

Sistem Pendeteksian Bahasa Isyarat Menggunakan Algoritma Long Short-Term Memory

1st Ilham Muhamad Firdaus
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

ilhaammf@student.telkomuniversity.ac.id

2nd Casi Setianingsih
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

setiacasie@telkomuniversity.ac.id

3rd Marisa W. Paryasto
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

marisaparyasto@telkomuniversity.ac.id

Abstrak — Penyandang disabilitas khususnya tunarungu menggunakan bahasa isyarat sebagai alat komunikasi. Namun, masih banyak orang yang tidak mengerti atau terbatas dalam menggunakan bahasa isyarat, sehingga menciptakan hambatan dalam komunikasi dengan penyandang tunarungu. Dengan kemajuan teknologi informasi di era modern, ada peluang besar untuk membuat solusi untuk mengatasi masalah komunikasi. Untuk mengatasi masalah ini, digunakan algoritma pembelajaran mesin long short-term memory untuk mengenali gerakan dalam bahasa isyarat. Dalam pengujian, algoritma long short-term memory berhasil mendeteksi dan menerjemahkan bahasa isyarat dengan akurasi 98,3%. Penggunaan berbagai jumlah kelas dataset, analisis perbandingan distribusi dataset, pemilihan optimizer yang paling optimal, dan penyesuaian jumlah epoch telah diterapkan secara cermat untuk meningkatkan akurasi algoritma secara keseluruhan. Metode ini memastikan bahwa setiap komponen proses pelatihan model dioptimalkan untuk menghasilkan hasil yang paling akurat dan konsisten. Dengan akurasi sebesar 98,3%, algoritma long short-term memory menunjukkan kinerja yang sangat baik dalam memahami dan menerjemahkan bahasa isyarat. Secara keseluruhan, pengguna menilai sistem penerjemah bahasa isyarat ini baik, tetapi perlu optimalisasi lebih lanjut untuk memenuhi kebutuhan pengguna ke depannya.

Kata kunci — Penyandang tunarungu, bahasa isyarat, pembelajaran mesin, long short-term memory.

I. PENDAHULUAN

Masalah utama yang dihadapi penyandang disabilitas khususnya tunarungu adalah kurangnya aksesibilitas terhadap informasi dan komunikasi. Orang yang menggunakan bahasa isyarat sebagai cara utama untuk berkomunikasi dapat lebih mudah diakses jika mereka menerjemahkan bahasa isyarat ke dalam teks atau ucapan. Akibatnya, masalah ini menunjukkan betapa pentingnya mencari cara untuk memberi komunitas ini akses yang lebih baik ke informasi. Bahasa isyarat adalah bahasa di mana orang berkomunikasi dengan menggunakan bahasa tubuh, bentuk tangan, dan gerak bibir mereka, gerak wajah dan tangan mereka untuk mengungkapkan pikirannya [1].

Penyandang tunarungu utamanya berkomunikasi melalui bahasa isyarat. Di berbagai belahan dunia, terdapat berbagai

jenis bahasa isyarat yang berbeda, misalnya ASL atau American Sign Language di Amerika Serikat dan SIBI atau bahasa isyarat Indonesia di Indonesia. Bahasa isyarat membantu penyandang tunarungu berkomunikasi, tetapi masyarakat umum masih sangat terbatas dalam memahami dan menggunakan bahasa ini. Hal ini menimbulkan hambatan bagi penyandang tunarungu dan orang yang tidak mengerti bahasa isyarat untuk berkomunikasi. Perkembangan teknologi informasi di era modern memberikan peluang besar untuk membuat solusi untuk mengatasi tantangan komunikasi.

Kurangnya kebutuhan khusus penyandang tunarungu sering mengakibatkan minimnya perhatian terhadap penyediaan fasilitas komunikasi yang sesuai. Akibatnya, penyandang tunarungu harus mempelajari materi bahasa isyarat di luar dari jam pelajaran. Terlebih lagi media belajar mereka berpusat pada sebuah buku atau kamus. Hanya saja metode pembelajaran ini kurang interaktif dan masih 2 tradisional, serta masih sedikit tersedia media digital yang digunakan untuk pembelajaran bahasa isyarat.

