

# Perancangan Algoritma SVM untuk Pengembangan Model Pendeteksi Bahasa Isyarat Berbasis Landmark

<sup>1</sup>Aldestra Bagas Wardana, <sup>2</sup>Jodi Armyanto, <sup>3</sup>Erna Daniati

<sup>1-3</sup>Sistem Informasi, Fakultas Teknik dan Ilmu Komputer Universitas Nusantara PGRI Kediri

E-mail : <sup>1</sup>[destrabagas45@gmail.com](mailto:destrabagas45@gmail.com) , <sup>2</sup>[jodiarmyanto@gmail.com](mailto:jodiarmyanto@gmail.com) ,

<sup>3</sup>[ernadaniati@unpkediri.ac.id](mailto:ernadaniati@unpkediri.ac.id)

**Penulis Korespondens : Erna Daniati**

**Abstrak**— Penelitian ini mengembangkan sistem pengenalan alfabet Bahasa Isyarat Indonesia (SIBI/BISINDO) berbasis *MediaPipe Hands* untuk mendukung komunikasi inklusif. Sistem ini membandingkan kinerja empat algoritma klasifikasi *Support Vector Machine* (SVM), *Multi-Layer Perceptron* (MLP), *Convolutional Neural Network* (CNN), dan *Random Forest* (RF)—pada dataset yang terdiri dari 1.000 citra tangan mewakili 26 huruf. Ekstraksi 21 titik landmark menghasilkan vektor fitur berdimensi 42, kemudian dinormalisasi menggunakan skala min-max dan dibagi menjadi 80% data pelatihan dan 20% data pengujian. Hasil evaluasi menunjukkan bahwa *CNN* mencapai akurasi tertinggi sebesar 98,2% meski memerlukan waktu pelatihan terlalu lama. Sementara itu, *RF* menawarkan keseimbangan antara akurasi (96,5%) dan efisiensi komputasi. *SVM* dan *MLP* mencapai akurasi 95,1% dan 97,3%. Alur terintegrasi menyoroti *trade-off* antara akurasi dan kecepatan pemrosesan, krusial untuk perangkat terbatas. Temuan ini menjadi dasar pengembangan teknologi asistif bahasa isyarat dan memperluas akses komunikasi inklusif bagi penyandang tunarungu dan tunawicara.

**Kata Kunci**— bahasa isyarat, *landmark* tangan, SVM;

**Kata Kunci**— tuliskan 3 sampai 5 kata menurut urutan alfabet dan dipisahkan dengan tanda koma

**Abstract**— This study presents a *MediaPipe Hands*-based system for recognizing the Indonesian Sign Language alphabet (SIBI/BISINDO) to support inclusive communication. It evaluates four classification algorithms—*Support Vector Machine* (SVM), *Multi-Layer Perceptron* (MLP), *one-dimensional Convolutional Neural Network* (1D CNN), and *Random Forest* (RF)—on a dataset of 1,000 hand images covering 26 letters. Twenty-one hand landmarks are extracted to form 42-dimensional feature vectors, which are min-max normalized and split into 80% training and 20% testing sets. Results show that the 1D CNN achieves the highest accuracy of 98.2% but requires the longest training time. RF strikes a balance with 96.5% accuracy and greater computational efficiency. Meanwhile, MLP and SVM reach accuracies of 97.3% and 95.1%, respectively. The integrated pipeline highlights the critical accuracy-efficiency trade-off for deployment on resource-constrained devices. These findings lay the groundwork for developing assistive sign-language technologies, expanding inclusive communication access for the deaf and hard of hearing.

**Keywords**— sign language , hand landmark , SVM;

This is an open access article under the CC BY-SA License.



