

## Arsitektur Hybrid Pengenalan Alfabet Bisindo: Mediapipe Hands Dan Mlp Pada Web Real-Time

Muhammad Alif Nur Firdaus\*, Tri Listyorini, Endang Supriyati  
Program Studi Teknik Informatika, Fakultas Teknik,  
Universitas Muria Kudus, Indonesia

Diterima: 25 Desember 2025 | Revisi: 30 Januari 2026 | Diterbitkan: 27 Februari 2026

DOI: <https://doi.org/10.33005/scan.v21i1.5582>

### ABSTRAK

Bahasa Isyarat Indonesia (BISINDO) digunakan luas oleh komunitas Tuli, namun media pembelajaran alfabet yang interaktif dan mudah diakses masih terbatas. Penelitian ini mengusulkan aplikasi web pembelajaran alfabet BISINDO yang mengenali gestur tangan secara real-time dengan pendekatan ringan berbasis hand landmark. Kebaruan penelitian terletak pada integrasi alur ekstraksi landmark MediaPipe Hands dan klasifikasi Multilayer Perceptron (MLP) yang dieksekusi sepenuhnya di browser menggunakan TensorFlow.js, serta augmentasi berbasis landmark untuk meningkatkan robustnes. Dataset citra alfabet A–Z diekstraksi menjadi 21 titik landmark (x, y, z), dinormalisasi terhadap wrist, diskalakan dengan jarak maksimum, dan dibentuk menjadi vektor fitur 126 dimensi dengan padding. Augmentasi meliputi horizontal flip, rotasi acak, dan Gaussian noise 0,02. Model MLP mencapai akurasi 97,77% pada data uji. Uji kegunaan menggunakan System Usability Scale (SUS) menghasilkan skor rata-rata 70,53 menunjukkan aplikasi cukup baik dan dapat digunakan sebagai media belajar mandiri.

Kata Kunci: BISINDO, MediaPipe Hands, TensorFlow.js, Multilayer Perceptron.

### Real-Time Web-Based BISINDO Alphabet Recognition Using MediaPipe and MLP

### ABSTRACT

Indonesian Sign Language (BISINDO) is widely used by the Deaf community, yet interactive and accessible alphabet-learning media remain limited. This study proposes a real-time web-based BISINDO alphabet learning application using a lightweight hand-landmark pipeline. The novelty lies in integrating MediaPipe Hands landmark extraction with an Multilayer Perceptron (MLP) classifier fully executed in the browser via TensorFlow.js, supported by landmark-based augmentation to improve robustness. Static A–Z images are processed to extract 21 hand landmarks (x, y, z), normalized relative to the wrist, scaled by the maximum landmark distance, and arranged into a fixed 126-dimensional feature vector with padding. Augmentation includes horizontal flipping, random rotation, and Gaussian noise injection 0.02. The proposed MLP achieves 97.77% classification accuracy on the test set. Usability evaluation using the System Usability Scale (SUS) yields an average score of 70.53 indicating fairly good usability. Overall, the system enables immediate feedback through frame-by-frame processing and can support self-paced BISINDO alphabet learning.

Keywords: BISINDO, MediaPipe Hands, TensorFlow.js, Multilayer Perceptron

### How to Cite:

Firdaus, M. A. N., Listyorini, T., & Supriyati, E. (2026). Arsitektur Hybrid Pengenalan Alfabet Bisindo: Mediapipe Hands Dan Mlp Pada Web Real-Time. *Scan : Jurnal Teknologi Informasi dan Komunikasi*, 21(1), 11-21.

### \*Corresponding Author:

Email : alipnf@proton.me

Alamat : Jl. Lingkar Utara UMK, Gondangmanis,  
Bae, Kudus, Jawa Tengah 59327



This article is published under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

## PENDAHULUAN

Bahasa isyarat merupakan sarana komunikasi utama bagi penyandang Tuli dan gangguan pendengaran [1]. Di Indonesia, Bahasa Isyarat Indonesia (BISINDO) berkembang secara natural di komunitas Tuli dan lebih umum digunakan dalam komunikasi sehari-hari [2]. Dalam konteks inklusi sosial, kemampuan masyarakat umum memahami BISINDO penting untuk membangun komunikasi yang setara [3].

Namun, akses pembelajaran BISINDO bagi masyarakat umum masih terbatas [4]. Media pembelajaran yang tersedia umumnya berupa buku, poster, atau video yang bersifat satu arah dan tidak memberi umpan balik ketika pengguna mempraktikkan gestur [5]. Akibatnya, pengguna tidak mengetahui apakah bentuk isyarat yang dilakukan sudah benar [6] [7].