Membaca buku adalah hal yang paling dasar dilakukan untuk mempelajari suatu ilmu, begitu juga dengan penyandang tunarungu menggunakan buku untuk mempelajari bahasa isyarat agar mereka berinteraksi satu sama lain. Salah satu buku yang membahas tentang pembelajaran bahasa isyarat adalah “Kamus Sistem Bahasa Isyarat Indonesia”. Dan terdapat beberapa website yang menyediakan pembelajaran bahasa isyarat untuk penyandang tunarungu, dan masyarakat umum. Website ini dapat diakses dengan mengunjungi laman website resmi Kamus SIBI yang dibuat oleh Kementerian Pendidikan dan Budaya Indonesia [2].

II. KAJIAN TEORI

Untuk membuat sistem penerjemahan bahasa isyarat yang efektif, diperlukan kombinasi berbagai keterampilan yang kompleks dan saling melengkapi. Machine learning adalah salah satu keahlian utama [3]. Ini mencakup pemahaman mendalam tentang algoritma dan teknik yang digunakan untuk mengajarkan mesin untuk mengenali dan memahami pola data seperti gerakan tangan dan ekspresi wajah bahasa isyarat. Pengetahuan tentang bahasa isyarat juga sangat

penting. Bahasa isyarat bukan hanya gerakan tangan, akan tetapi merupakan sistem komunikasi yang kaya dan kompleks dengan berbagai budaya, tata bahasa, dan nuansa ekspresi. Karena setiap gerakan tangan atau ekspresi wajah memiliki arti yang berbeda, memahami tentang hal ini membantu dalam pembuatan algoritma pengajaran mesin yang lebih akurat dan responsif. Bahasa isyarat yang disebut Sistem Isyarat Bahasa Indonesia (SIBI) dibuat untuk memenuhi kebutuhan komunikasi penyandang tunarungu di Indonesia [4]. Sebagai alat komunikasi utama bagi komunitas tunarungu, SIBI membantu mereka berinteraksi dan mengekspresikan diri dalam kehidupan sehari-hari, baik dalam lingkungan sosial, akademis, maupun di lingkungan kerja. Rutkumar Nitinchandra Patel dari SVIT, Vasad, Anand, Gujarat, India, melakukan penelitian bahasa isyarat dengan pembelajaran mesin. Metode Long Short-Term Memory (LSTM) digunakan untuk menerapkan konsep pembelajaran mendalam dalam penelitian ini. Penelitian ini bertujuan untuk membantu mereka yang mengalami gangguan pendengaran. Hasil penelitian menunjukkan bahwa dengan menggunakan algoritma pembelajaran mesin untuk proses pelatihan dan pengujian data, LSTM dapat menerjemahkan bahasa isyarat dengan akurasi 66% [5].

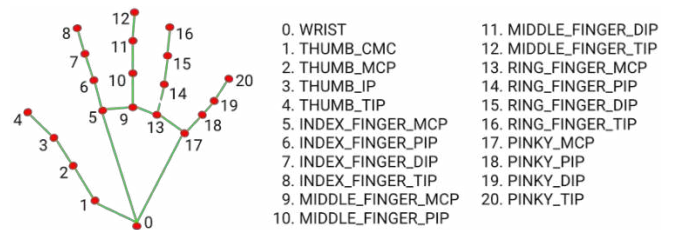
III. METODE

Studi ini mengembangkan sistem penerjemahan bahasa isyarat Indonesia dengan menggunakan algoritma LSTM. Algoritma LSTM digunakan dalam penerjemahan bahasa isyarat untuk mengidentifikasi dan menginterpretasikan rangkaian gerakan tangan dari waktu ke waktu. Sistem pengenalan bahasa isyarat Indonesia menggunakan algoritma LSTM untuk mendeteksi gerakan tangan. Tujuannya adalah untuk membuat sistem pengenalan gerakan bahasa isyarat Indonesia yang tepat dan efisien yang bisa digunakan di dalam lingkungan sekolah.

