

# Implementasi Model Deep Learning Untuk Deteksi Gerakan Tangan Pada Bahasa Isyarat Indonesia

1<sup>st</sup> Siti Elna Fauziah

Fakultas Ilmu Terapan

Universitas Telkom

Bandung, Indonesia

selnafauziah@student.telkomuniversity.ac.id

2<sup>nd</sup> Yuli Sun Hariyani

Fakultas Ilmu Terapan

Universitas Telkom

Bandung, Indonesia

yulisun@telkomuniversity.ac.id

3<sup>rd</sup> Suci Aulia

Fakultas Ilmu Terapan

Universitas Telkom

Bandung, Indonesia

sucia@tass.telkomuniversity.ac.id

**Abstrak** — Bahasa isyarat merupakan sarana komunikasi utama bagi teman tuli. Namun, kurangnya pemahaman masyarakat terhadap bahasa ini sering menjadi hambatan dalam berinteraksi. Penelitian ini bertujuan mengembangkan sistem deteksi gerakan tangan pada Bahasa Isyarat Indonesia (SIBI) dengan memanfaatkan model deep learning berbasis Convolutional Neural Network (CNN). Sistem dirancang untuk mengenali 26 huruf alfabet, angka 0–9, serta lima kata dasar, yaitu “saya”, “kamu”, “dia”, “makan”, dan “tidur”. Implementasi sistem menggunakan bahasa pemrograman Python dengan framework TensorFlow, serta pustaka pendukung seperti OpenCV, MediaPipe, scikit-learn, seaborn, dan numpy. Dataset dikumpulkan mandiri menggunakan kamera laptop, disimpan dalam format .h5, dan diberi label dengan bantuan Roboflow. Perangkat keras yang digunakan berupa laptop dengan GPU serta webcam internal untuk proses pengambilan data dan pengujian. Hasil pengujian menunjukkan sistem mampu berjalan secara real-time dengan akurasi 100% saat diuji terpisah pada huruf, angka, maupun lima kata dasar. Pada pengujian gabungan, akurasi mencapai ±97%. Meski kadang terjadi kesalahan akibat kemiripan bentuk gerakan, sistem tetap stabil dan dapat digunakan untuk membantu penerjemahan bahasa isyarat, sehingga mempermudah komunikasi antara teman tuli dan masyarakat. Kata kunci— bahasa isyarat indonesia, convolutional neural network, mediapipe, tensorflow, real-time processing, aksesibilitas komunikasi

## I. PENDAHULUAN

Bahasa isyarat adalah sarana komunikasi utama bagi teman tuli. Di Indonesia terdapat dua sistem yang umum digunakan, yakni Bahasa Isyarat Indonesia (BISINDO) dan Sistem Isyarat Bahasa Indonesia (SIBI) [1]. SIBI diresmikan pemerintah dengan mengadopsi tata bahasa Bahasa Indonesia, namun penggunaannya belum meluas akibat rendahnya pemahaman masyarakat, minimnya edukasi, dan keterbatasan teknologi [2].

Kemajuan teknologi komputer dan kecerdasan buatan, khususnya deep learning, membuka peluang pengenalan bahasa isyarat secara otomatis. Penelitian sebelumnya dengan CNN, LSTM, atau kombinasinya masih terbatas pada

huruf atau angka serta bergantung pada pencahayaan, sudut kamera, dan dataset yang terbatas [3].

Penelitian ini bertujuan untuk :

1. Merancang dan membangun sistem pengenalan bahasa isyarat SIBI berbasis CNN-LSTM dengan dukungan MediaPipe untuk deteksi real-time.
2. Mengembangkan dataset khusus yang mencakup 26 huruf alfabet, 10 angka, dan 5 kata umum SIBI.
3. Mengukur performa sistem dalam mengenali kategori isyarat secara akurat pada berbagai kondisi pencahayaan dan sudut pandang kamera.

## II. KAJIAN TEORI

### A. Sistem Isyarat Bahasa Indonesia (SIBI)

SIBI adalah sistem komunikasi visual resmi di Indonesia yang mengadopsi tata bahasa Bahasa Indonesia. Digunakan terutama di pendidikan formal untuk teman tuli, namun penyebarannya terbatas karena kurangnya sosialisasi, edukasi, dan media teknologi pendukung [4].