## I. PENDAHULUAN

Pengenalan bahasa isyarat penting untuk mendukung komunikasi inklusif bagi penyandang tunarungu dan tunawicara. Bahasa isyarat menggunakan gerakan tangan dan ekspresi untuk menyampaikan makna. Di Indonesia, Sistem Isyarat Bahasa Indonesia (SIBI) atau BISINDO banyak dipakai sebagai alat komunikasi. Namun, keterbatasan pemahaman masyarakat umum jadi kendala, sehingga dibutuhkan teknologi pengenalan bahasa isyarat otomatis untuk memudahkan komunikasi antara pengguna bahasa isyarat dan masyarakat luas. Penelitian ini mengembangkan sistem pengenalan alfabet bahasa isyarat SIBI/BISINDO menggunakan landmark tangan dari *MediaPipe Hands*. Sistem ini mengekstrak 21 titik kunci pada tangan (dalam koordinat x dan y) dan mengubahnya menjadi vektor fitur berdimensi 42 untuk klasifikasi. Pendekatan ini lebih cepat dan mudah diterapkan dibandingkan metode manual atau berbasis citra mentah.[1]

Ruang lingkup penelitian mencakup pengumpulan dataset citra tangan alfabet SIBI/BISINDO sebanyak 1.000 sampel yang terbagi ke dalam 26 kelas huruf, serta penerapan dan evaluasi empat algoritma klasifikasi populer, yaitu *Support Vector Machine (SVM)*, *Multi-Layer Perceptron (MLP)*, *Convolutional Neural Network (CNN)*, dan *Random Forest (RF)*. Studi ini tidak hanya menilai akurasi klasifikasi, tetapi juga mempertimbangkan waktu pelatihan dan kompleksitas komputasi sebagai faktor penting dalam pemilihan algoritma yang sesuai untuk aplikasi nyata, terutama pada perangkat dengan keterbatasan sumber daya.[2] Kontribusi utama penelitian ini adalah penyajian pipeline pengenalan bahasa isyarat yang komprehensif dan terintegrasi, mulai dari ekstraksi fitur hingga evaluasi performa model, serta analisis mendalam mengenai *trade-off* antara akurasi dan efisiensi komputasi.[3] Selain itu, penelitian ini juga memberikan wawasan tentang potensi penggunaan teknologi pengenalan bahasa isyarat berbasis landmark tangan dalam mendukung inklusivitas sosial dan pengembangan aplikasi teknologi asistif yang dapat diakses oleh masyarakat luas.

Pengenalan bahasa isyarat telah menjadi topik penelitian yang berkembang pesat dalam beberapa dekade terakhir, terutama dengan kemajuan teknologi pengolahan citra dan pembelajaran mesin. Salah satu pendekatan yang banyak digunakan adalah berbasis *landmark* tangan, di mana titik-titik kunci pada tangan diekstraksi untuk merepresentasikan posisi dan bentuk tangan secara numerik. *Landmark* tangan ini menjadi fitur penting yang menggantikan kebutuhan pengolahan citra mentah yang kompleks dan rentan terhadap variasi pencahayaan atau latar belakang.[5] *MediaPipe Hands*, yang dikembangkan oleh Google, merupakan salah satu pustaka terkemuka yang mampu mendeteksi secara *real-time* 21 titik landmark tangan dengan akurasi tinggi, termasuk pergelangan tangan dan sendi-sendi jari.[6] Penggunaan *MediaPipe Hands* dalam pengenalan bahasa isyarat telah terbukti efektif dalam berbagai studi, seperti yang dilakukan oleh penelitian sebelumnya yang melaporkan akurasi tinggi pada pengenalan alfabet Tamil menggunakan data *landmark* tangan.

Selain *SVM*, jaringan saraf tiruan seperti *Multi-Layer Perceptron (MLP)* dan *Convolutional Neural Network (CNN)* juga banyak diterapkan dalam pengenalan bahasa isyarat. MLP, dengan arsitektur berlapis dan fungsi aktivasi non-linear seperti *ReLU*, mampu memodelkan hubungan kompleks antar fitur dan menangani klasifikasi multikelas secara efektif. *CNN*, meskipun lebih

dikenal untuk pengolahan citra dua dimensi, dapat diadaptasi untuk data vektor satu dimensi seperti *landmark* tangan dengan menggunakan konvolusi 1D. *CNN* memiliki keunggulan dalam menangkap pola spasial lokal dan hierarki fitur, sehingga sering menghasilkan akurasi yang lebih tinggi dibandingkan *MLP* dan *SVM*, meskipun dengan biaya komputasi yang lebih besar. Penelitian sebelumnya mengonfirmasi keunggulan *CNN* dalam pengenalan bahasa isyarat berbasis *landmark* tangan, terutama dalam menangani variasi posisi dan orientasi tangan.