Untuk mencapai tujuannya, penelitian ini menggunakan pendekatan terstruktur, yang dibagi menjadi beberapa tahap. Pertama adalah pengumpulan dataset pelatihan dan dataset pengujian sistem. Setelah itu, lanjut pada proses preprocessing. Selain itu, langkah ketiga adalah melatih model dengan dataset pelatihan yang telah disiapkan sebelumnya. Langkah terakhir adalah mengevaluasi model dengan parameter tertentu. Proses pembuatan sistem yang dapat mendeteksi bahasa isyarat digambarkan sebagai berikut:

A. Proses Pengumpulan Dataset

Pengambilan dataset untuk gerakan bahasa isyarat dilakukan dengan sebuah laptop yang dilengkapi dengan webcam eksternal untuk hasil yang maksimal dan memanfaatkan Library MediaPipe sebagai sensor untuk mendeteksi gerakan. Library MediaPipe adalah sebuah model neural network yang dibuat oleh Google yang memiliki fungsi untuk mendeteksi keberadaan titik-titik pada tangan manusia. Library ini akan mendeteksi 21 titik pada telapak tangan manusia.



GAMBAR 1.

Penanda tangan dari library mediapipe.

Salah satu masalah yang menghambat pengambilan dataset adalah delay dari kamera. Library MediaPipe mengalami delay ketika ada perubahan gerakan atau ketika melakukan beberapa gerakan dinamis, dan menggunakan kedua tangan, yang menyebabkan error dalam proses deteksi gerakan.

B. Proses Preprocessing Dataset

Dalam proses pengumpulan dataset, hal yang harus dilakukan selanjutnya adalah pengecekan dataset untuk melihat apakah ada dataset yang koordinat keypoints-nya kosong. Untuk mengatasi masalah ini, setiap keypoints diperiksa apakah ada koordinat yang kosong.

```
keypoints_0.npy contains only zero values.
keypoints_1.npy contains only zero values.
keypoints_2.npy contains only zero values.
keypoints_3.npy contains only zero values.
keypoints_4.npy contains only zero values.
keypoints_5.npy contains only zero values.
keypoints_6.npy contains only zero values.
```

GAMBAR 2.

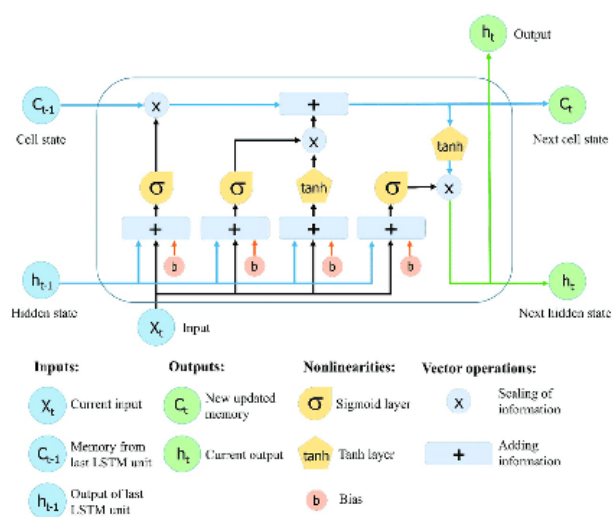
Dataset yang koordinat keypoints-nya kosong.

Pengujian di atas untuk mengenali keypoints dari gerakan bahasa isyarat selama pengambilan dataset. Keypoints ini sangat penting untuk menentukan posisi dan gerakan tangan dalam setiap frame video. Dataset ini kemudian digunakan untuk melatih model algoritma LSTM.

C. Proses Pelatihan Model

Pada pembuatan aplikasi penerjemah bahasa isyarat ini menggunakan algoritma LSTM, LSTM adalah jenis khusus dari algoritma RNN atau Recurrent Neural Network yang dirancang untuk memprediksi informasi yang terkait dengan waktu. Algoritma LSTM dipilih dalam pembuatan aplikasi ini karena algoritma LSTM dapat membaca dataset dalam bentuk sequence.

Algoritma LSTM terdiri dari beberapa komponen yaitu cell, input gate, output gate, dan forget gate. Gerbang-gerbang ini mengatur aliran informasi ke dalam dan ke luar sel, yang memungkinkan jaringan untuk secara selektif mempertahankan atau membuang informasi saat memproses data secara berurutan. Berikut adalah gambar arsitektur algoritma LSTM:

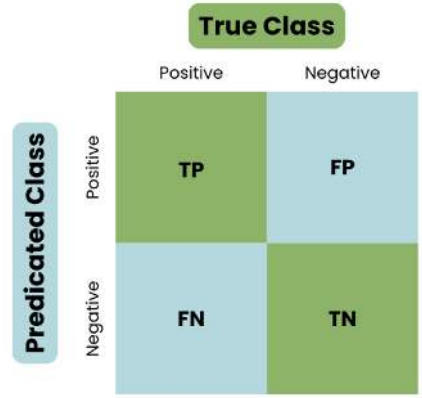


GAMBAR 3.
Arsitektur algoritma LSTM.