Perkembangan *computer vision* dan *machine learning* memungkinkan pembelajaran yang lebih interaktif. Banyak pendekatan pengenalan gestur menggunakan citra mentah, tetapi cenderung lebih berat untuk dijalankan langsung pada perangkat pengguna [8]. Alternatif yang lebih ringan adalah memanfaatkan *hand landmark* (titik kunci), sehingga masukan model berupa koordinat numerik yang jauh lebih kecil dibanding citra [9].

Agar model dapat berjalan langsung di browser, penelitian ini menggunakan TensorFlow.js sebagai kerangka kerja *machine learning* di web, dan *MediaPipe Hands* sebagai modul ekstraksi *landmark* tangan. Untuk proses klasifikasi alfabet A–Z digunakan *Multilayer Perceptron* (MLP), yaitu jaringan saraf *feed-forward* yang cocok untuk masukan vektor berdimensi tetap. Selain itu, Uji kegunaan dilakukan dengan metode *System Usability Scale* (SUS) untuk mengukur persepsi pengguna terhadap kemudahan penggunaan aplikasi. (Lugaresi dkk., 2019).

Rumusan masalah penelitian ini berfokus pada bagaimana merancang sistem pengenalan alfabet BISINDO yang ringan, *real-time*, mudah diakses melalui browser, dan memberikan umpan balik langsung. Tujuan penelitian ini adalah membangun *pipeline* pengenalan alfabet BISINDO berbasis *landmark*, melatih model MLP untuk klasifikasi 26 huruf alfabet, mengintegrasikan model ke aplikasi web menggunakan TensorFlow.js, dan mengevaluasi performa model serta kegunaan sistem menggunakan SUS [11] [12].

## METODE PENELITIAN

Metode penelitian ini disusun untuk membangun sistem pengenalan alfabet BISINDO yang ringan dan dapat berjalan di browser. Tahapan penelitian meliputi perancangan arsitektur sistem, penyusunan *dataset*, ekstraksi *landmark* dan pra-pemrosesan fitur, pelatihan model klasifikasi MLP, ekspor model ke TensorFlow.js serta integrasi pada aplikasi web, evaluasi performa model, dan uji kegunaan sistem menggunakan SUS.

### Perancangan Arsitektur Sistem

Arsitektur sistem dibagi menjadi dua proses besar, yaitu pelatihan model (offline) dan inferensi (online) di browser. Pada proses pelatihan, *dataset* citra alfabet BISINDO diproses menggunakan *MediaPipe Hands* untuk mengekstraksi 21 titik *landmark*. Koordinat *landmark* diolah dan disusun menjadi vektor fitur berdimensi tetap sebagai masukan model MLP. Setelah performa validasi terbaik diperoleh, model disimpan dan diekspor ke format TensorFlow.js [13] [14].

Pada proses inferensi, pengguna mengakses aplikasi web dan memberikan izin kamera. Setiap frame video diproses oleh *MediaPipe Hands* untuk memperoleh *landmark*. Fitur dibentuk dengan prosedur yang sama seperti tahap pelatihan, lalu diklasifikasikan oleh model MLP yang berjalan pada TensorFlow.js. Hasil prediksi ditampilkan pada antarmuka web [15].

Istilah real-time pada penelitian ini merujuk pada pemrosesan per frame di sisi klien (browser), yaitu alur *webcam* → *MediaPipe Hands* → pembentukan fitur → prediksi TensorFlow.js → tampilan hasil yang berjalan berulang sehingga pengguna memperoleh umpan balik prediksi secara langsung saat melakukan perubahan gestur. Pengukuran kuantitatif latensi/FPS tidak menjadi fokus penelitian ini dan dapat dijadikan pengujian lanjutan.

