JITSI Vol 1 No. 1 (2026)
Journal of Information Technology and System Integration
https://journal2.ummat.ac.id/index.php/jitsi

ANALISIS PENGARUH NOISE PADA SPEECH RECOGNITION MENGGUNAKAN MFCC DAN SVM

ANALYSIS THE EFFECT OF NOISE ON SPEECH RECOGNITION USING MFCC AND SVM

Joelianto Darmawan[1*], Nur Aulia Putri[2], Lalu Muhammad Gibran[3], Muhammad Ibrahim[4]
[1,2,3,4]Sistem dan Teknologi Informasi, Universitas Muhammadiyah Mataram
email: darmawanjoe313@gmail.com[1*]
email: nuraulia.putri150405@gmail.com[2]
email: lalugibran879@gmail.com[3]
email: muhammadibrahimvrca18@gmail.com[4]

Abstrak: *Speech recognition* merupakan teknologi yang banyak digunakan pada sistem berbasis suara, namun performanya dapat menurun akibat keberadaan noise pada sinyal audio. Penelitian ini bertujuan menganalisis pengaruh noise terhadap kinerja sistem speech recognition menggunakan ekstraksi fitur *Mel Frequency Cepstral Coefficients* (MFCC) dan metode klasifikasi *Support Vector Machine* (SVM). Dataset yang digunakan adalah *Speech Commands* yang terdiri atas 20.000 data audio dari delapan kelas perintah suara. Untuk menguji ketahanan model, dilakukan penambahan tiga jenis *noise*, yaitu *Pink Noise*, *Running Tap Noise*, dan *White Noise* sehingga total data menjadi 80.000 sampel. Ekstraksi fitur dilakukan menggunakan MFCC, Delta MFCC, dan Delta-Delta MFCC yang direpresentasikan menjadi vektor fitur berdimensi 39. Proses klasifikasi menggunakan SVM dengan kernel *Radial Basis Function* (RBF), parameter C = 10, dan gamma = scale. Evaluasi dilakukan menggunakan *Accuracy, Precision, Recall, Weighted F1-Score, Confusion Matrix,* dan *Character Error Rate* (CER). Hasil penelitian menunjukkan bahwa kondisi audio normal menghasilkan akurasi tertinggi sebesar 97,35%, sedangkan *White Noise* menghasilkan akurasi terendah sebesar 72,95%. Hasil penelitian menunjukkan bahwa *noise* berpengaruh terhadap performa *speech recognition*, meskipun kombinasi MFCC dan SVM masih mampu mempertahankan kinerja yang baik pada seluruh skenario pengujian.
Kata Kunci: MFCC*; Noise; Speech Recognition; Support Vector Machine; Character Error Rate.*

Abstract: *Speech recognition is widely used in voice-based systems; however, its performance can decrease due to noise in audio signals. This study aims to analyze the effect of noise on speech recognition performance using Mel Frequency Cepstral Coefficients (MFCC) and Support Vector Machine (SVM). The Speech Commands dataset consisting of 20,000 audio samples from eight command classes was used in this study. Three types of noise, namely Pink Noise, Running Tap Noise, and White Noise, were added to the dataset, resulting in a total of 80,000 audio samples. Feature extraction was performed using MFCC, Delta MFCC, and Delta-Delta MFCC, which were represented as a 39-dimensional feature vector. Classification was conducted using an SVM with a Radial Basis Function (RBF) kernel, C = 10, and gamma = scale. Performance was evaluated using Accuracy, Precision, Recall, Weighted F1-Score, Confusion Matrix, and Character Error Rate (CER). The results showed that clean audio achieved the highest accuracy of 97.35%, while White Noise produced the lowest accuracy of 72.95%. These findings indicate that noise significantly affects speech recognition performance, although the MFCC-SVM combination remains effective across all testing scenarios*.
Keywords: MFCC*, Noise, Speech Recognition, Support Vector Machine, Character Error Rate.*

PENDAHULUAN

Perkembangan teknologi pengenalan ucapan (*speech recognition*) mengalami peningkatan yang signifikan seiring dengan kemajuan kecerdasan buatan dan pengolahan sinyal digital [1]. Teknologi ini telah diterapkan pada berbagai bidang, seperti asisten virtual, sistem navigasi, perangkat rumah pintar, hingga sistem otomatisasi industri [2]. Kemampuan sistem dalam mengenali dan menginterpretasikan perintah suara menjadi faktor penting dalam meningkatkan interaksi antara manusia dan komputer [3], [4]. Peningkatan kebutuhan terhadap sistem yang responsif dan efisien mendorong penelitian mengenai metode pengenalan ucapan yang mampu menghasilkan tingkat akurasi tinggi pada berbagai kondisi lingkungan [5]. Oleh karena itu, pengembangan sistem *speech recognition* yang stabil dan adaptif masih menjadi topik penelitian yang relevan dalam bidang pengolahan sinyal audio dan *machine learning* [6].

Meskipun teknologi *speech recognition* terus berkembang, performa sistem masih dipengaruhi oleh kualitas sinyal audio yang digunakan [7]. Salah satu permasalahan utama dalam pengenalan ucapan adalah keberadaan *noise* atau gangguan suara pada proses perekaman audio. *Noise* dapat berasal dari lingkungan sekitar, seperti suara kendaraan, percakapan, angin, maupun gangguan elektronik pada perangkat perekam [8]. Keberadaan *noise* menyebabkan perubahan karakteristik sinyal suara sehingga informasi penting pada audio menjadi sulit dikenali oleh sistem klasifikasi [9]. Kondisi tersebut berdampak pada penurunan tingkat akurasi model, terutama ketika sistem diterapkan pada lingkungan nyata yang memiliki kondisi akustik berbeda-beda [10]. Oleh sebab itu, analisis pengaruh *noise* terhadap performa *speech recognition* menjadi aspek penting untuk menghasilkan sistem yang lebih andal [11].

Salah satu metode ekstraksi fitur yang banyak digunakan dalam penelitian pengenalan ucapan adalah *Mel Frequency Cepstral Coefficients* (MFCC) [12]. Metode MFCC mampu merepresentasikan karakteristik suara berdasarkan

persepsi pendengaran manusia melalui pendekatan skala mel [13]. Proses ekstraksi fitur pada MFCC dilakukan melalui beberapa tahapan, yaitu *framing*, *windowing*, transformasi *Fourier*, *mel filterbank*, dan *discrete cosine transform* (DCT) [14]. Hasil ekstraksi MFCC menghasilkan representasi numerik yang lebih ringkas dan informatif sehingga mempermudah proses klasifikasi suara. Selain memiliki kompleksitas komputasi yang relatif rendah, MFCC juga dikenal efektif dalam mempertahankan informasi penting pada sinyal audio [6], [15]. Oleh karena itu, metode ini masih banyak digunakan dalam penelitian *speech recognition*, khususnya pada sistem pengenalan perintah suara [16].

Setelah proses ekstraksi fitur dilakukan, tahapan berikutnya adalah klasifikasi data suara menggunakan metode *machine learning*. Salah satu metode klasifikasi yang sering digunakan pada pengenalan ucapan adalah *Support Vector Machine* (SVM). Metode SVM memiliki kemampuan yang baik dalam memisahkan data berdasarkan *hyperplane* optimal sehingga efektif digunakan pada permasalahan klasifikasi multikelas [17]. Selain itu, SVM mampu bekerja dengan baik pada jumlah data yang besar dan dimensi fitur yang tinggi, termasuk fitur hasil ekstraksi MFCC [15], [18]. Penggunaan kernel pada SVM juga memungkinkan proses klasifikasi dilakukan secara lebih fleksibel terhadap pola data nonlinier [19]. Beberapa penelitian menunjukkan bahwa kombinasi MFCC dan SVM mampu menghasilkan performa klasifikasi yang baik pada sistem pengenalan ucapan dengan tingkat kompleksitas komputasi yang relatif stabil [16].