Setelah dataset melalui proses preprocessing, dataset diatas digunakan untuk proses training model algoritma LSTM. Hasil pelatihan menunjukkan bahwa model sangat akurat dan memiliki nilai loss yang rendah. Nilai akurasi yang tinggi menunjukkan bahwa model memiliki kemampuan yang sangat baik dalam mengenali dan mengklasifikasikan data dengan benar, dan nilai loss yang rendah menunjukkan bahwa model memiliki tingkat kesalahan yang kecil dalam prediksi.

D. Proses Evaluasi Model

Untuk memastikan bahwa model algoritma LSTM untuk sistem pendeteksi bahasa isyarat berfungsi secara optimal, beberapa parameter harus dipertimbangkan ketika mengevaluasi model. Pertama, confusion matrix dan akurasi digunakan untuk mengukur seberapa baik model mengklasifikasikan isyarat dengan benar. Hal ini juga memberikan gambaran tentang kinerja umum dan distribusi kesalahan klasifikasi. Struktur confusion matrix adalah alat yang sangat penting dalam pembelajaran mesin yang memungkinkan evaluasi yang komprehensif terhadap kinerja model klasifikasi. Ini sangat berguna untuk menilai model yang output-nya mungkin termasuk dalam banyak kelas. Confusion matrix adalah matriks persegi dengan baris yang mewakili kelas nyata (atau label sebenarnya) dan kolom yang menunjukkan kelas yang diprediksi oleh model.



GAMBAR 4.
Struktur confusion matrix.

Confusion matrix adalah alat yang penting untuk mengevaluasi model klasifikasi karena memberikan rincian menyeluruh tentang kinerja model dengan membandingkan kelas yang sebenarnya dan kelas yang diantisipasi. False positive (FP), true negative (TN), true positive (TP), dan false negative (FN) adalah empat komponen utama matriks ini. Faktor-faktor ini berkontribusi pada metrik penting seperti akurasi, presisi, recall, dan F1-score, yang mengevaluasi kemampuan model untuk mengidentifikasi contoh secara akurat [6].

IV. HASIL DAN PEMBAHASAN

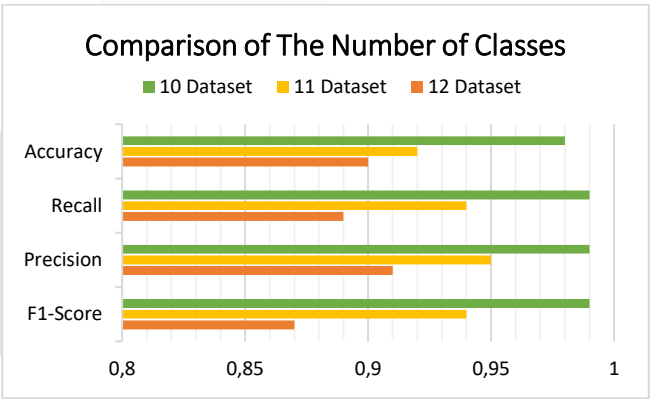
Untuk memastikan bahwa model bekerja dengan baik dan siap untuk digunakan, beberapa tahap pengujian harus dilakukan sebelum menggunakan algoritma Long Short-Term Memory (LSTM). Berikut adalah daftar pengujian yang perlu dilakukan:

A. Jumlah Kelas Dalam Dataset

Jumlah kelas dalam dataset sangat penting untuk menentukan arsitektur model LSTM. Jumlah kelas dataset yang berbeda adalah 10 kelas, 11 kelas, dan 12 kelas. Dataset tersebut merupakan kata yang sering digunakan dalam percakapan, dimana terdapat kemiripan gerakan dari bahasa isyarat tersebut. Berikut ini adalah tabel perbedaan jumlah skenario uji coba dataset:

TABEL 1.
Perbandingan dari jumlah kelas dataset.