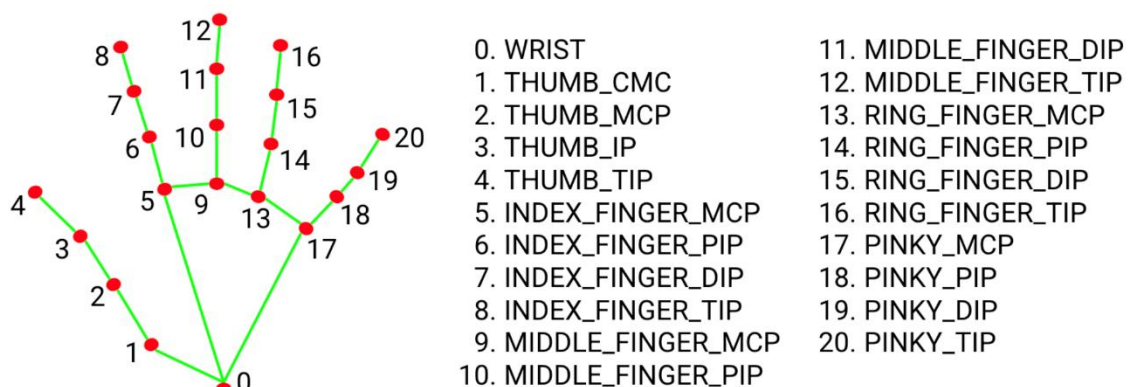
### Dataset

Dataset penelitian berupa citra alfabet BISINDO A–Z yang dikumpulkan sebagai gambar statis dengan variasi kondisi wajar seperti variasi jarak, sudut pengambilan terbatas, dan pencahayaan. Latar belakang dibuat relatif sederhana untuk menjaga kestabilan deteksi tangan. Setiap citra diproses menggunakan *MediaPipe Hands* untuk memastikan *landmark* dapat diekstraksi. Sampel yang tidak menghasilkan *landmark* (misalnya *blur* berat, tangan tidak terdeteksi, atau *occlusion* signifikan) tidak digunakan agar fitur konsisten. Variasi data ditingkatkan melalui augmentasi (*flip*, rotasi, dan *noise*) pada tahap pembentukan fitur [16]. Data dibagi menggunakan skema *train/test* 80/20 secara *stratified* sehingga distribusi kelas pada data latih dan uji tetap sebanding. Pada pengujian ini, total data uji adalah 1657 sampel.

### Ekstraksi Landmark

*MediaPipe Hands* digunakan sebagai modul ekstraksi fitur karena mampu mendeteksi tangan dan menghasilkan titik kunci yang stabil. Untuk setiap tangan, *MediaPipe* menghasilkan 21 titik yang merepresentasikan struktur pergelangan dan ruas jari. Setiap titik memiliki koordinat (x, y, z) pada sistem koordinat relatif citra [17] [18].

Keuntungan penggunaan *landmark* adalah ukuran input menjadi kecil (koordinat numerik), sehingga model klasifikasi lebih ringan dibanding pendekatan berbasis citra mentah serta lebih sesuai untuk deployment di browser. *Model hand landmark MediaPipe Hands* ditunjukkan pada Gambar 1.



**Gambar 1.** hand landmark model

Sumber: Google AI Edge, *Hand landmarks detection guide*

**Tabel 1**  
**Arsitektur model MLP**

| Lapisan  | Neuron | Aktivasi | Keterangan                   |
|----------|--------|----------|------------------------------|
| Input    | 126    | -        | Vektor fitur <i>landmark</i> |
| Hidden 1 | 128    | Relu     | BatchNorm + Dropout 0,3      |
| Hidden 2 | 64     | Relu     | BatchNorm + Dropout 0,3      |
| Hidden 3 | 32     | Relu     | Dropout 0,2                  |
| Output   | 26     | Softmax  | Probabilitas kelas A–Z       |

Sumber: Hasil penelitian

### Preprocessing Data

Tahap *preprocessing* dilakukan untuk menghasilkan representasi fitur *landmark* yang stabil dan konsisten sebelum digunakan sebagai masukan model. Koordinat *landmark* terlebih dahulu dinormalisasi secara relatif terhadap titik pergelangan tangan (*wrist*) sehingga fitur lebih tahan terhadap pergeseran posisi tangan di dalam frame kamera. Setelah itu dilakukan normalisasi skala dengan membagi koordinat terhadap nilai jarak maksimum titik *landmark* dari *wrist*, sehingga pengaruh perbedaan ukuran tangan maupun jarak pengguna terhadap kamera dapat dikurangi.

Koordinat yang telah dinormalisasi kemudian disusun menjadi vektor satu dimensi (*flattening*). Karena *MediaPipe* menghasilkan 21 titik dan masing-masing memiliki tiga koordinat (x, y, z), maka satu tangan direpresentasikan menjadi 63 fitur. Untuk menjaga ukuran masukan model selalu tetap dan membuka kemungkinan penggunaan dua tangan, sistem membentuk vektor fitur berdimensi 126 dengan menggabungkan fitur dari dua tangan. Jika hanya satu tangan yang terdeteksi, bagian fitur tangan kedua diisi nol (*padding*) sehingga seluruh sampel memiliki dimensi input yang sama [14].

Untuk meningkatkan kemampuan generalisasi, diterapkan *augmentasi* berbasis *landmark* pada tahap pembentukan fitur, meliputi pembalikan horizontal (*horizontal flip*), rotasi acak untuk toleransi variasi orientasi tangan, serta penambahan *Gaussian noise* dengan standar deviasi 0,02 untuk mensimulasikan *jitter* deteksi *landmark* dan getaran alami tangan pengguna. Label huruf A–Z diubah menjadi label numerik dan dikonversi ke bentuk *one-hot encoding* berdimensi 26 agar sesuai untuk klasifikasi multikelas pada keluaran *softmax*.