Berbagai penelitian terkait *speech recognition* telah dilakukan menggunakan kombinasi MFCC dan SVM dengan hasil akurasi yang cukup tinggi pada kondisi audio normal [15], [16], [20]. Namun, sebagian besar penelitian masih berfokus pada pengujian tanpa mempertimbangkan pengaruh *noise* terhadap stabilitas performa model [7], [21]. Selain itu, penelitian yang membandingkan pengaruh berbagai jenis *noise* terhadap performa *speech recognition* masih relatif terbatas [22]. Analisis mengenai perubahan akurasi model SVM pada kondisi audio yang memiliki *noise* juga belum banyak dibahas secara mendalam, khususnya pada dataset berukuran besar [23]. Selain itu, beberapa penelitian sebelumnya masih menggunakan dataset dengan jumlah data yang relatif terbatas dan berfokus pada kondisi eksperimen tertentu, sehingga evaluasi performa model pada variasi kondisi audio yang lebih luas masih perlu diteliti lebih lanjut [8], [24]. Berdasarkan kondisi tersebut, diperlukan penelitian yang mampu menganalisis pengaruh *noise* terhadap performa *speech recognition* secara lebih komprehensif.

Penelitian ini bertujuan untuk menganalisis pengaruh *noise* terhadap tingkat akurasi sistem *speech recognition* menggunakan metode ekstraksi fitur MFCC dan identifikasi *Support Vector Machine* (SVM). Dataset yang digunakan terdiri dari 20.000 data suara yang terbagi ke dalam delapan kelas perintah, yaitu *yes*, *no*, *up*, *down*, *left*, *right*, *stop*, dan *go*. Penelitian dilakukan melalui beberapa skenario penambahan *noise* untuk mengetahui perubahan performa model pada kondisi audio yang berbeda. Tahapan penelitian meliputi *preprocessing* audio, ekstraksi fitur MFCC, pelatihan model SVM, serta evaluasi menggunakan *Accuracy, Precision, Recall, F1-Score, Confusion Matrix, dan Character Error Rate (CER)*. Hasil penelitian diharapkan dapat memberikan informasi mengenai pengaruh berbagai jenis noise terhadap akurasi sistem speech recognition serta tingkat ketahanan kombinasi MFCC dan SVM dalam mengenali perintah suara pada kondisi audio yang berbeda.

TINJAUAN PUSTAKA

Penelitian mengenai *speech recognition* terus berkembang seiring dengan meningkatnya pemanfaatan teknologi kecerdasan buatan dan pengolahan sinyal digital. Berbagai metode ekstraksi fitur dan algoritma klasifikasi telah digunakan untuk meningkatkan kemampuan sistem dalam mengenali ucapan atau perintah suara. Penelitian terdahulu menunjukkan bahwa kualitas fitur yang dihasilkan serta metode klasifikasi yang digunakan memiliki pengaruh yang signifikan terhadap performa sistem *speech recognition*.

Penelitian yang dilakukan oleh Rahman dkk. (2025) menganalisis performa pengenalan suara manusia menggunakan metode *Mel Frequency Cepstral Coefficients* (MFCC) dan *Convolutional Neural Network* (CNN). Penelitian tersebut menggunakan data suara dalam format WAV yang diproses melalui beberapa tahapan preprocessing dan ekstraksi fitur MFCC dengan variasi parameter koefisien serta panjang sinyal (*max length*). Hasil penelitian menunjukkan bahwa kombinasi MFCC dan CNN mampu menghasilkan akurasi tertinggi sebesar 98,96% pada parameter koefisien MFCC 40 dan *max length* 48000. Hasil tersebut menunjukkan bahwa parameter ekstraksi fitur MFCC berpengaruh terhadap performa sistem pengenalan suara dan mampu meningkatkan akurasi klasifikasi suara secara signifikan [25].

Penelitian yang dilakukan oleh Khizirova dkk. (2025) mengembangkan sistem identifikasi pembicara berbasis MFCC dan *Convolutional Neural Network* (CNN) dengan fokus pada peningkatan ketahanan sistem terhadap *noise*. Penelitian menggunakan data suara berbahasa Kazakh yang kemudian ditambahkan *pink noise* dengan berbagai tingkat *Signal-to-Noise Ratio* (SNR). Hasil pengujian menunjukkan bahwa akurasi identifikasi mencapai sekitar 96% pada kondisi tanpa *noise*, namun menurun menjadi sekitar 69% ketika data terpapar *noise*. Setelah diterapkan teknik augmentasi data pada proses pelatihan, akurasi meningkat kembali hingga sekitar 89–90%. Hasil tersebut menunjukkan bahwa noise memiliki pengaruh yang signifikan terhadap performa sistem pengenalan suara dan bahwa augmentasi data dapat meningkatkan ketahanan model terhadap gangguan akustik [21].

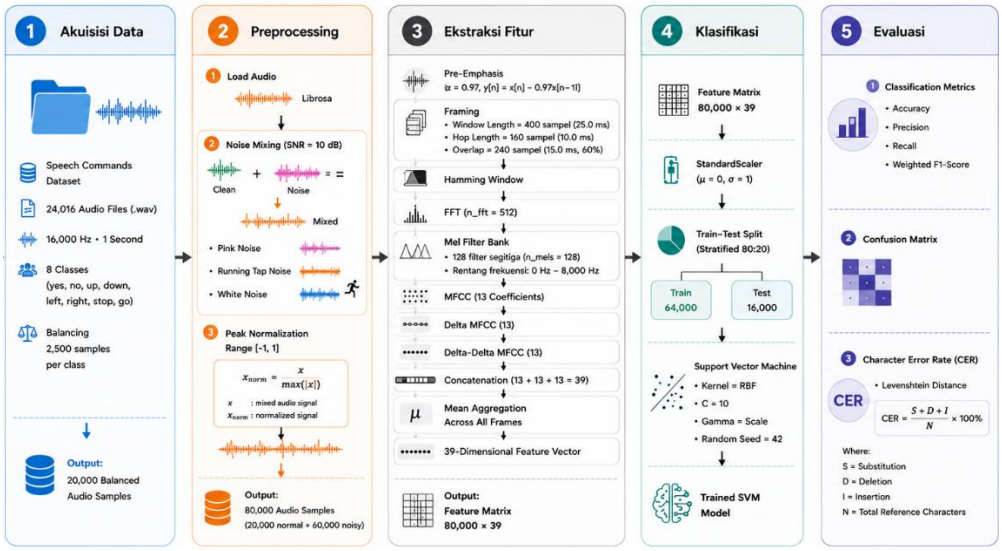
Penelitian yang dilakukan oleh Hanif dkk. (2026) menerapkan metode MFCC sebagai ekstraksi fitur dan *K-Nearest Neighbor* (KNN) sebagai metode klasifikasi untuk pengenalan perintah suara. Penelitian menggunakan empat

kelas perintah suara, yaitu buka, tutup, pesan, dan galeri yang diucapkan oleh beberapa pembicara berbeda. Hasil penelitian menunjukkan tingkat akurasi sebesar 60% pada perintah suara buka, 60% pada perintah suara tutup, 60% pada perintah suara pesan, dan 65% pada perintah suara galeri. Penelitian tersebut menunjukkan bahwa MFCC mampu mengekstraksi karakteristik suara yang dapat digunakan untuk proses klasifikasi perintah suara menggunakan algoritma *machine learning* [26].