Jumlah kelas	Accuracy	Recall	Precision	F1-score
10	0.98	0.99	0.99	0.99
11	0.92	0.94	0.95	0.94
12	0.9	0.89	0.91	0.87



GAMBAR 5.
Visualisasi perbandingan dari jumlah kelas dataset.

Dari hasil pengujian dengan berbagai jumlah kelas dataset, ditemukan bahwa model memiliki performa terbaik dengan 10 kelas, dengan akurasi sebesar 0.98, recall sebesar 0.99, precision sebesar 0.99, dan F1-score sebesar 0.99. Namun, ketika jumlah kelas ditambah menjadi 11, performa menurun dengan akurasi sebesar 0.92, recall sebesar 0.94, precision sebesar 0.95, dan F1-score sebesar 0.94. Penurunan lebih lanjut terjadi pada pengujian dengan 12 kelas, di mana akurasi turun menjadi 0.90, recall menjadi 0.89, precision

menjadi 0.91, dan F1-score menjadi 0.87, yang mengindikasikan bahwa semakin banyak kelas semakin menyulitkan model untuk mengklasifikasikan dan membedakan kelas dengan benar.

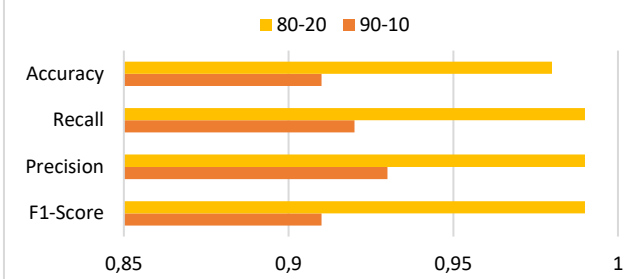
B. Distribusi Data yang akan Digunakan

Pembagian dataset menjadi data pelatihan dan data validasi merupakan langkah penting dalam pemodelan dengan algoritma LSTM. Hal ini dilakukan untuk memastikan bahwa model dapat belajar secara efektif tanpa adanya overfitting dan validasi model dapat diukur dengan tepat. Berikut ini adalah hasil uji dari distribusi dataset:

TABEL 2.
Perbandingan dari distribusi dataset.

Distribusi data	Accuracy	Recall	Precision	F1-score
80:20	0.98	0.99	0.99	0.99
90:10	0.91	0.92	0.93	0.91

Comparison of The Data Distribution



GAMBAR 6.

Visualisasi perbandingan dari distribusi dataset.

Pada skenario pertama, di mana data dibagi dengan rasio 80% untuk dataset pelatihan dan 20% untuk dataset validasi. Model ini mencapai kinerja yang sangat tinggi dengan akurasi, recall, presisi, dan F1-score yang semuanya mendekati nilai sempurna yaitu 0,99. Sementara itu, skenario kedua dengan 90% data untuk pelatihan dan hanya 10% untuk validasi, menunjukkan penurunan kinerja di semua metrik: akurasi 0,91, recall 0,92, presisi 0,93, dan F1-score 0,91. Penurunan ini mengindikasikan risiko overfitting yang lebih tinggi ketika data validasi dikurangi. Akibatnya, model menjadi kurang cocok untuk data baru atau data yang tidak terlihat.

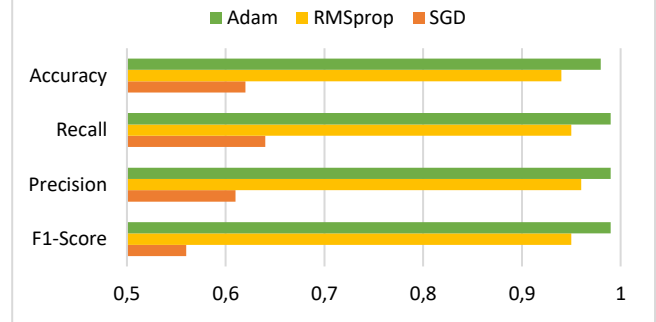
C. Memilih Optimizer Terbaik

Dalam melatih model algoritma Long Short-Term Memory (LSTM), pemilihan optimizer sangat penting karena merupakan bagian penting dalam proses pelatihan yang mempengaruhi kecepatan dan kualitas pembelajaran. Berikut ini adalah skenario yang dilakukan dalam pengujian masing-masing optimizer:

TABEL 3.
Perbandingan dari optimizer.