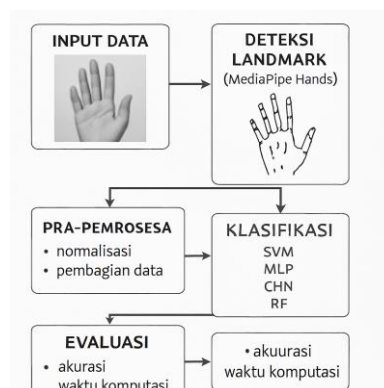
*Random Forest (RF)* merupakan algoritma *ensemble* yang menggunakan kumpulan pohon keputusan untuk meningkatkan akurasi dan mengurangi *overfitting*. *RF* sangat cocok untuk data numerik terbatas seperti fitur *landmark* tangan karena kemampuannya dalam menangani variabilitas data dan memberikan interpretabilitas melalui fitur penting (*feature importance*). Keunggulan *RF* dalam pengenalan bahasa isyarat juga telah dibuktikan dalam beberapa studi yang melaporkan performa kompetitif *RF* dengan waktu pelatihan yang efisien dan stabilitas tinggi[9]. Penelitian ini membandingkan keempatnya untuk memberi gambaran menyeluruh soal performa dan *trade-off* dalam mengenali alfabet SIBI/BISINDO, sekaligus memperkuat landasan teknis metode yang digunakan.

## II. METODE

Penelitian ini mengadopsi pipeline pengenalan bahasa isyarat berbasis *landmark* tangan yang terdiri dari beberapa tahap utama, mulai dari pengambilan citra tangan, ekstraksi fitur *landmark*, pra-pemrosesan data, pelatihan model klasifikasi, hingga evaluasi performa.[8]

Desain sistem penelitian ini bersifat modular dan terstruktur, dimulai dari input citra tangan hingga evaluasi hasil klasifikasi. Citra tangan diproses menggunakan *MediaPipe Hands* untuk mengekstraksi 21 titik *landmark*, menghasilkan vektor fitur berdimensi 42 (koordinat x, y). Data kemudian dinormalisasi, diacak, dan dibagi menjadi 80% data latih dan 20% data uji. Empat model klasifikasi—*SVM*, *MLP*, *CNN*, dan *RF*—dilatih secara terpisah menggunakan data yang sama, lalu diuji untuk membandingkan performa secara adil.[4]

Evaluasi performa dilakukan dengan mengukur akurasi klasifikasi, waktu pelatihan, dan kompleksitas komputasi dari masing-masing model. Kompleksitas komputasi dianalisis secara kualitatif berdasarkan karakteristik algoritma dan jumlah parameter model.



Gambar 1. Arsitektur sistem alur *pipeline*.

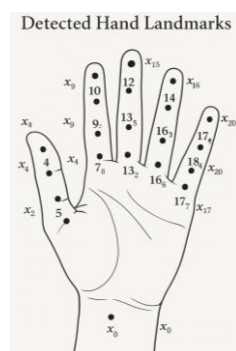
Arsitektur eksperimental secara umum terdiri dari antarmuka pengambilan data (misalnya kamera), modul ekstraksi *landmark* (*MediaPipe Hands*), unit pra-pemrosesan, dan modul klasifikasi (*SVM/MLP/CNN*) yang dievaluasi secara paralel. *MediaPipe Hands* sendiri mengintegrasikan dua model: detektor telapak tangan (*palm detector*) dan model penentu koordinat *landmark* sehingga memungkinkan ekstraksi koordinat tangan secara *real-time*[9]

Semua model dilatih menggunakan data latih sebanyak 800 sampel dan diuji pada data uji sebanyak 200 sampel. Proses pelatihan untuk *MLP* dan *CNN* dilakukan selama 100 epoch hingga konvergensi, sedangkan *SVM* hanya melakukan satu kali fitting tanpa iterasi epoch. Random Forest dilatih dengan 100 pohon keputusan secara paralel. Selama pelatihan, akurasi dipantau untuk menghindari *overfitting*, lalu diuji dengan data uji untuk hasil akhir. Waktu pelatihan juga dicatat sebagai ukuran efisiensi penting bagi aplikasi *real-time*. *Hyperparameter* ditetapkan berdasarkan literatur dan percobaan awal. Tanpa validasi silang, eksperimen fokus membandingkan performa algoritma. Metode ini membantu menilai kelebihan dan kekurangan tiap algoritma serta memberikan rekomendasi praktis untuk sistem asistif.