Berdasarkan penelitian terdahulu, sebagian besar penelitian memanfaatkan MFCC sebagai metode ekstraksi fitur dan berbagai algoritma klasifikasi seperti CNN, KNN, maupun SVM untuk melakukan pengenalan suara. Namun, sebagian besar penelitian masih berfokus pada pengukuran akurasi klasifikasi pada kondisi audio normal. Penelitian yang secara khusus menganalisis pengaruh beberapa jenis *noise* terhadap performa sistem *speech recognition* menggunakan kombinasi MFCC dan SVM masih relatif terbatas. Selain itu, kajian mengenai perbandingan dampak berbagai jenis *noise* seperti *White Noise*, *Pink Noise*, dan *Running Tap Noise* terhadap kinerja model juga masih belum banyak dilakukan. Oleh karena itu, penelitian ini dilakukan untuk menganalisis pengaruh ketiga jenis *noise* tersebut terhadap performa speech recognition menggunakan MFCC dan SVM dengan evaluasi menggunakan *accuracy*, *precision*, *recall*, *weighted F1-score*, serta *Character Error Rate* (CER) sehingga dapat memberikan gambaran yang lebih komprehensif mengenai ketahanan sistem terhadap gangguan suara.

METODE

Metode penelitian pada penelitian ini dirancang untuk menganalisis pengaruh *noise* terhadap akurasi sistem speech recognition menggunakan ekstraksi fitur *Mel Frequency Cepstral Coefficients* (MFCC) dan metode klasifikasi *Support Vector Machine* (SVM). Tahapan penelitian meliputi Akuisisi Data, *Preprocessing* audio, Ekstraksi fitur MFCC, pelatihan model SVM, proses *speech recognition*, hingga evaluasi performa menggunakan *Character Error Rate* (CER). Alur penelitian ditunjukkan pada Gambar 1.



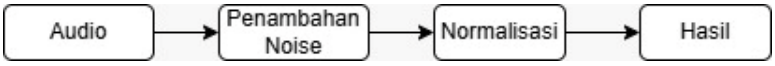
Gambar 1. Alur Penelitian

Akuisisi Data

Penelitian ini menggunakan dataset *Speech Commands* yang diperoleh dari platform Kaggle. Dataset tersebut berisi rekaman perintah suara berformat WAV dengan kanal mono dan laju pengambilan sampel sebesar 16.000 Hz. Setiap rekaman memiliki durasi 1 detik sehingga karakteristik data relatif seragam dan sesuai untuk proses klasifikasi perintah suara. Penelitian difokuskan pada delapan kelas perintah suara, yaitu *yes*, *no*, *up*, *down*, *left*, *right*, *stop*, dan *go*. Masing-masing kelas terdiri atas 2.500 rekaman suara sehingga total data yang digunakan sebanyak 20.000 audio. Keseragaman jumlah data pada setiap kelas bertujuan untuk mengurangi ketidakseimbangan kelas (*class imbalance*) yang dapat memengaruhi performa model klasifikasi [27].

Preprocessing Audio

Tahap *preprocessing* dilakukan untuk mempersiapkan data audio sebelum proses ekstraksi fitur. Pada tahap ini, setiap file audio dimuat menggunakan pustaka *Librosa* dan selanjutnya dilakukan penambahan *noise* sesuai skenario eksperimen yang telah ditentukan, yaitu *Pink Noise*, *Running Tap Noise*, dan *White Noise*. Proses pencampuran dilakukan menggunakan nilai *Signal-to-Noise Ratio* (SNR) sebesar 10 dB dengan menyesuaikan daya sinyal suara dan daya *noise* sebelum kedua sinyal digabungkan. Alur *preprocessing* yang digunakan pada penelitian ini ditunjukkan pada Gambar 1.



Gambar 2. Alur *Preprocessing*

Gambar 1 menunjukkan tahapan *preprocessing* yang dilakukan pada penelitian ini. Proses diawali dengan data audio asli sebagai masukan (*input*). Selanjutnya dilakukan penambahan *noise* sesuai dengan skenario eksperimen yang telah ditentukan. Setelah proses pencampuran selesai, sinyal audio dinormalisasi untuk menyamakan rentang amplitudo antar rekaman. Hasil normalisasi kemudian digunakan sebagai masukan pada tahap ekstraksi fitur MFCC. Dataset awal terdiri atas 20.000 data audio tanpa *noise*. Setelah proses penambahan tiga jenis *noise*, dihasilkan sebanyak 60.000 data audio tambahan. Dengan demikian, total data yang digunakan dalam penelitian ini berjumlah 80.000 data audio yang terdiri atas 20.000 data audio asli dan 60.000 data audio hasil penambahan *noise*. Setelah proses pencampuran selesai, amplitudo sinyal dinormalisasi ke rentang -1 hingga 1 untuk mengurangi perbedaan amplitudo antar rekaman dan menjaga kestabilan proses ekstraksi fitur. Normalisasi dilakukan menggunakan Persamaan (1).

$$x_{norm} = \frac{x}{\max(|x|)} \qquad (1)$$

Pada Persamaan (1), x merupakan sinyal audio hasil pencampuran *noise*, $\max(|x|)$ merupakan nilai amplitudo absolut maksimum pada sinyal audio, sedangkan x_{norm} merupakan sinyal audio yang telah dinormalisasi. Proses normalisasi dilakukan dengan membagi seluruh amplitudo sinyal terhadap nilai amplitudo maksimum sehingga rentang amplitudo berada pada interval -1 hingga 1. Langkah ini bertujuan untuk mengurangi perbedaan amplitudo antar rekaman dan menjaga kestabilan proses ekstraksi fitur [28].

Ekstraksi Fitur MFCC

Ekstraksi fitur dilakukan menggunakan metode *Mel Frequency Cepstral Coefficients* (MFCC) karena mampu merepresentasikan karakteristik suara sesuai dengan persepsi pendengaran manusia dan banyak digunakan pada sistem pengenalan ucapan [29]. Nilai *hop length* sebesar 160 dan *window length* sebesar 400 menghasilkan *overlap window* sebanyak 240 sampel atau sekitar 15 ms (60% dari panjang *window*). Pada tahap *Mel Filterbank*, digunakan 128 filter segitiga (*triangular filters*) dengan rentang frekuensi 0–8.000 Hz yang sesuai dengan frekuensi Nyquist pada *sampling rate* 16.000 Hz. Tahap ekstraksi fitur MFCC terdiri atas proses *framing*, *windowing*, *Fast Fourier Transform* (FFT), *Mel Filterbank*, dan *Discrete Cosine Transform* (DCT).

Proses *framing* dinyatakan pada Persamaan (2):

$$x_i(n) = x(n + iH) \qquad (2)$$

Fungsi Hamming pada tahap *windowing* dinyatakan pada Persamaan (3):

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right) \qquad (3)$$

Fast Fourier Transform (FFT) dinyatakan pada Persamaan (4):

$$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j2\pi kn/N} \qquad (4)$$

Konversi frekuensi ke skala Mel ditunjukkan pada Persamaan (5):

$$Mel(f) = 2595\log_{10}\left(1+\frac{f}{700}\right) \qquad (5)$$

Perhitungan koefisien MFCC menggunakan *Discrete Cosine Transform* (DCT) dinyatakan pada Persamaan (6):

$$C_n = \sum_{k=1}^{K} \log(S_k)\cos\left[\frac{\pi n}{K}\left(k-\frac{1}{2}\right)\right] \qquad (6)$$

Pada Persamaan (2), $x_i(n)$ menyatakan frame ke-i, $x(n)$ menyatakan sinyal suara, sedangkan H menyatakan nilai *hop length*. Pada Persamaan (3), $w(n)$ merupakan fungsi jendela Hamming, n merupakan indeks sampel, dan N

merupakan panjang *frame*. Pada Persamaan (4), $X(k)$ menyatakan spektrum frekuensi hasil transformasi Fourier dengan k sebagai indeks frekuensi. Pada Persamaan (5), f merupakan frekuensi dalam satuan Hertz (Hz), sedangkan $Mel(f)$ merupakan frekuensi yang telah dikonversi ke skala Mel. Pada Persamaan (6), C_n menyatakan koefisien MFCC ke-n, S_k merupakan keluaran *Mel Filterbank*, dan K merupakan jumlah filter Mel yang digunakan.

