisien MFCC dari Hiragana 'A'

MFCC	Koefisien
1	-915,676
2	221,55
3	137,533
4	41,465
5	73,474
6	72,979
7	-34,014
8	96,624
9	134,396
10	64,058
11	33,18
12	-10,521
13	144,847

Berdasarkan Tabel 2, terlihat bahwa MFCC-1 memiliki nilai dominan yang merepresentasikan energi total sinyal suara, sedangkan koefisien menengah hingga tinggi (seperti MFCC-9 dan MFCC-13) mencerminkan karakteristik spektral dan formant yang berhubungan dengan artikulasi vokal. Pola ini berbeda antar huruf Hiragana, khususnya antara vokal dan konsonan, sehingga menghasilkan representasi fitur yang saling terpisah di ruang fitur. Perbedaan pola MFCC inilah yang kemudian dimanfaatkan oleh CNN untuk mempelajari ciri khas akustik setiap huruf dan meningkatkan kemampuan klasifikasi. Dengan demikian, proses preprocessing tidak hanya berfungsi sebagai tahap persiapan data, tetapi juga berperan penting dalam menentukan kualitas informasi yang diterima oleh model pada tahap pelatihan.

3.2 Hasil *Training*

Proses pelatihan model CNN dilakukan menggunakan fitur MFCC sebagai input selama 50 *epoch*. Dataset dibagi menjadi data latih dan data validasi dengan rasio 80:20, di mana data validasi digunakan untuk memantau kemampuan generalisasi model selama proses *training*. Pelatihan dilakukan menggunakan optimizer Adam dengan *learning rate* sebesar 0,001, serta fungsi *loss categorical cross-entropy* yang sesuai untuk klasifikasi multikelas.



Gambar 2 Grafik Hasil *Training* Akurasi dan L*oss* Pada Model

Gambar 2 menunjukkan grafik perkembangan akurasi dan loss pada data latih dan data validasi selama proses pelatihan. Pada awal epoch, akurasi data latih berada pada kisaran 10%, kemudian meningkat secara bertahap hingga mencapai sekitar 85% pada epoch ke-50. Sementara itu, akurasi data validasi mengalami peningkatan yang lebih cepat, melampaui akurasi data latih sejak sekitar epoch ke-5, dan mencapai nilai sekitar 92–93% pada epoch ke-30, sebelum akhirnya stabil hingga akhir pelatihan.

Perbedaan akurasi antara data latih dan data validasi yang relatif kecil, yaitu sekitar 7–8%, menunjukkan bahwa model memiliki kemampuan generalisasi yang baik dan tidak mengalami overfitting yang signifikan. Hal ini juga diperkuat oleh pola kurva loss, di mana nilai loss pada data latih dan validasi sama-sama mengalami penurunan tajam pada fase awal pelatihan, kemudian menurun secara lebih gradual hingga mencapai kondisi konvergen pada rentang epoch ke-40 hingga ke-50. Selisih nilai loss antara data latih dan validasi yang kecil serta tidak adanya fluktuasi ekstrem pada kurva validasi menandakan bahwa proses optimisasi berjalan secara stabil. Penggunaan optimizer Adam dengan learning rate 0,001 terbukti efektif dalam mempercepat konvergensi model sekaligus menjaga stabilitas pelatihan. Selain itu, penerapan Dropout pada beberapa lapisan CNN turut berkontribusi dalam mencegah overfitting dan meningkatkan kemampuan generalisasi model terhadap data yang belum pernah dilihat sebelumnya.

3.3 Hasil *Testing*

Tahap pengujian dilakukan untuk mengevaluasi performa model CNN dalam mengenali huruf Hiragana dari data uji yang tidak digunakan selama proses pelatihan. Pengujian dilakukan secara *end-to-end*, dimulai dari perekaman suara secara *real-time*, ekstraksi fitur MFCC, normalisasi menggunakan *StandardScaler*, hingga proses klasifikasi menggunakan model CNN terlatih dengan fungsi aktivasi *Softmax* pada lapisan output.