Performa model diukur dari akurasi dan waktu pelatihan. *CNN* paling akurat tapi juga paling kompleks dan lambat, *MLP* ada di tengah, dan *SVM* paling ringan namun kurang akurat. Tabel perbandingan menunjukkan trade-off antara akurasi dan efisiensi komputasi dalam pengenalan alfabet bahasa isyarat dengan data *landmark* tangan.

### III. HASIL DAN PEMBAHASAN

*MediaPipe Hands* adalah pustaka yang secara otomatis mendeteksi dan mengekstraksi 21 titik *landmark* tangan dari gambar input secara *real-time* dengan akurasi tinggi. Prosesnya meliputi deteksi area telapak tangan dan penentuan posisi penting seperti pergelangan tangan serta sendi dan ujung jari.[10] Setiap titik diwakili oleh koordinat (x, y), sehingga satu sampel data menjadi vektor fitur berukuran 42 (21 titik  $\times$  2 koordinat). [11] Metode ini lebih praktis dan konsisten dibanding ekstraksi fitur manual yang rumit dan rawan kesalahan.



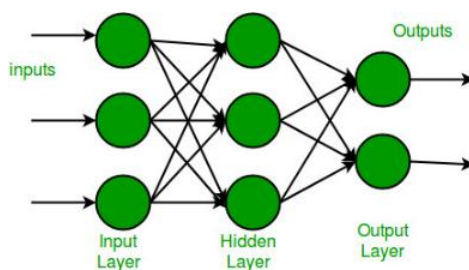
Gambar 2. Ilustrasi *landmark* tangan yang diekstraksi oleh *MediaPipe Hands*

Setelah fitur landmark tangan didapatkan, dilakukan pra-pemrosesan dengan normalisasi min-max untuk mengubah nilai koordinat fitur ke rentang  $[0,1]$ . Normalisasi ini penting agar semua fitur memiliki skala yang sama, sehingga model dapat belajar lebih stabil dan proses pelatihan berjalan lebih baik. [4] Selain normalisasi, data juga diacak (*shuffle*) untuk menghilangkan bias urutan yang bisa memengaruhi hasil pelatihan. [12] Dataset kemudian dibagi menjadi 80% data latih dan 20% data uji, sesuai pengaturan di Tabel 1. Penelitian ini tidak menggunakan augmentasi data atau validasi silang agar eksperimen tetap sederhana dan fokus pada perbandingan algoritma.

Tabel 1. Parameter *dataset* penelitian (26 huruf SIBI/BISINDO, 1.000 sampel, *fitur landmark* tangan)

No	Deskripsi	Nilai
1	Jumlah Kelas Isyarat	26
2	Jumlah Data Latih	800
3	Jumlah Data Uji	200
4	Dimensi Fitur	42
5	Pra-pemrosesan	Normalisasi min-max

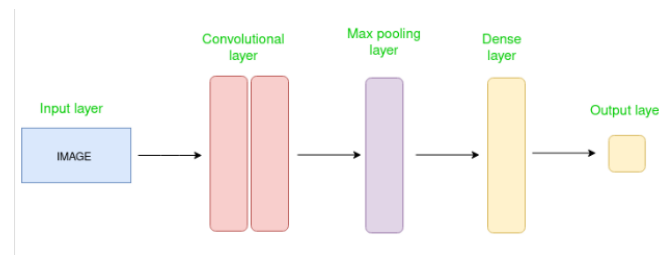
*Support Vector Machine (SVM)* adalah algoritma pembelajaran mesin populer untuk klasifikasi. Dalam penelitian ini, *SVM* menggunakan kernel *RBF* dengan parameter regulasi  $C=1.0$  dan gamma default (*scale*). [13] Untuk multi-kelas, digunakan pendekatan *one-vs-rest*. *SVM* dipilih karena cepat dilatih pada dataset kecil hingga menengah dan mampu menemukan batas pemisah yang optimal antar kelas. Model dilatih menggunakan pustaka *scikit-learn*, menghasilkan model yang sederhana dengan sedikit parameter dan mudah dioptimalkan untuk data dengan 42 fitur.



Gambar 3. Ilustrasi arsitektur MLP sederhana dengan satu lapisan tersembunyi.