Hasil ekstraksi menghasilkan 13 koefisien MFCC. Selanjutnya dihitung fitur *Delta MFCC* dan *Delta-Delta MFCC* untuk merepresentasikan perubahan karakteristik spektral antar *frame* suara. Ketiga kelompok fitur tersebut kemudian digabungkan sehingga menghasilkan 39 koefisien fitur pada setiap *frame* audio. Untuk memperoleh representasi fitur yang lebih ringkas, dilakukan perhitungan nilai rata-rata (*mean*) pada setiap koefisien MFCC, *Delta MFCC*, dan *Delta-Delta MFCC*. Hasil proses tersebut menghasilkan vektor fitur berdimensi 39 yang digunakan sebagai masukan pada proses pelatihan dan pengujian model SVM.

Klasifikasi dan Validasi Model SVM

Proses pengenalan perintah suara dilakukan menggunakan metode *Support Vector Machine* (SVM). Sebelum proses pelatihan, seluruh fitur hasil ekstraksi MFCC dinormalisasi menggunakan *StandardScaler* untuk menyamakan skala antar fitur sehingga dapat meningkatkan stabilitas dan performa model klasifikasi [30]. Implementasi SVM dilakukan menggunakan pustaka Scikit-Learn dengan pendekatan klasifikasi multikelas (*multiclass classification*). Tahap klasifikasi menggunakan SVM terdiri atas proses pembentukan vektor fitur, penentuan *hyperplane* optimal, penerapan kernel *Radial Basis Function* (RBF), dan proses pengambilan keputusan (*decision function*).

Vektor fitur dinyatakan pada Persamaan (7):

$$\mathrm{x} = (x_1, x_2, \dots, x_n) \qquad (7)$$

Persamaan *hyperplane* pada SVM dinyatakan sebagai Persamaan (8):

$$\mathrm{w}^T\mathrm{x} + b = 0 \qquad (8)$$

Fungsi kernel RBF dinyatakan pada Persamaan (9):

$$K(x_i, x_j) = exp\left(-\gamma \left\| x_i - x_j \right\|^2\right) \qquad (9)$$

Fungsi keputusan (*decision function*) yang dinyatakan pada Persamaan (10):

$$f(x) = \mathrm{sign}(\textstyle\sum_{i=1}^{n} \alpha_i\, y_i K(x_i, x) + b) \qquad (10)$$

Pada Persamaan (7), x merupakan vektor fitur hasil ekstraksi MFCC. Pada Persamaan (8), w merupakan vektor bobot (*weight vector*), x merupakan vektor fitur masukan, dan b merupakan nilai bias. Pada Persamaan (9), $K(x_i, x_j)$ merupakan fungsi kernel RBF, sedangkan x_i dan x_j merupakan pasangan data masukan dan γ merupakan parameter kernel. Pada Persamaan (10), $f(x)$ merupakan fungsi keputusan (*decision function*), α_i merupakan koefisien *support vector*, y_i merupakan label kelas, dan b merupakan nilai bias.

Dataset yang telah digabungkan menjadi 80.000 sampel, yang terdiri atas 20.000 data audio normal dan 60.000 data audio hasil penambahan *noise*, kemudian dibagi menjadi data pelatihan dan data pengujian menggunakan metode *train-test split* dengan rasio 80:20. Proses pembagian data dilakukan menggunakan *stratified sampling* berdasarkan kombinasi label kelas dan kondisi eksperimen untuk menjaga proporsi distribusi data pada setiap kelas dan kondisi tetap seimbang pada data pelatihan maupun data pengujian. Hasil pembagian data menghasilkan 64.000 sampel sebagai data pelatihan dan 16.000 sampel sebagai data pengujian. Seluruh data pelatihan dari empat kondisi eksperimen digunakan untuk membangun satu model SVM gabungan (*combined model*). Dengan demikian, model dilatih menggunakan variasi data normal maupun data yang telah ditambahkan *noise* sehingga mampu mempelajari karakteristik suara pada berbagai kondisi lingkungan. Pada penelitian ini tidak dilakukan optimasi *hyperparameter* menggunakan *Grid Search* maupun *Cross Validation*. Parameter model ditetapkan secara langsung berdasarkan konfigurasi eksperimen yang telah ditentukan sebelumnya. Setelah proses pelatihan selesai, model digunakan untuk melakukan pengenalan perintah suara pada data pengujian. Hasil prediksi yang diperoleh selanjutnya digunakan pada tahap evaluasi untuk menganalisis pengaruh penambahan *Pink Noise*, *Running Tap Noise*, dan *White Noise* terhadap kinerja sistem pengenalan suara.

Evaluasi Kinerja Model

Kinerja sistem dievaluasi menggunakan data evaluasi pada setiap kondisi eksperimen, yaitu audio normal tanpa *noise*, audio dengan *Pink Noise*, audio dengan *Running Tap Noise*, dan audio dengan *White Noise*. Evaluasi dilakukan dengan membandingkan label hasil prediksi model terhadap label sebenarnya untuk mengetahui kemampuan sistem dalam mengenali perintah suara. Metrik evaluasi yang digunakan meliputi *Accuracy*, *Precision*, *Recall*, dan *Weighted F1-Score*. *Accuracy* digunakan untuk mengukur persentase prediksi yang sesuai dengan label sebenarnya. *Precision* digunakan untuk mengukur tingkat ketepatan prediksi pada setiap kelas, sedangkan *Recall* digunakan untuk mengukur kemampuan model dalam mengenali seluruh data yang termasuk ke dalam kelas tertentu. *Weighted F1-Score* digunakan untuk memberikan ukuran performa yang mempertimbangkan keseimbangan antara nilai *Precision* dan *Recall* pada seluruh kelas [31], [32].

Accuracy dihitung menggunakan Persamaan (11).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \qquad (11)$$

Precision dihitung menggunakan Persamaan (12).

$$Precision = \frac{TP}{TP+FP} \times 100\% \qquad (12)$$

Recall dihitung menggunakan Persamaan (13).

$$Recall = \frac{TP}{TP+FN} \times 100\% \qquad (13)$$

F1-Score dihitung menggunakan Persamaan (14).