Pada setiap pengujian, sistem menghasilkan probabilitas untuk masing-masing dari 46 kelas huruf Hiragana, dan prediksi akhir ditentukan berdasarkan nilai probabilitas tertinggi (*argmax*). Sebagai contoh, pada pengujian dengan input suara huruf *"sa"*, sistem berhasil mengklasifikasikan huruf tersebut dengan probabilitas 99,99%, sementara probabilitas kelas lain berada mendekati nol. Hasil ini menunjukkan bahwa model mampu mengenali pola suara dengan sangat baik ketika karakteristik akustik input sesuai dengan representasi data latih.

Selain prediksi yang sepenuhnya benar, ditemukan pula kasus prediksi dengan probabilitas yang tersebar pada beberapa kelas yang memiliki kemiripan fonetik. Sebagai contoh, pada input suara *"i"*, model menghasilkan probabilitas tertinggi pada huruf *"ki"* sebesar 47,51%, diikuti oleh *"hi"* sebesar 43,01%, dan *"i"* sebesar 6,68%. Pola distribusi probabilitas ini menunjukkan bahwa model masih mampu menangkap kemiripan akustik antar huruf, meskipun prediksi akhir tidak sesuai dengan label yang diharapkan. Fenomena ini mencerminkan sensitivitas model terhadap kesamaan ciri fonetik antar suku kata Hiragana.

Di sisi lain, terdapat pula kasus prediksi yang sepenuhnya keliru, misalnya pada input *"i"* yang diprediksi sebagai *"ku"* dengan probabilitas sangat tinggi. Kesalahan semacam ini menunjukkan bahwa model masih rentan terhadap variasi intonasi, artikulasi pengguna, maupun

pengaruh noise lingkungan. Selain itu, keterbatasan variasi data latih juga berpotensi menyebabkan model belum sepenuhnya mampu merepresentasikan seluruh kemungkinan variasi pengucapan.

Hasil pengujian menunjukkan bahwa model CNN berbasis fitur MFCC memiliki kemampuan klasifikasi yang baik, ditunjukkan oleh probabilitas tinggi pada prediksi benar dan distribusi probabilitas yang logis pada kasus huruf yang memiliki kemiripan fonetik. Namun demikian, untuk meningkatkan ketahanan model terhadap variasi input suara, diperlukan pengembangan lebih lanjut seperti penambahan variasi data latih, augmentasi yang lebih beragam, serta penyempurnaan tahap preprocessing.

3.4 Evaluasi Model

Evaluasi model dilakukan untuk mengukur kinerja sistem pengenalan suara huruf Hiragana secara kuantitatif. Evaluasi mencakup metrik akurasi dan loss selama pelatihan, serta metrik performa klasifikasi berupa precision, recall, dan F1-score pada data pengujian. Selain itu, evaluasi juga mempertimbangkan distribusi kesalahan prediksi untuk memahami keterbatasan model secara lebih mendalam.

Tabel 3. Evaluasi Model Keseluruhan

Metrik	Nilai
Akurasi	0.951
Total Data	2946
Prediksi Benar	2802
Prediksi Salah	144

Berdasarkan hasil evaluasi keseluruhan yang ditunjukkan pada Tabel 3, model mencapai tingkat akurasi sebesar 95,1% pada data pengujian. Dari total 2.946 sampel data uji, sebanyak 2.802 berhasil diklasifikasikan dengan benar, sementara 144 data mengalami kesalahan klasifikasi. Nilai ini menunjukkan bahwa model CNN berbasis fitur MFCC mampu mengenali pola akustik huruf Hiragana secara konsisten dengan tingkat kesalahan yang relatif rendah.

Tabel 4. Tabel Performa Klasifikasi Kuruf Hiragana

Huruf	Precision	Recall	F1-Score	Support
a	0.95	0.82	0.88	72
i	0.92	0.96	0.94	73
u	0.98	1.00	0.99	58
e	0.95	0.85	0.90	61
o	0.97	0.89	0.93	70
ka	0.90	0.95	0.92	57
ki	0.95	0.87	0.91	70
ku	0.88	0.94	0.91	71
ke	0.81	0.95	0.88	60
ko	0.93	0.94	0.94	69
sa	0.97	1.00	0.99	71
shi	0.93	0.98	0.96	65
su	0.94	0.96	0.95	71
se	0.97	0.97	0.97	65
so	0.97	0.97	0.97	75
ta	0.97	0.96	0.96	70
chi	0.98	0.95	0.97	65
tsu	0.98	0.94	0.96	65
te	0.95	0.95	0.95	65
to	0.87	0.98	0.92	49
na	1.00	1.00	1.00	57
ni	0.88	0.92	0.90	63
nu	0.92	0.97	0.94	61
ne	0.95	0.97	0.96	64
no	1.00	0.96	0.98	54
ha	1.00	0.99	0.99	76
hi	0.95	0.90	0.92	61