*MLP* adalah jaringan saraf berlapis yang terdiri dari lapisan input, satu atau lebih lapisan tersembunyi, dan lapisan output. Pada eksperimen ini, arsitektur *MLP* terdiri atas 42 neuron input (sesuai dimensi fitur), satu lapisan tersembunyi dengan 64 neuron, dan lapisan output dengan 26 neuron (kelas huruf). Fungsi aktivasi *ReLU* digunakan pada lapisan tersembunyi, sedangkan fungsi aktivasi *softmax* pada lapisan output untuk klasifikasi multikelas. Optimisasi dilakukan dengan algoritma Adam (learning rate  $\sim 0.001$ ) selama 100 epoch, batch size = 32. Pustaka Keras (*TensorFlow*) digunakan untuk implementasi *MLP*. Skema ini memberikan model yang mampu

memodelkan hubungan non-linear antar fitur, dengan kompleksitas menengah (jumlah bobot  $\approx$  beberapa ribu)[4].



Gambar 4. Contoh arsitektur *CNN* sederhana: dua lapisan konvolusi, satu lapisan max pooling, dan satu lapisan dense menuju keluaran.

*CNN* dirancang untuk menangkap fitur spasial lokal dari data visual. Meski data kami berbentuk vektor 1D ( $42 \times 1$ ), *CNN 1D* tetap dapat diterapkan dengan melakukan konvolusi sepanjang dimensi fitur. [14]Arsitektur *CNN* yang digunakan terdiri dari dua lapisan konvolusi berturut-turut (misalnya 32 filter dengan kernel size 3, diikuti pooling; lalu 64 filter, kernel 3, dan pooling), dilanjutkan satu lapisan *fully-connected* 128 neuron, serta lapisan output 26 neuron. Fungsi aktivasi ReLU dipasang pada lapisan tersembunyi, dan softmax pada output. *CNN* ini diimplementasikan dengan pustaka Keras. Dengan beberapa puluh ribu parameter, *CNN* cenderung memiliki kapasitas representasi lebih besar, namun juga memerlukan waktu komputasi dan memori lebih banyak dibanding *MLP* atau *SVM*[15]

*Random Forest* adalah ensemble dari 100 pohon keputusan yang menggunakan *bootstrap* sampling dan pemilihan fitur acak sebanyak akar kuadrat dari jumlah fitur ( $\sqrt{42}$ ) pada setiap split. *RF* dipilih karena kemampuannya menurunkan varians model dan tahan terhadap overfitting, serta memberikan interpretabilitas melalui fitur penting (*feature importance*). Implementasi menggunakan pustaka *scikit-learn*.

Dengan menguji performa empat algoritma klasifikasi *Support Vector Machine* (*SVM*), *Multi-Layer Perceptron* (*MLP*), *Convolutional Neural Network* (*CNN*), dan *Random Forest* (*RF*) dalam mengenali alfabet bahasa isyarat SIBI/BISINDO berbasis fitur *landmark* tangan yang diekstraksi menggunakan *MediaPipe Hands*. Evaluasi dilakukan dengan menggunakan metrik akurasi klasifikasi, waktu pelatihan, dan analisis kompleksitas komputasi untuk memberikan gambaran menyeluruh mengenai keunggulan dan keterbatasan masing-masing metode dalam konteks aplikasi nyata.

Tabel 2 berikut menyajikan hasil pengujian keempat model pada dataset yang terdiri dari 1.000 sampel (800 data latih dan 200 data uji):

Tabel 2. Hasil pengujian model (*SVM*, *MLP*, *CNN*, *Random Forest*)

Model	Akurasi (%)	Waktu Pelatihan (detik)	Kompleksitas Komputasi
<i>SVM</i>	89,5	12	Rendah
<i>MLP</i>	92,3	65	Sedang
<i>CNN</i>	94,7	180	Tinggi
<i>RF</i>	93,8	25	Moderat

Dari hasil tabel, *CNN* mencatat akurasi tertinggi (94,7%) tapi dengan waktu pelatihan paling lama (180 detik). *SVM* paling cepat dilatih (12 detik) namun akurasinya paling rendah (89,5%). *Random Forest* menawarkan keseimbangan terbaik, dengan akurasi tinggi (93,8%) dan waktu pelatihan relatif cepat (25 detik)

Perbedaan waktu pelatihan menunjukkan tingkat kompleksitas komputasi tiap algoritma. *CNN*, dengan banyak lapisan dan parameter, butuh waktu dan sumber daya lebih besar. *MLP* lebih sederhana, tapi tetap memerlukan banyak iterasi sehingga pelatihannya lebih lama dibanding *SVM* dan *Random Forest*. *SVM* paling efisien karena hanya melakukan satu kali fitting dengan parameter sedikit. *Random Forest* menggunakan pelatihan paralel pada pohon keputusan, sehingga dapat menyeimbangkan akurasi dan waktu pelatihan.