Optimizer	Accuracy	Recall	Precision	F1-score
Adam	0.98	0.99	0.99	0.99
RMSprop	0.94	0.95	0.96	0.95
SGD	0.62	0.64	0.61	0.56

Comparison of The Best Optimizer



GAMBAR 7.

Visualisasi perbandingan dari optimizer.

Pada skenario pengujian pertama dengan optimizer Adam menunjukkan hasil yang paling impresif dengan mendapatkan nilai akurasi sebesar 0.98, recall 0.99, precision 0.99, dan F1-score 0.99. Dan pada skenario pengujian kedua dengan optimizer RMSprop juga menunjukkan performa yang cukup baik dengan nilai akurasi 0.94, recall 0.95, precision 0.96, dan F1-score 0.95. Skenario pengujian ketiga dengan optimizer SGD memberikan hasil yang kurang memuaskan dibandingkan dua optimizer sebelumnya, dengan hanya memperoleh akurasi sebesar 0.62, recall sebesar 0.64, precision sebesar 0.61, dan F1-score sebesar 0.56.

D. Memilih jumlah epoch

Salah satu metode terbaik untuk menghindari overfitting dan memaksimalkan performa model dalam pelatihan model algoritma LSTM adalah dengan menggunakan callback stop-training. Menggunakan fitur callback untuk memilih jumlah epoch. Pada proses pelatihan algoritma LSTM, callback memungkinkan untuk menghentikan proses pelatihan ketika performa model tidak membaik lagi setelah epoch yang ditentukan. Jadi, tidak ada batasan untuk epoch, namun tetap ada ambang batas yang menjadi batas epoch untuk berhenti.

Meskipun memungkinkan untuk melatih model LSTM dengan jumlah epoch yang sudah ditentukan, dalam praktiknya penggunaan callback seperti Early Stopping membantu untuk menghentikan proses pelatihan secara otomatis ketika model sudah tidak menunjukkan peningkatan performa.

E. Pengujian Kondisi Eksternal

Pengujian kondisi eksternal mencakup faktor pengujian seperti intensitas cahaya dan jarak. Jenis pengujian ini sering kali penting untuk menilai bagaimana suatu sistem akan berfungsi dalam berbagai kondisi lingkungan yang dapat memengaruhi kinerjanya. Pengujian berikut ini akan dilakukan:

1. Pengujian intensitas cahaya

Pengujian intensitas cahaya dalam algoritma Long Short-Term Memory (LSTM), tentunya sangat penting karena keypoints tidak akan muncul pada kamera jika tidak ada cukup cahaya. Maka dari itu dilakukan pengujian intensitas cahaya sebagai berikut:

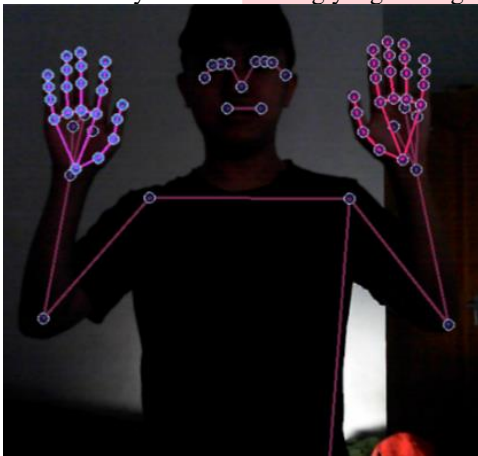
a. Intensitas cahaya di dalam ruang yang cukup cahaya.



GAMBAR 8.

Intensitas cahaya dalam ruangan yang cukup cahaya.

b. Intensitas cahaya di dalam ruang yang kurang cahaya.



GAMBAR 9.

Intensitas cahaya di dalam ruang yang kurang cahaya.

c. Intensitas cahaya di luar ruang yang cukup cahaya.



GAMBAR 10.

Intensitas cahaya di luar ruang yang cukup cahaya.