Huruf	Precision	Recall	F1-Score	Support
fu	0.97	0.99	0.98	73
he	0.90	0.95	0.93	60
ho	1.00	0.95	0.97	77
ma	1.00	1.00	1.00	68
mi	0.92	0.92	0.92	66
mu	0.95	0.88	0.91	72
me	0.95	0.98	0.96	56
mo	0.96	0.94	0.95	69
ya	1.00	1.00	1.00	54
yu	0.94	1.00	0.97	64
yo	1.00	0.95	0.97	77
ra	1.00	0.97	0.98	63
ri	0.98	0.98	0.98	45
ru	0.89	0.93	0.91	55
re	0.93	0.96	0.95	56
ro	0.93	0.93	0.93	54
wa	1.00	0.97	0.98	61
wo	0.93	0.98	0.96	56
n	1.00	0.98	0.99	62

Evaluasi lebih rinci terhadap masing-masing kelas huruf Hiragana disajikan pada Tabel 4 dalam bentuk nilai precision, recall, dan F1-score. Secara umum, hampir seluruh kelas memperoleh nilai F1-score di atas 0,90, bahkan sebagian besar berada di atas 0,95, yang menandakan keseimbangan yang baik antara ketepatan prediksi (precision) dan kemampuan model dalam mengenali seluruh sampel kelas yang benar (recall).

Namun demikian, beberapa huruf seperti ke, to, dan ru menunjukkan nilai recall yang relatif lebih rendah dibandingkan huruf lainnya. Kondisi ini dapat dijelaskan dari sisi karakteristik fonetik dan artikulasi. Huruf *"ke"* memiliki pola konsonan-vokal yang mirip dengan *"te"* dan *"he"*, terutama pada bagian awal artikulasi konsonan yang relatif singkat, sehingga menghasilkan spektrum MFCC yang saling tumpang tindih. Hal serupa juga terjadi pada huruf *"to"*, yang secara fonetik dekat dengan *"ko"* dan *"do"* (dalam konteks artikulasi plosif alveolar dan velar), sehingga meningkatkan kemungkinan kesalahan klasifikasi.

Sementara itu, huruf *"ru"* memiliki karakteristik transisi konsonan yang lebih lemah dan durasi artikulasi yang pendek, sehingga informasi akustik yang terekam pada beberapa koefisien MFCC menjadi kurang dominan. Akibatnya, model cenderung mengalami kesulitan dalam membedakan *"ru"* dari huruf lain yang memiliki pola vokal serupa, seperti *"ri"* atau *"ro"*. Faktor ini diperkuat apabila variasi intonasi atau kecepatan pengucapan pengguna berbeda dari data latih.

Analisis ini menunjukkan bahwa kesalahan klasifikasi tidak semata-mata disebabkan oleh kelemahan model, tetapi juga dipengaruhi oleh kemiripan fonetik antar huruf Hiragana serta keterbatasan representasi variasi pengucapan dalam data latih. Oleh karena itu, peningkatan performa pada kelas-kelas tersebut dapat dicapai melalui penambahan variasi data latih, khususnya pada huruf yang memiliki kemiripan artikulasi, serta penerapan teknik augmentasi yang lebih beragam untuk memperkaya representasi fitur akustik.