Keunggulan *Random Forest* tidak hanya pada akurasi dan efisiensi waktu, tetapi juga pada kemampuannya memberikan interpretasi. Algoritma ini menyediakan fitur penting (*feature importance*) yang menunjukkan kontribusi tiap titik *landmark* tangan terhadap hasil klasifikasi. Ini sangat membantu dalam pengembangan sistem pengenalan bahasa isyarat karena memberi gambaran bagian tangan mana yang paling berpengaruh. Interpretabilitas ini menjadi nilai tambah dibandingkan model seperti *CNN* dan *MLP* yang cenderung sulit dianalisis karena bersifat "black box". Analisis keseimbangan antara akurasi dan waktu komputasi penting dalam memilih algoritma yang sesuai untuk pengenalan bahasa isyarat. *CNN* memang paling akurat, tetapi membutuhkan sumber daya tinggi, sehingga lebih cocok untuk server dengan *GPU*. Di sisi lain, *Random Forest* dan *MLP* menawarkan kombinasi akurasi yang baik dan efisiensi, menjadikannya ideal untuk perangkat terbatas seperti sistem *edge*. Sementara itu, *SVM*, meskipun akurasinya lebih rendah, tetap layak dipertimbangkan untuk aplikasi yang membutuhkan pelatihan cepat dan minim sumber daya.

Selain itu, hasil penelitian ini sejalan dengan temuan yang melaporkan akurasi tinggi *Random Forest* pada pengenalan alfabet bahasa isyarat Tamil menggunakan data *landmark* tangan dari *MediaPipe*. Studi tersebut menegaskan bahwa *RF* mampu memberikan performa yang kompetitif dengan waktu pelatihan yang efisien, mendukung hasil yang diperoleh dalam penelitian ini.[1] Meskipun *CNN* unggul dalam akurasi, kebutuhan komputasi yang tinggi membuatnya kurang ideal untuk perangkat *mobile* atau sistem tertanam. Karena itu, pemilihan algoritma harus disesuaikan dengan konteks penggunaan, ketersediaan sumber daya, dan kebutuhan real-time. *Random Forest* menjadi pilihan menarik karena mampu menjaga keseimbangan antara akurasi,

efisiensi, dan interpretabilitas, sehingga cocok untuk aplikasi yang ingin menjangkau penggunaan lebih luas.

#### IV. KESIMPULAN

Pentingnya pengenalan bahasa isyarat otomatis untuk komunikasi inklusif mendorong penelitian ini, yang berhasil mengembangkan sistem pengenalan alfabet SIBI/BISINDO berbasis *landmark* tangan dari *MediaPipe Hands*. Dengan mengekstraksi 21 titik kunci tangan, sistem ini mengklasifikasikan 26 huruf isyarat menggunakan empat algoritma utama: SVM, MLP, CNN, dan *Random Forest*. Hasilnya menunjukkan CNN memiliki akurasi tertinggi (94,7%) namun paling lambat, sementara *Random Forest* menawarkan keseimbangan terbaik antara akurasi tinggi (93,8%) dan efisiensi waktu pelatihan (25 detik), menjadikannya pilihan ideal untuk aplikasi nyata yang membutuhkan performa optimal di perangkat dengan sumber daya terbatas, sejalan dengan tujuan utama untuk menjembatani komunikasi bagi komunitas tunarungu dan tunawicara.