GAMBAR 1  
SIBI

### B. Convolutional Neural Network (CNN)

CNN adalah arsitektur deep learning untuk pengolahan citra yang mengekstraksi fitur secara otomatis melalui lapisan konvolusi dan pooling. Cocok untuk mengenali pola bentuk tangan dalam bahasa isyarat.

### C. Long Short-Term Memory (LSTM)

LSTM merupakan varian Recurrent Neural Network (RNN) yang efektif memproses data sekuensial seperti video

gerakan tangan. Mempertahankan konteks antar frame sehingga meningkatkan akurasi pengenalan isyarat [5].

#### D. Mediapipe

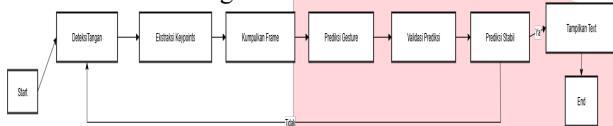
Framework dari Google untuk deteksi dan pelacakan landmark tubuh, termasuk tangan, secara real-time. Menghasilkan koordinat 21 titik tangan sebagai input model.

#### E. TensorFlow & Keras

Framework deep learning yang digunakan untuk membangun, melatih, dan mengimplementasikan model CNN-LSTM pada penelitian ini.

### III. METODE

#### A. Flowchart Perancangan Sistem Sibi

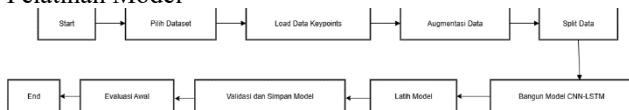


GAMBAR 2  
PERANCANGAN SISTEM

Penelitian ini dilakukan untuk merancang sistem penerjemah bahasa isyarat SIBI menjadi teks secara real-time dengan tahapan sebagai berikut:

1. Identifikasi kebutuhan penerjemahan bahasa isyarat yang akurat, responsif, dan mampu mengenali huruf, angka, serta kata.
2. Studi literatur terkait metode ekstraksi fitur dengan MediaPipe Holistic serta pengenalan gesture menggunakan CNN-LSTM.
3. Pengembangan sistem akuisisi citra melalui kamera untuk menangkap frame tangan pengguna.
4. Deteksi dan ekstraksi 21 landmark tiap tangan (dimensi 126) pada setiap frame menggunakan MediaPipe.
5. Pengumpulan 30 frame berturut-turut menjadi satu sequence sebagai representasi gerakan utuh.
6. Pemrosesan sequence menggunakan model CNN-LSTM untuk mengenali gesture dengan mempertimbangkan informasi spasial dan temporal.
7. Validasi prediksi dengan pengecekan konsistensi label dan tingkat confidence untuk menghindari perubahan hasil yang tidak stabil.
8. Penampilan hasil prediksi dalam bentuk teks di antarmuka, disertai persentase keyakinan, serta dukungan perubahan mode pengenalan (huruf, angka, kata) dan penghentian sistem secara manual.

#### B. Pelatihan Model

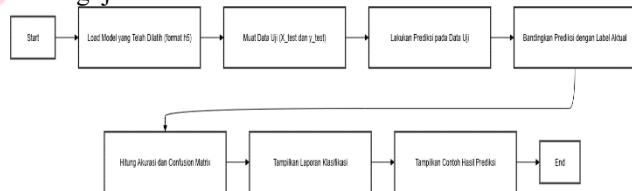


GAMBAR 3  
PELATIHAN MODEL

1. Pemilihan Dataset, proses diawali dengan memilih dataset gesture Sistem Isyarat Bahasa Indonesia (SIBI) yang mencakup huruf, angka, dan kata.

2. Pemuatan Data Keypoints, sistem memuat data koordinat 3D titik-titik tangan (x, y, z) yang diekstraksi per frame.
3. Augmentasi Data, dilakukan penambahan variasi data seperti noise dan scaling untuk meningkatkan kemampuan generalisasi model.
4. Pembagian Data, dataset dibagi menjadi data latih, validasi, dan uji untuk memastikan model dapat menggeneralisasi data baru.
5. Pembangunan Model CNN-LSTM, arsitektur dibangun dengan Conv1D untuk ekstraksi fitur spasial, diikuti LSTM untuk menangkap pola temporal, dan lapisan Dense untuk klasifikasi.
6. Pelatihan Model, model dilatih menggunakan categorical crossentropy sebagai fungsi loss dan Adam sebagai optimizer hingga mencapai performa optimal.
7. Validasi dan Penyimpanan Model, model dievaluasi dengan data validasi dan model terbaik disimpan dalam format .h5.
8. Evaluasi Akhir, performa model diuji pada data uji menggunakan metrik akurasi, precision, recall, dan F1-score.