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \times 100\% \qquad (14)$$

Pada Persamaan (2), (TP) (*True Positive*) menyatakan jumlah data yang berhasil diklasifikasikan dengan benar sebagai kelas positif, (TN) (*True Negative*) menyatakan jumlah data yang berhasil diklasifikasikan dengan benar sebagai kelas negatif, (FP) (*False Positive*) menyatakan jumlah data yang salah diklasifikasikan sebagai kelas positif, sedangkan (FN) (*False Negative*) menyatakan jumlah data yang salah diklasifikasikan sebagai kelas negatif. Nilai *Accuracy* digunakan untuk mengukur persentase keseluruhan prediksi yang berhasil dikenali dengan benar oleh model. Pada Persamaan (3), nilai *Precision* menunjukkan tingkat ketepatan model dalam memberikan prediksi terhadap suatu kelas. Semakin tinggi nilai *Precision*, semakin kecil jumlah data yang salah diprediksi sebagai kelas tertentu. Pada Persamaan (4), nilai *Recall* menunjukkan kemampuan model dalam menemukan seluruh data yang benar-benar termasuk ke dalam kelas yang diamati. Semakin tinggi nilai *Recall*, semakin sedikit data yang gagal dikenali oleh model. Pada Persamaan (5), *F1-Score* merupakan rata-rata harmonik antara nilai *Precision* dan *Recall*. Pada penelitian ini digunakan *Weighted F1-Score*, yaitu nilai F1 yang dihitung dengan mempertimbangkan proporsi jumlah data pada setiap kelas [33]. Semakin tinggi nilai *Accuracy*, *Precision*, *Recall*, dan *Weighted F1-Score*, semakin baik kemampuan sistem dalam mengenali perintah suara.

Selain menggunakan metrik numerik, penelitian ini juga memanfaatkan *Confusion Matrix* untuk menganalisis distribusi prediksi benar dan kesalahan pengenalan pada setiap kelas perintah suara [34]. Analisis ini digunakan untuk mengidentifikasi kelas yang paling sering mengalami kesalahan prediksi serta mengetahui pengaruh penambahan *noise* terhadap kinerja sistem *speech recognition*.

Evaluasi *Character Error Rate* (CER)

Selain metrik klasifikasi, penelitian ini menggunakan *Character Error Rate* (CER) sebagai evaluasi tambahan untuk mengukur tingkat kesalahan karakter antara label sebenarnya (*reference*) dan label hasil prediksi (*hypothesis*) [37]. Pada penelitian ini, label kelas asli digunakan sebagai *reference*, sedangkan hasil prediksi SVM digunakan sebagai *hypothesis*. Nilai CER dihitung berdasarkan jumlah operasi *substitution*, *deletion*, dan *insertion* yang diperlukan untuk mengubah hasil prediksi menjadi label referensi [35]. Character Error Rate (CER) dihitung menggunakan Persamaan (15):

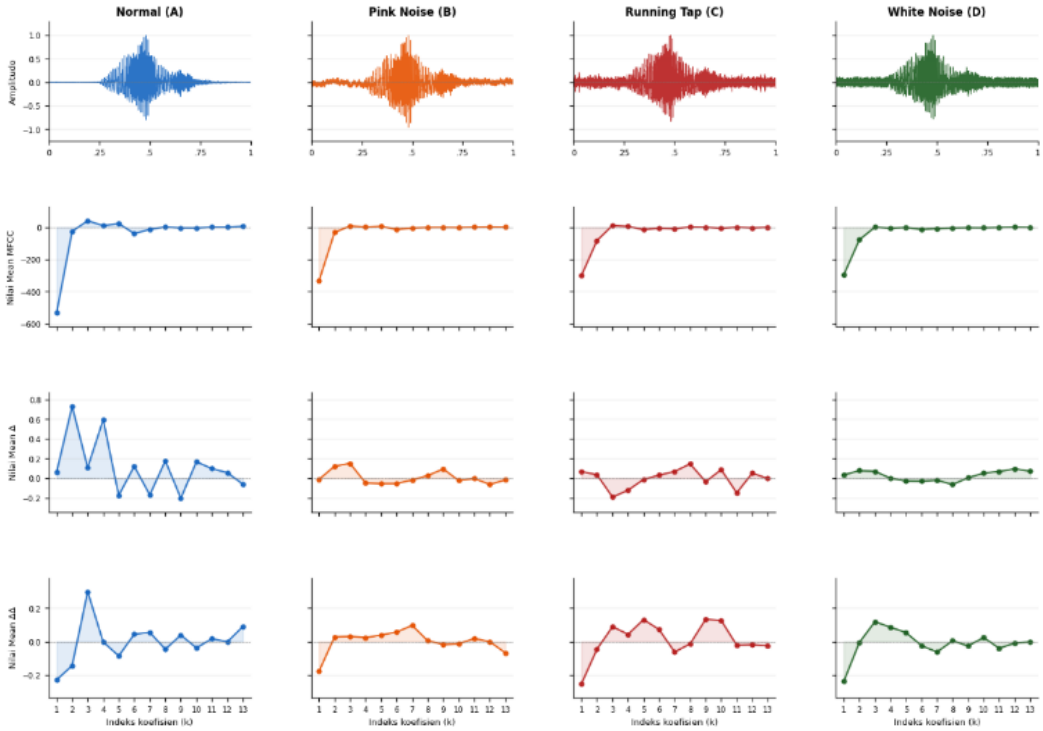
$$CER = \frac{S+D+I}{N} \times 100\% \qquad (15)$$

Pada Persamaan (15), S menyatakan jumlah karakter yang mengalami *substitution* (penggantian karakter), Dmenyatakan jumlah karakter yang mengalami *deletion* (penghapusan karakter), I menyatakan jumlah karakter yang mengalami *insertion* (penambahan karakter), sedangkan N merupakan jumlah karakter pada label referensi. Nilai CER menunjukkan persentase kesalahan karakter antara hasil prediksi dan label sebenarnya. Semakin kecil nilai CER yang diperoleh, semakin baik kemampuan sistem dalam menghasilkan prediksi yang sesuai dengan label referensi.

HASIL DAN PEMBAHASAN

Hasil Ekstraksi Fitur dan Pelatihan Model

Dataset yang digunakan pada penelitian ini terdiri atas 20.000 data audio asli yang terbagi ke dalam delapan kelas perintah suara, yaitu *yes*, *no*, *up*, *down*, *left*, *right*, *stop*, dan *go*. Setelah dilakukan penambahan tiga jenis *noise*, yaitu *Pink Noise*, *Running Tap Noise*, dan *White Noise*, jumlah data bertambah menjadi 80.000 sampel yang terdiri atas 20.000 data audio normal dan 60.000 data audio hasil penambahan *noise*. Proses ekstraksi fitur dilakukan menggunakan metode MFCC yang dikombinasikan dengan fitur *Delta MFCC* dan *Delta-Delta MFCC*. Setiap sinyal audio diekstraksi menjadi 13 koefisien MFCC, kemudian dihitung fitur *Delta MFCC* dan *Delta-Delta MFCC* sehingga diperoleh 39 koefisien fitur pada setiap frame audio. Selanjutnya dilakukan perhitungan nilai rata-rata (*mean*) pada seluruh koefisien untuk menghasilkan vektor fitur berdimensi 39 yang digunakan sebagai masukan pada model SVM. Hasil visualisasi sinyal audio dan fitur MFCC pada masing-masing kondisi eksperimen ditunjukkan pada Gambar 3.



Gambar 3. Visualisasi sinyal audio, MFCC, Delta MFCC, dan Delta-Delta MFCC pada empat kondisi eksperimen

Berdasarkan Gambar 3, terlihat bahwa penambahan *noise* menyebabkan perubahan pada bentuk gelombang (*waveform*) sinyal audio dibandingkan kondisi normal. Pada kondisi *Pink Noise*, *Running Tap Noise*, dan *White Noise*, amplitudo sinyal mengalami gangguan akibat adanya sinyal tambahan yang tercampur dengan suara asli. Perubahan tersebut turut memengaruhi nilai koefisien MFCC yang dihasilkan, terutama pada beberapa koefisien awal yang merepresentasikan karakteristik spektral utama dari sinyal suara. Selain memengaruhi nilai MFCC, penambahan *noise* juga menyebabkan perubahan pada pola *Delta MFCC* dan *Delta-Delta MFCC*. Perubahan ini menunjukkan bahwa karakteristik spektral antar *frame* menjadi lebih bervariasi setelah adanya gangguan suara. Kondisi tersebut terjadi karena *noise* menambahkan komponen frekuensi yang tidak berasal dari sinyal ucapan sehingga distribusi energi pada domain frekuensi mengalami perubahan. Akibatnya, representasi fitur yang dihasilkan tidak lagi sepenuhnya menggambarkan karakteristik suara asli dan berpotensi menurunkan kemampuan model dalam membedakan setiap kelas perintah suara.