4. KESIMPULAN DAN SARAN

Penelitian ini menunjukkan bahwa pendekatan CNN berbasis fitur MFCC efektif digunakan untuk pengenalan suara huruf Hiragana dalam skenario klasifikasi multikelas dengan jumlah kelas yang relatif besar (46 kelas). Model yang dikembangkan mampu mencapai akurasi pengujian sebesar 95,1%, yang mengindikasikan kemampuan generalisasi yang baik terhadap data uji. Hasil ini menegaskan bahwa representasi MFCC mampu menangkap karakteristik akustik huruf Hiragana secara konsisten dan dapat dimanfaatkan sebagai dasar sistem pembelajaran bahasa Jepang berbasis suara.

Kontribusi utama penelitian ini terletak pada penerapan CNN untuk pengenalan huruf Hiragana sebagai satuan fonetik dasar, yang masih relatif terbatas dikaji dalam konteks

pembelajaran bahasa Jepang, khususnya pada lingkungan pengguna berbahasa Indonesia. Evaluasi per kelas menunjukkan bahwa sebagian besar huruf memperoleh nilai precision, recall, dan F1-score yang tinggi. Namun demikian, beberapa huruf dengan kemiripan fonetik dan artikulasi, seperti *ke*, *to*, dan *ru*, menunjukkan nilai recall yang lebih rendah, yang mengindikasikan adanya tumpang tindih pola akustik yang belum sepenuhnya dapat dibedakan oleh model.

Meskipun sistem telah diuji menggunakan input suara secara real-time melalui mikrofon dan menunjukkan performa yang baik, pengujian tersebut masih dilakukan dalam lingkungan yang terbatas dan terkontrol. Oleh karena itu, hasil uji real-time belum sepenuhnya merepresentasikan kondisi dunia nyata yang kompleks, seperti variasi aksen, intonasi, serta gangguan suara lingkungan.

Sebagai pengembangan lanjutan, penelitian selanjutnya disarankan untuk memperluas dataset dengan melibatkan variasi penutur dan kondisi perekaman yang lebih beragam guna meningkatkan ketahanan model terhadap variasi pengucapan dan noise. Selain itu, eksplorasi arsitektur yang lebih sensitif terhadap sifat sekuensial sinyal audio, seperti LSTM, GRU, atau pendekatan hibrida CNN–RNN, berpotensi meningkatkan kemampuan model dalam membedakan huruf dengan pola bunyi yang saling menyerupai. Integrasi sistem ke dalam aplikasi pembelajaran interaktif serta penerapan teknik penanganan noise secara adaptif juga dapat meningkatkan nilai praktis dan kesiapan sistem untuk digunakan dalam skenario dunia nyata.

UCAPAN TERIMA KASIH

Penulis menyampaikan terima kasih yang sebesar-besarnya kepada dosen pembimbing yang telah memberikan arahan, masukan, dan bimbingan selama proses penelitian ini berlangsung. Ucapan terima kasih juga ditujukan kepada pihak kampus yang telah menyediakan fasilitas serta kepada keluarga dan rekan-rekan yang senantiasa memberikan dukungan moral maupun motivasi. Tanpa bantuan dan dukungan dari berbagai pihak, penelitian ini tidak akan dapat terselesaikan dengan baik.

DAFTAR PUSTAKA

[1] T. O. E. Mulyana, "Faktor Kesulitan Belajar Menulis Huruf Hiragana Pada Siswa Kelas X Sma Labschool Surabaya Tahun Ajaran 2019/2020," *Hikari*, vol. 1, no. 4, pp. 61–67, 2020, [Online]. Available: https://ejournal.unesa.ac.id/index.php/kejepangan-unesa/article/view/33865

[2] B. P. Zhelita and R. Arni, "Efektivitas Media Puzzle Terhadap Penguasaan Hiragana Siswa SMA," *Omi. J. Bhs. dan Pembelajaran Bhs. Jepang*, vol. 6, no. 2, pp. 242–255, 2023, doi: 10.24036/omg.v6i2.725.

[3] I. K. S. Buana, "Implementasi Aplikasi Speech to Text untuk Memudahkan Wartawan Mencatat Wawancara dengan Python," *J. Sist. dan Inform.*, vol. 14, no. 2, pp. 135–142, 2020, doi: 10.30864/jsi.v14i2.293.

[4] D. C. Khrisne and T. Hendrawati, "Indonesian Alphabet Speech Recognition forEarly Literacy using Convolutional NeuralNetwork Approach," *J. Electr. Electron. Informatics*, vol. 4, no. 1, pp. 34–37, 2020, doi: 10.17509/ijal.v9i3.23223.