#### C. Pengujian Model



GAMBAR 4  
PENGUJIAN MODEL

1. Start, inisialisasi awal sistem pengujian, termasuk pengecekan library, direktori kerja, dan file model .h5 yang akan diuji.
2. Load Model (.h5), memuat model CNN-LSTM yang sudah dilatih menggunakan fungsi `load_model()` Keras, memastikan arsitektur dan bobot sama dengan model pelatihan.
3. Muat Data Uji ( $X_{test}$ ,  $y_{test}$ ), mengambil data uji dengan format dan preprocessing identik seperti pelatihan untuk memastikan evaluasi objektif.
4. Prediksi Data Uji, memberikan data uji ke model untuk menghasilkan probabilitas tiap kelas, lalu menentukan label prediksi dengan `argmax`.
5. Bandingkan dengan Label Aktual, mengubah  $y_{test}$  dari one-hot encoding ke label numerik, lalu membandingkan dengan  $y_{pred}$  untuk menilai kebenaran prediksi.
6. Hitung Akurasi & Confusion Matrix, menghitung persentase prediksi benar dan menyusun matriks kesalahan untuk menganalisis performa per kelas.
7. Tampilkan Laporan Klasifikasi, menyajikan precision, recall, dan F1-score per kelas sebagai evaluasi menyeluruh.

$$\text{Akurasi} = \frac{\text{Jumlah prediksi benar}}{\text{Jumlah total prediksi}} \times 100\%$$

8. Tampilkan Contoh Hasil Prediksi, menampilkan beberapa sampel prediksi beserta confidence score, termasuk contoh salah prediksi untuk analisis.
9. End, mengakhiri proses evaluasi dan mendokumentasikan hasil untuk menentukan kelayakan model digunakan pada sistem real-time.

#### IV. HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil dan pembahasan dari sistem deteksi gesture angka dalam bahasa isyarat SIBI yang telah dikembangkan. Evaluasi kinerja dilakukan secara kuantitatif menggunakan 10% data uji yang dipisahkan secara acak dari keseluruhan dataset. Analisis mencakup penilaian objektif terhadap kemampuan model dalam melakukan klasifikasi pada data uji, serta pengujian performa sistem dalam skenario penggunaan langsung untuk mendeteksi gesture angka secara real-time.

##### A. Evaluasi Sistem Deteksi Sibi

Hasil pengujian model CNN-LSTM pada tiga kategori data, yaitu angka, huruf, dan kata, menunjukkan kinerja yang sangat optimal. Model berhasil mencapai akurasi uji sebesar 100% pada seluruh kategori dengan nilai loss yang sangat rendah, seperti ditunjukkan pada Tabel berikut.

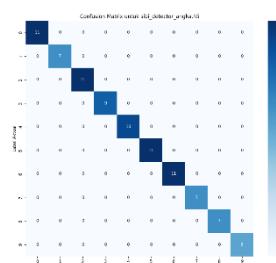
TABEL 1  
KINERJA PADA DATA UJI

Kategori	Akurasi Uji (%)	Loss Uji
Angka	100.00	0.0170
Huruf	100.00	0.0156
Kata	100.00	0.0126

Capaian ini menunjukkan bahwa model mampu mengenali pola dan karakteristik gerakan tangan SIBI pada setiap kategori secara tepat tanpa adanya kesalahan prediksi. Nilai loss yang kecil juga mengindikasikan perbedaan antara hasil prediksi dan label sebenarnya sangat minim, sehingga model memiliki kemampuan generalisasi yang sangat baik terhadap data uji.

##### B. Confusion Matrix Hasil Evaluasi Model

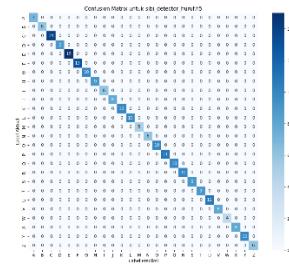
Gambar menunjukkan confusion matrix untuk ketiga model deteksi gerakan tangan SIBI yang dikembangkan. Pada confusion matrix model deteksi angka (sibi\_detector\_angka.h5), terlihat bahwa seluruh data uji pada masing-masing kelas dari 0 hingga 9 terprediksi dengan tepat, ditandai dengan nilai yang berada sepenuhnya pada diagonal utama dan tidak ada nilai di luar diagonal. Hal ini menegaskan bahwa model mampu mengenali setiap angka dengan tingkat ketepatan sempurna.