Model klasifikasi dibangun menggunakan algoritma SVM dengan kernel *Radial Basis Function* (RBF), parameter $C = 10$, dan *gamma = scale*. Sebelum proses pelatihan dilakukan, seluruh fitur dinormalisasi menggunakan

StandardScaler. Dataset kemudian dibagi menjadi data pelatihan sebesar 80% dan data pengujian sebesar 20% menggunakan metode *stratified sampling* sehingga distribusi kelas dan kondisi eksperimen tetap seimbang. Hasil pembagian data menghasilkan 64.000 sampel data pelatihan dan 16.000 sampel data pengujian. Seluruh data pelatihan kemudian digunakan untuk membangun model SVM yang selanjutnya dievaluasi pada setiap kondisi eksperimen guna mengetahui pengaruh penambahan *noise* terhadap performa sistem *speech recognition*.

Hasil Evaluasi Sistem

Evaluasi dilakukan pada empat kondisi eksperimen, yaitu audio normal tanpa *noise*, audio dengan *Pink Noise*, audio dengan *Running Tap Noise*, dan audio dengan *White Noise*. Kinerja model diukur menggunakan metrik *Accuracy*, *Precision*, *Recall*, *Weighted F1-Score*, dan *Character Error Rate* (CER). Ringkasan hasil pengujian ditunjukkan pada Tabel 1.

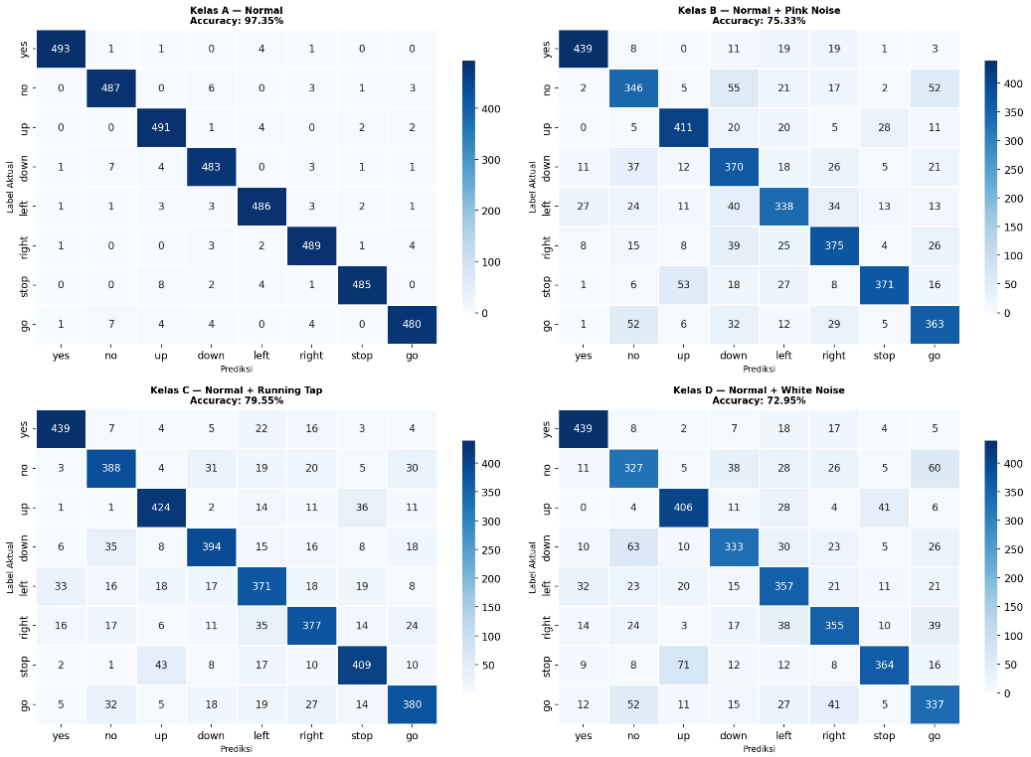
Tabel 1. Hasil Evaluasi Sistem

Eksperimen	Accuracy(%)	Precision(%)	Recall(%)	F1(%)	CER(%)
Normal	97.35	97.36	97.35	97.35	97.25
Pink Noise	75.33	75.79	75.33	75.44	73.72
Running Tap	79.55	79.55	79.55	79.54	77.76
White Noise	72.95	73.09	72.95	72.93	71.46

Berdasarkan hasil pengujian, kondisi audio normal menghasilkan performa terbaik dibandingkan seluruh kondisi audio yang telah ditambahkan *noise*. Nilai *Accuracy*, *Precision*, *Recall*, dan *Weighted F1-Score* cenderung mengalami penurunan setelah penambahan *noise*, sedangkan nilai CER menunjukkan peningkatan. Kondisi tersebut menunjukkan bahwa keberadaan *noise* memengaruhi kualitas informasi suara yang diterima sistem sehingga proses identifikasi perintah suara menjadi lebih sulit dilakukan. Perbedaan tingkat penurunan performa pada setiap skenario menunjukkan bahwa karakteristik masing-masing jenis *noise* memberikan pengaruh yang berbeda terhadap proses pengenalan ucapan. *Noise* yang memiliki spektrum frekuensi lebih luas cenderung menghasilkan gangguan yang lebih besar pada proses ekstraksi fitur sehingga menurunkan kemampuan model dalam membedakan karakteristik setiap kelas perintah suara. Hasil tersebut menunjukkan bahwa kombinasi MFCC dan SVM masih mampu mempertahankan performa pengenalan yang baik pada kondisi audio yang mengandung *noise*, meskipun terjadi penurunan performa dibandingkan kondisi audio normal.

Analisis *Confusion Matrix* dan *Character Error Rate*

Analisis *Confusion Matrix* dilakukan untuk mengevaluasi distribusi hasil klasifikasi pada setiap kelas perintah suara serta mengidentifikasi pola kesalahan prediksi yang terjadi pada masing-masing kondisi eksperimen. Hasil *Confusion Matrix* untuk seluruh skenario pengujian disajikan pada Gambar 4.



Gambar 4. Confusion Matrix

Gambar 4 menunjukkan hasil *Confusion Matrix* pada salah satu skenario pengujian. Secara umum, sebagian besar sampel berhasil diklasifikasikan sesuai dengan label sebenarnya. Namun demikian, penambahan *noise* menyebabkan peningkatan jumlah kesalahan prediksi pada beberapa kelas perintah suara. Kesalahan pengenalan umumnya terjadi pada kelas yang memiliki karakteristik akustik yang relatif mirip sehingga lebih sulit dibedakan ketika sinyal suara mengalami gangguan. Kondisi tersebut menunjukkan bahwa perubahan karakteristik spektral akibat *noise* dapat memengaruhi kualitas fitur MFCC yang dihasilkan dan berdampak pada proses klasifikasi yang dilakukan oleh model SVM.