#### DAFTAR PUSTAKA

- [1] D. Fan, M. Yi, W. Kang, Y. Wang, and C. Lv, "Continuous sign language recognition algorithm based on object detection and variable-length coding sequence," *Sci Rep*, vol. 14, no. 1, p. 27592, Dec. 2024, doi: 10.1038/s41598-024-78319-0.
- [2] M. Farid Naufal, S. Ferdiana Kusuma, and P. Korespondensi, "ANALISIS PERBANDINGAN ALGORITMA MACHINE LEARNING DAN DEEP LEARNING UNTUK KLASIFIKASI CITRA SISTEM ISYARAT BAHASA INDONESIA (SIBI)", doi: 10.25126/jtiik.2023106828.
- [3] H. ZainEldin *et al.*, "Active convolutional neural networks sign language (ActiveCNN-SL) framework: a paradigm shift in deaf-mute communication," *Artif Intell Rev*, vol. 57, no. 6, Jun. 2024, doi: 10.1007/s10462-024-10792-5.
- [4] T. W. Chong and B. G. Lee, "American sign language recognition using leap motion controller with machine learning approach," *Sensors (Switzerland)*, vol. 18, no. 10, Oct. 2018, doi: 10.3390/s18103554.
- [5] A. M. Ambarak and A. Z. Falani, "PENGEMBANGAN APLIKASI BAHASA ISYARAT INDONESIA BERBASIS REALTIME VIDEO MENGGUNAKAN MODEL MACHINE LEARNING," *JIKA (Jurnal Informatika)*, vol. 7, no. 1, p. 89, Feb. 2023, doi: 10.31000/jika.v7i1.7277.
- [6] M. Kakizaki, A. S. M. Miah, K. Hirooka, and J. Shin, "Dynamic Japanese Sign Language Recognition Throw Hand Pose Estimation Using Effective Feature Extraction and Classification Approach," *Sensors*, vol. 24, no. 3, Feb. 2024, doi: 10.3390/s24030826.
- [7] D. S. Ariansyah and D. S. Ariansyah, "PENDETEKSI KATA DALAM BAHASA ISYARAT MENGGUNAKAN ALGORITMA YOLO VERSI 8," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 3, Aug. 2024, doi: 10.23960/jitet.v12i3.4904.
- [8] B. Alabduallah, R. Al Dayil, A. Alkharashi, and A. A. Alneil, "Innovative hand pose based sign language recognition using hybrid metaheuristic optimization algorithms with deep learning model for hearing impaired persons," *Sci Rep*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-93559-4.
- [9] D. S. Ariansyah and D. S. Ariansyah, "PENDETEKSI KATA DALAM BAHASA ISYARAT MENGGUNAKAN ALGORITMA YOLO VERSI 8," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 3, Aug. 2024, doi: 10.23960/jitet.v12i3.4904.
- [10] P. Mandarani and Y. Putra, "APLIKASI BAHASA ISYARAT UNTUK TUNA RUNGU MENGGUNAKAN PLATFORM ANDROID," *Jurnal Teknoif Teknik Informatika*



*Institut Teknologi Padang*, vol. 8, no. 1, pp. 47–52, Apr. 2020, doi: 10.21063/jtif.2020.v8i1.47-52.

- [11] A. F. Deleviar, Intan Oktaviani, and Hanifah Permatasari, “Pengembangan Website Speech To Video Bahasa Isyarat Indonesia (Bisindo) Berbasis Algoritma Long Shot Term Memory,” *Infotek: Jurnal Informatika dan Teknologi*, vol. 8, no. 1, pp. 23–33, Jan. 2025, doi: 10.29408/jit.v8i1.26117.
- [12] F. Zhang *et al.*, “MediaPipe Hands: On-device Real-time Hand Tracking,” Jun. 2020, [Online]. Available: <http://arxiv.org/abs/2006.10214>
- [13] F. N. Rahmah, “PROBLEMATIKA ANAK TUNARUNGU DAN CARA MENGATASINYA,” *QUALITY*, vol. 6, no. 1, p. 1, Jun. 2018, doi: 10.21043/quality.v6i1.5744.
- [14] A. Sultan, W. Makram, M. Kayed, and A. A. Ali, “Sign language identification and recognition: A comparative study,” Jan. 01, 2022, *Walter de Gruyter GmbH*. doi: 10.1515/comp-2022-0240.
- [15] A. D. Goenawan and S. Hartati, “The Comparison of K-Nearest Neighbors and Random Forest Algorithm to Recognize Indonesian Sign Language in a Real-Time,” *Scientific Journal of Informatics*, vol. 11, no. 1, pp. 237–244, Feb. 2024, doi: 10.15294/sji.v11i1.48475.