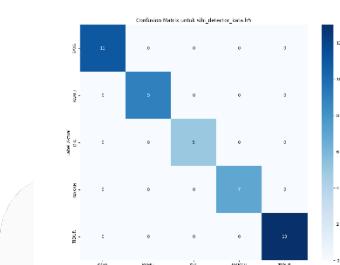
GAMBAR 5

CONFUSION MATRIX HASIL EVALUASI MODEL ANGKA

Pada confusion matrix model deteksi huruf (sibi\_detector\_huruf.h5), setiap huruf dari A hingga Z juga terkласifikasi dengan benar. Seluruh nilai prediksi berada pada diagonal utama, menunjukkan bahwa model berhasil membedakan setiap huruf tanpa kesalahan prediksi. Meski jumlah data uji per huruf bervariasi, hal ini tidak memengaruhi performa model dalam mengenali gerakan yang merepresentasikan masing-masing huruf.



GAMBAR 6  
CONFUSION MATRIX HASIL EVALUASI MODEL HURUF



GAMBAR 7  
CONFUSION MATRIX HASIL EVALUASI MODEL KATA

Sementara itu, confusion matrix model deteksi kata (sibi\_detector\_kata.h5) memperlihatkan hasil serupa, di mana kelima kata yang diuji—"SAYA", "KAMU", "DIA", "MAKAN", dan "TIDUR"—terprediksi secara sempurna pada diagonal utama tanpa adanya kesalahan klasifikasi. Keseluruhan pola pada tiga confusion matrix tersebut memperkuat hasil evaluasi numerik yang menunjukkan akurasi 100%, sekaligus membuktikan konsistensi model dalam mengenali gerakan tangan SIBI secara presisi pada semua kategori.

#### V. KESIMPULAN

Tugas Akhir ini berhasil membangun pipeline deteksi gerakan tangan bahasa isyarat SIBI menggunakan MediaPipe untuk ekstraksi fitur dan tiga model CNN-LSTM untuk mengenali gestur huruf, angka, dan kata. Model huruf dan angka memanfaatkan MediaPipe Hands, sedangkan model kata menggunakan MediaPipe Holistic. Hasil pengujian menunjukkan akurasi 100% tanpa kesalahan klasifikasi, baik pada data uji maupun pengujian *real-time* melalui kamera. Sistem ini telah memenuhi tujuan penelitian dan berpotensi dikembangkan sebagai aplikasi penerjemah bahasa isyarat otomatis berbasis komputer.

#### REFERENSI

- [1] A. A. & Suharjito, "BISINDO (Bahasa Isyarat Indonesia) Sign Language Recognition Using CNN

- and LSTM," *Advances in Science, Technology and Engineering Systems Journal*, 2020.
- [2] M. A. L. & D. Tristianto, "SIBI Alphabet Detection System Based on *Convolutional Neural Network* (CNN) Method as Learning Media," *IoTA Journal*, vol. 4, p. 1, 2024.
- [3] R. A. T. Kurniawan, "SIBI Dynamic Gesture Translation Using *MediaPipe* and LSTM," *Atlantis Press*, 2024.
- [4] HALIM, "Implementasi *MediaPipe* Hand Tracking untuk Gesture Classification Menggunakan *Dense Neural Network* pada Aplikasi 3D," 2023.
- [5] F. Zaka, "Pemahaman Mendalam dan Perhitungan *Long Short-Term Memory* (LSTM) menggunakan *Python*," GABUT-IT, 16 September 2023. [Online]. Available: <https://www.gabut-it.com/pemahaman-mendalam-dan-perhitungan-long-short-term-memory-LSTM-menggunakan-Python/>.
- [6] "Perbandingan *TensorFlow* vs *Keras* dalam *Machine learning Python*," learn.com, 26 Maret 2024. [Online]. Available: <https://stats-learn.com/id/perbandingan-TensorFlow-vs-Keras-dalam-machine-learning-Python/>.
- [7] Q. Chalik, "Mouse Tracking Tangan dengan Klasifikasi Gestur Menggunakan," *Sinov*, vol. Vol 1 No. 2, pp. 1-9, 2021.
- [8] Y. a. H. G. R. Jannah, "Implementasi *Deep learning* dalam Pengenalan Sistem," *Jurnal Tektro*, pp. 266-273, 2024.