Selain menggunakan metrik klasifikasi, penelitian ini juga mengevaluasi kinerja sistem menggunakan CER. Nilai CER digunakan untuk mengukur tingkat kesalahan karakter antara label sebenarnya dan hasil prediksi model. Hasil pengujian menunjukkan bahwa peningkatan tingkat kesalahan klasifikasi umumnya diikuti oleh peningkatan nilai CER. Dengan demikian, CER dapat digunakan sebagai indikator tambahan untuk mengevaluasi kualitas hasil pengenalan ucapan pada berbagai kondisi *noise*. Secara keseluruhan, hasil penelitian menunjukkan bahwa keberadaan *noise* memberikan pengaruh terhadap kinerja sistem *speech recognition*. Meskipun demikian, kombinasi MFCC dan SVM masih mampu menghasilkan performa yang baik dalam mengenali perintah suara pada kondisi normal maupun kondisi audio yang mengandung gangguan suara.

KESIMPULAN DAN SARAN

Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa penambahan *noise* (*Pink*, *Running Tap*, dan *White*) secara nyata menurunkan kinerja sistem pengenalan perintah suara berbasis MFCC dan SVM. Akurasi turun dari 97,35% (audio bersih) menjadi 75,33% (*Pink Noise*), 79,55% (*Running Tap*), dan 72,95% (*White Noise*), sementara *Character Error Rate* meningkat sejalan dengan penurunan metrik klasifikasi. Penurunan performa disebabkan oleh perubahan karakteristik spektral yang diakibatkan *noise*, *noise* menambahkan komponen frekuensi dan energi non-ucapan sehingga koefisien MFCC (beserta Delta dan Delta-Delta) mengalami pergeseran *mean* dan peningkatan *varians*. Akibatnya, jarak antar-centroid kelas di ruang fitur mengecil dan margin pemisah SVM menjadi kurang efektif sehingga frekuensi kesalahan klasifikasi meningkat. *White Noise* memberikan pengaruh paling besar karena spektrum frekuensinya yang sangat luas (menyebar di hampir semua frekuensi) mengganggu hampir seluruh bank Mel, sedangkan *Running Tap* yang lebih terstruktur (hanya pada frekuensi tertentu/*tonal*) menyebabkan penurunan yang relatif lebih kecil. *Pink Noise* berada di tengah karena spektrum frekuensinya lebih terkonsentrasi pada frekuensi rendah (mirip persepsi pendengaran manusia) sehingga dampaknya lebih ringan daripada *White Noise* namun lebih besar daripada *Running Tap*. Dengan demikian, kombinasi MFCC dan SVM masih menunjukkan ketahanan yang wajar pada tingkat kebisingan yang diuji,

namun peningkatan ketahanan pada kondisi lingkungan nyata memerlukan strategi tambahan seperti pengujian pada berbagai SNR, penerapan *speech enhancement*, augmentasi data, atau eksplorasi metode fitur dan klasifikasi lain.

Saran

Berdasarkan hasil penelitian yang telah dilakukan, beberapa saran dapat diberikan untuk pengembangan penelitian selanjutnya. Penelitian berikutnya disarankan untuk menguji sistem pada berbagai tingkat *Signal-to-Noise Ratio* (SNR) yang berbeda agar dapat diketahui batas ketahanan metode MFCC dan SVM terhadap gangguan suara dengan intensitas yang lebih beragam. Selain itu, penggunaan jenis noise lain yang lebih bervariasi juga perlu dilakukan untuk memperoleh gambaran yang lebih komprehensif mengenai performa sistem *speech recognition* pada berbagai kondisi lingkungan. Penelitian selanjutnya juga dapat menerapkan teknik peningkatan kualitas sinyal, seperti *noise reduction* atau *speech enhancement*, sebelum proses ekstraksi fitur dilakukan. Langkah tersebut diharapkan mampu mengurangi pengaruh *noise* terhadap karakteristik suara sehingga kualitas fitur yang dihasilkan menjadi lebih baik dan performa sistem dapat meningkat. Selain itu, metode ekstraksi fitur maupun algoritma klasifikasi lain dapat digunakan sebagai pembanding untuk mengetahui metode yang paling efektif dalam menghadapi gangguan *noise*.

DAFTAR PUSTAKA

[1] N. Xue, "Analysis Model of Spoken English Evaluation Algorithm Based on Intelligent Algorithm of Internet of Things," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–8, 2022, doi: 10.1155/2022/8469945.

[2] D. N. Sari, D. Kusnadi, R. H. Saputra, and M. U. Khan, "Digital Signal Processing for The Development of Deep Learning-Based Speech Recognition Technology," *International Journal of Electronics and Communications Systems*, vol. 4, no. 1, pp. 27–41, 2024, doi: 10.24042/ijecs.v4i1.22918.

[3] S. Rai, T. Li, and B. Lyu, "Keyword spotting -- Detecting commands in speech using deep learning," 2023, [Online]. Available: http://arxiv.org/abs/2312.05640

[4] M. Cohn, A. Lanzi, Y. Ishihara, C. Chuah, G. Zellou, and A. Weakley, "Accepted to Conference on Human Factors in Computing (CHI) 2026," 2026.

[5] A. Wibowo and A. R. Isnain, "Implementasi Algoritma Machine Learning untuk Klasifikasi Suara Lingkungan," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 5, no. 2, pp. 616–625, 2025, doi: 10.57152/malcom.v5i2.1712.

[6] K. M. Rezaul *et al.*, "Enhancing Audio Classification Through MFCC Feature Extraction and Data Augmentation with CNN and RNN Models," *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 7, pp. 37–53, 2024, doi: 10.14569/IJACSA.2024.0150704.

[7] A. A. Pangestu, O. V. Putra, J. T. Informatika, U. D. Gontor, K. Ponorogo, and J. Timur, "Pengembangan pengendali karakter permainan menggunakan suara real time berbasis lstm 1* 1," vol. 3, no. 1, pp. 417–424, 2026.

[8] H. Wijaya, "Teknologi Pengenalan Suara tentang Metode, Bahasa dan Tantangan: Systematic Literature Review," *bit-Tech*, vol. 7, no. 2, pp. 533–544, 2024, doi: 10.32877/bt.v7i2.1888.

[9] M. H. Herdiansyah, S. Hidayat, and N. Iksan, "Wavelet-Based MFCC and CNN Framework for Automatic Detection of Cleft Speech Disorders," *JTIM : Jurnal Teknologi Informasi dan Multimedia*, vol. 7, no. 3, pp. 653–662, 2025, doi: 10.35746/jtim.v7i3.780.

[10] N. A. Nokra and A. R. Hussian, "Statistical Study for Enhancing the MFCC Algorithm for Real-World and Highly Noisy Environments in Voiceprint Extraction," vol. 27, no. February, pp. 10–23, 2024.

[11] D. Li, Y. Gao, C. Zhu, Q. Wang, and R. Wang, "Improving Speech Recognition Performance in Noisy Environments by Enhancing Lip Reading Accuracy," *Sensors*, vol. 23, no. 4, 2023, doi: 10.3390/s23042053.

[12] V. S. Wicaksana and A. Z. S. Kom, "Spoken Language Identification on Local Language using MFCC, Random Forest, KNN, and GMM," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 5, pp. 394–398, 2021, doi: 10.14569/IJACSA.2021.0120548.

[13] G. Ajinurseto, L. O. Bakrim, and N. Islamuddin, "Penerapan Metode Mel Frequency Cepstral Coefficients pada Sistem Pengenalan Suara Berbasis Desktop," *Infomatek*, vol. 25, no. 1, pp. 11–20, 2023, doi: 10.23969/infomatek.v25i1.6109.