Dari tiga percobaan yang dilakukan menggunakan sistem deteksi gesture tangan menggunakan keypoints dengan kondisi pencahayaan yang berbeda, hasilnya menunjukkan pengaruh signifikan dari kondisi cahaya terhadap performa sistem. Sistem deteksi gesture tangan menggunakan keypoints sangat sensitif terhadap kondisi pencahayaan. Performa sistem paling optimal dalam ruangan dengan pencahayaan yang cukup, sementara pencahayaan yang kurang atau kondisi luar ruangan dengan faktor variabel lain mempengaruhi keakuratan sistem secara signifikan. Untuk

penggunaan yang efektif, disarankan untuk memastikan bahwa pencahayaan selalu cukup.

2. Pengujian multi pengguna saat deteksi gesture

Pengujian skenario multi-pengguna dengan sistem LSTM hanya mampu mendeteksi satu pengguna pada satu waktu, sehingga memerlukan strategi untuk menentukan mana pengguna yang akan spesifik. Dengan menggunakan kriteria seperti posisi terdekat ke kamera atau area fokus tertentu dalam frame, sistem dapat memilih satu pengguna untuk dideteksi, mengabaikan pengguna lain yang juga berada dalam frame. Berikut adalah contoh dalam penggunaannya:



GAMBAR 11.

Dua pengguna, tetapi hanya satu yang difokuskan.

3. Pengujian jarak pada saat deteksi gesture

Pengujian jarak dalam penggunaan algoritma Long Short-Term Memory (LSTM) sebenarnya tidak ada aturan khususnya. Tetapi deteksi gesture menggunakan keypoints setidaknya mencakup setengah bagian atas tubuh melalui kamera. Meskipun tidak ada aturan yang jelas tentang seberapa jauh jarak dari kamera, satu hal yang sangat penting adalah bahwa kamera harus dapat melihat setengah bagian atas tubuh. Karena keypoints yang mendeteksi gesture di bagian atas tubuh seperti tangan, kepala, dan bahu.



GAMBAR 12.

Jarak yang disarankan dalam deteksi gerakan.

V. KESIMPULAN

Algoritma LSTM dapat mendeteksi dan menerjemahkan gerakan tangan yang membentuk bahasa isyarat. Dengan akurasi 98,3%, algoritme ini menunjukkan kinerja yang sangat baik dalam memahami dan menerjemahkan bahasa

isyarat. Hasil ini menunjukkan bahwa algoritma LSTM dapat menjadi algoritma yang efektif untuk mengembangkan sistem pendeteksi bahasa isyarat yang dapat diterapkan untuk berbagai aplikasi di dunia nyata. Penggunaan algoritma LSTM dalam penerjemahan bahasa isyarat menawarkan banyak peluang untuk meningkatkan komunikasi dan aksesibilitas bagi komunitas penyandang tunarungu.

REFERENSI

- [1] J. Khikam Hikmalansya, D. Cahyono, dan S. Surabaya, "Aplikasi Pembelajaran Bahasa Isyarat Berbasis Android," 2016.
- [2] kemdikbud, "Kamus SIBI Kerjasama antara Kementerian Pendidikan dan Kebudayaan dengan Lembaga Penelitian dan Pengembangan Sistem Isyarat Bahasa Indonesia ;," kemdikbud.go.id. Diakses: 19 Oktober 2023. [Daring]. Tersedia pada: <https://pmpk.kemdikbud.go.id/sibi/>
- [3] D. Bragg dkk., "Sign language recognition, generation, and translation: An interdisciplinary perspective," dalam ASSETS 2019 - 21st International ACM SIGACCESS Conference on Computers and Accessibility, Association for Computing Machinery, Inc, Okt 2019, hlm. 16–31. doi: 10.1145/3308561.3353774.
- [4] "MENELITI BAHASA ISYARAT DALAM PERSPEKTIF VARIASI BAHASA Silva Tenrisara Isma Abstrak."
- [5] R. B. Nitinchandra Patel, "SIGN LANGUAGE RECOGNITION USING LSTM DEEP LEARNING MODEL," 2022. [Daring]. Tersedia pada: <http://www.ijeast.com>
- [6] Nisha Arya Ahmed, "What is A Confusion Matrix in Machine Learning? The Model Evaluation Tool Explained," datacamp.com. Diakses: 27 Juli 2024. [Daring]. Tersedia pada: <https://www.datacamp.com/tutorial/what-is-a-confusion-matrix-in-machine-learning>