[5] S. Dwijayanti, A. Y. Putri, and B. Y. Suprapto, "Speaker Identification Using a Convolutional Neural Network," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 6, no. 1, pp. 140–145, 2022, doi: 10.29207/resti.v6i1.3795.

[6] M. Musaev, I. Khujayorov, and M. Ochilov, "Image Approach to Speech Recognition on CNN," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, 2019. doi: 10.1145/3386164.3389100.

[7] W. Mustikarini, R. Hidayat, and A. Bejo, "Real-Time Indonesian Language Speech Recognition with MFCC Algorithms and Python-Based SVM," *IJITEE*, vol. 3, no. 2, pp. 55–60, 2019, doi: 10.22146/ijitee.49426.

[8] S. Shevira, I. Made, A. D. Suarjaya, and P. Wira Buana, "Pengaruh Kombinasi dan Urutan

Pre-Processing pada Tweets Bahasa Indonesia," *JITTER-Jurnal Ilm. Teknol. dan Komput.*, vol. 3, no. 2, 2022, doi: 10.24843/JTRTI.2022.v03.i02.p06.

[9] S. Roy, P. Sharma, K. Nath, D. K. Bhattacharyya, and J. K. Kalita, "Pre-processing: A data preparation step," *Encycl. Bioinforma. Comput. Biol. ABC Bioinforma.*, vol. 1–3, pp. 463–471, Jan. 2018, doi: 10.1016/B978-0-12-809633-8.20457-3.

[10] A. Alsobhani, H. M. A. Alabboodi, and H. Mahdi, "Speech Recognition using Convolution Deep Neural Networks," *J. Phys. Conf. Ser.*, vol. 1973, no. 1, 2021, doi: 10.1088/1742-6596/1973/1/012166.

[11] H. Rafliansyah, B. Rahmat, and C. A. Putra, "Klasifikasi Suara Instrumen Musik Tiup Menggunakan Metode Convolutional Neural Network," *Merkurius J. Ris. Sist. Inf. dan Tek. Inform.*, vol. 2, no. 4, pp. 01–09, 2024, doi: 10.61132/merkurius.v2i4.119.

[12] U. Mawaddah, H. Armanto, and E. Setyati, "Prediksi Karakteristik Personal Menggunakan Analisis Tanda Tangan Dengan Mengggunakan Metode Convolutional Neural Network (Cnn)," *Antivirus J. Ilm. Tek. Inform.*, vol. 15, no. 1, pp. 123–133, 2021, doi: 10.35457/antivirus.v15i1.1526.

[13] S. Dua *et al.*, "Developing a Speech Recognition System for Recognizing Tonal Speech Signals Using a Convolutional Neural Network," *Appl. Sci.*, vol. 12, no. 12, 2022, doi: 10.3390/app12126223.

[14] S. Yusdiantoro and T. B. Sasongko, "Implementasi Algoritma MFCC dan CNN dalam Klasifikasi Makna Tangisan Bayi," *Indones. J. Comput. Sci.*, vol. 12, no. 1, pp. 1957–1968, 2023, doi: 10.33022/ijcs.v12i4.3243

[15] N. Asanah and I. Pratama, "Deep Learning Approach for Music Genre Classification using Multi - Feature Audio Representations," *Sist. J. Sist. Inf.*, vol. 14, pp. 2045–2054, 2025, doi: 10.32520/stmsi.v14i5.5369.

[16] I. Zulhaedi, "Kenapa Testing itu Penting?," School of Information Systems. Accessed: Jan. 21, 2025. [Online]. Available: https://sis.binus.ac.id/2023/11/08/kenapa-testing-itu-penting/

[17] O. Colliot, *A Non-technical Introduction to Machine Learning*, vol. 197. 2023. doi: 10.1007/978-1-0716-3195-9_1.

[18] V. J. Varma *et al.*, "Enhancing dysarthria severity classification: efficient audio based deep learning models," *Discov. Appl. Sci.*, vol. 7, no. 8, 2025, doi: 10.1007/s42452-025-07260-2.