[14] J. Elektronik Ilmu Komputer Udayana, D. Kristomo, M. Mediadtrix Sebatubun, J. Raya Janti, and K. Jambe Yogyakarta, "Ekstraksi Ciri dan Klasifikasi Isyarat Suara Tutur Menggunakan Metode Mel Frequency Cepstral Coefficients," *JELIKU (Jurnal Elektronik Ilmu Komputer Udayana)*, vol. 13, no. 3, pp. 665–670, 2025, [Online]. Available: https://ojs.unud.ac.id/index.php/jlk/article/view/122028

[15] K. P. Naserwan, K. J. Miraswan, and M. Utari, "Performance Comparison of Naive Bayes and Support Vector Machine Methods in Music Genre Classification Based on Audio Signal Feature Extraction Using Mel-Frequency Cepstral Coefficients (MFCC)," vol. 10, no. 1, pp. 309–315, 2026.

[16] A. R. Nurjannah, E. S. Maydiana, and P. Kasih, "Pengenalan Suara Dzikir Menggunakan Ekstraksi Fitur MFCC dan Support Vector Machine," vol. 5, pp. 515–521, 2026.

[17] X. Kang, "Speech emotion recognition algorithm of intelligent robot based on ACO-SVM," *International Journal of Cognitive Computing in Engineering*, vol. 6, no. November 2024, pp. 131–142, 2025, doi: 10.1016/j.ijcce.2024.11.008.

[18] P. Studi Informatika, F. Matematika dan Ilmu Pengetahuan Alam, J. Raya Kampus Udayana, B. Jimbaran, K. Selatan, and B. Indonesia, "Analisis Penggunaan Metode MFCC dalam Mendeteksi Emosi pada Musik Indonesia I Komang Sutrisna Eka Wijaya a1 , Luh Arida Ayu Rahning Putri a2," vol. 1, pp. 1179–1186, 2023.

[19] S. Saifuddin, L. Azhari, E. Widarti, and W. Wartono, "Evaluasi Kinerja Kernel Linear, RBF, dan Polynomial pada Model Support Vector Machine untuk Prediksi Risiko Hipertensi," *Jurnal Ilmiah FIFO*, vol. 17, no. 2, p. 192, 2025, doi: 10.22441/fifo.2025.v17i2.008.

[20] R. T. Al-Hassani, D. C. Atilla, and Ç. Aydin, "Development of High Accuracy Classifier for the Speaker Recognition System," *Appl. Bionics Biomech.*, vol. 2021, 2021, doi: 10.1155/2021/5559616.

[21] M. Khizirova, K. Chezhimbayeva, A. Kassimov, M. Yermekbaev, and A. Iskakova, "Development and Increase of Noise Immunity of a Model of Biometric Identification of a Speaker Based on Metal-Frequency Cepstral Coefficients and a Convolutional Neural Network," *Eastern-European Journal of Enterprise Technologies*, vol. 6, no. 9, pp. 37–53, 2025, doi: 10.15587/1729-4061.2025.347451.

[22] W. Styorini *et al.*, "Implementasi Offline Speech Recognition Pada Home Device," *Seminar Nasional Teknologi Informasi Komunikasi dan Industri*, vol. 0, no. 0, pp. 265–273, 2022, [Online]. Available: https://ejournal.uin-suska.ac.id/index.php/SNTIKI/article/view/19326%0Afile:///C:/Users/user/Documents/MY JOB/JUDUL 1 PEMECAHAN MASALAH 500 JURNAL/19326-59951-1-PB.pdf

[23] M. Burhanis Sulthan, T. Nafisah, and L. Suhartini, "Komparasi Kinerja Algoritma Support Vector Machine-Recursive Feature Elimination (SVM-RFE) dan Recurrent Neural Network-Long Short Term Memory (RNN-LSTM) untuk Analisis Sentimen Review Mahasiswa," *Jurnal Aplikasi Teknologi Informasi dan Manajemen (JATIM)*, vol. 5, no. 2, pp. 112–122, 2024.

[24] T. Novela, Martin; Basaruddin, "Dataset Suara Dan Teks Berbahasa Indonesia Pada Rekaman," vol. 11, no. 2, pp. 61–66, 2021.

[25] F. N. Rahman, T. Listyorini, and E. Supriyati, "Analisis Akurasi Cnn Pada Data Olah Suara Manusia Menggunakan Parameter Koefisien Mfcc Dan Max Length," *Jurnal Digit*, vol. 15, no. 1, p. 1, 2025, doi: 10.51920/jd.v15i1.416.

[26] V. Issue, A. Hanif, F. Triadi, A. K. Jaya, S. Hartanto, and A. Basir, "JUTIN : Jurnal Teknik Industri Terintegrasi Identification of Speech Recognition Using K- Nearest Neighbor Method," vol. 9, no. 1, 2026, doi: 10.31004/jutin.v9i1.56699.

[27] S. Mutmainah, "Penanganan Imbalance Data Pada Klasifikasi Kemungkinan Penyakit Stroke," *Jurnal Sains, Nalar, dan Aplikasi Teknologi Informasi*, vol. 1, no. 1, pp. 10–16, 2021, doi: 10.20885/snati.v1i1.2.

[28] R. F. Christianti, "Estimasi Jarak Berbasis Convolutional Neural Network dan Regresi Dalam Sistem Deteksi Kendaraan Udara Nirawak Berbasis Suara," *Jurnal SINTA: Sistem Informasi dan Teknologi Komputasi*, vol. 2, no. 3, pp. 153–162, 2025, doi: 10.61124/sinta.v2i3.92.

[29] H. Lee and M. Nadeem, "Toward Efficient Speech Emotion Recognition via Spectral Learning and Attention," 2025, [Online]. Available: http://arxiv.org/abs/2507.03251

[30] R. Abrah, M. A. M. Hayat, and L. Lukman, "Penerapan Machine Learning untuk Mengklasifikasikan Genre Musik Berdasarkan Fitur Audio," *Arus Jurnal Sains dan Teknologi*, vol. 3, no. 2, pp. 204–211, 2025, doi: 10.57250/ajst.v3i2.1699.

[31] M. Adeel, Z. Y. Tao, S. Y. Jin, C. J. Guo, and M. Alsuhaibani, "Assessing random forest performance in low resource speech emotion recognition," *Sci. Rep.*, vol. 16, no. 1, pp. 1–94, 2026, doi: 10.1038/s41598-025-30511-6.

[32] H. Han, Y. Wu, J. Wang, and A. Han, *Interpretable machine learning assessment*, vol. 561. 2023. doi: 10.1016/j.neucom.2023.126891.

[33] S. Clara, D. Laksmi Prianto, R. Al Habsi, E. Friscila Lumbantobing, and N. Chamidah, "Feature Selection Implementation in Machine Learning Classification Algorithms to Predict Income on Adult Income Dataset," *Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA) Jakarta-Indonesia*, vol. 2, no. 1, pp. 741–747, 2021.

[34] A. Anggreni, A. Hakim, A. Rahman, and M. I. Dinata, "Klasifikasi Partisipasi Pemilih pada Pemilihan Walikota Bima Tahun 2024 Menggunakan Metode Naive Bayes Classifier," *Jurnal Sains Natural*, vol. 4, no. 1, pp. 1–13, 2025, doi: 10.35746/jsn.v4i1.898.

[35] Z. Chen, C. Zhong, and D. Chen, "A syllable-character collaborative model for enhanced Pinyin and Chinese recognition," *PLoS One*, vol. 20, no. 7 July, pp. 1–14, 2025, doi: 10.1371/journal.pone.0325045.