### Model MLP, Pelatihan, dan Evaluasi

Model klasifikasi menggunakan *Multilayer Perceptron* (MLP) karena input berupa vektor numerik berdimensi tetap (126) [19]. Arsitektur model terdiri dari tiga lapisan tersembunyi dengan aktivasi *ReLU* dan lapisan keluaran dengan aktivasi *softmax* untuk menghasilkan probabilitas 26 kelas huruf. *Batch Normalization* digunakan untuk membantu stabilitas pelatihan, sedangkan *Dropout* digunakan untuk mengurangi *overfitting*. Rincian arsitektur ditunjukkan pada Tabel 1.

Pelatihan dilakukan dengan fungsi kerugian *categorical cross-entropy* dan *optimizer* Adam, epoch maksimum 100, *batch size* 32, serta *early stopping* (patience 15) berdasarkan *validation loss*. Evaluasi kinerja model dilakukan dengan menghitung akurasi, serta metrik *precision*, *recall*, dan F1-score pada masing-masing kelas, serta *macro-average* dan *weighted-average* sebagai ringkasan performa. Analisis kesalahan

dilakukan menggunakan *confusion matrix* untuk melihat pola kekeliruan prediksi antar huruf [20] [21].

## HASIL DAN PEMBAHASAN

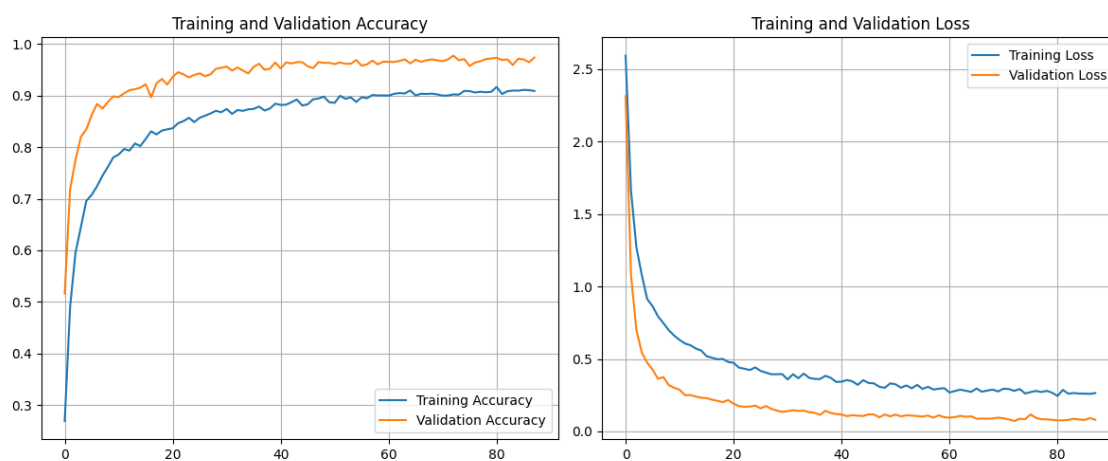
Bagian ini memaparkan hasil pelatihan model, evaluasi kuantitatif (akurasi, *confusion matrix*, metrik per kelas), implementasi model pada aplikasi web, serta hasil uji kegunaan SUS.

### Hasil Pelatihan Model

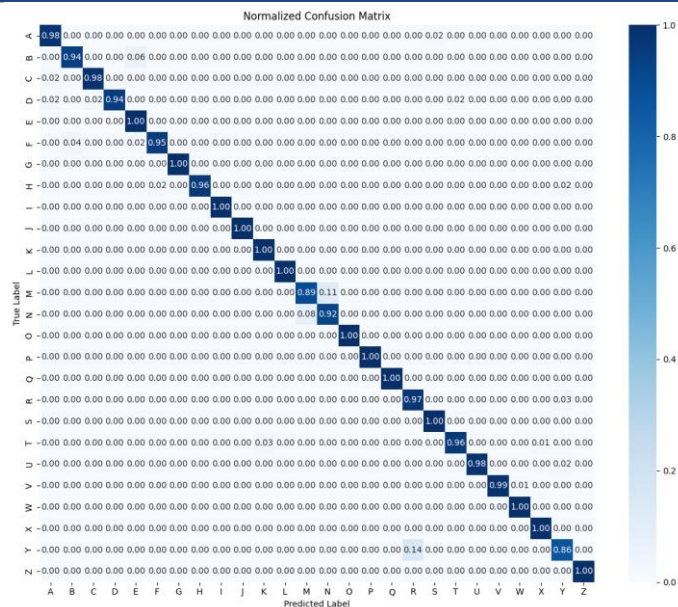
Pelatihan model *Multilayer Perceptron* (MLP) menunjukkan proses pembelajaran yang stabil. Pada fase awal pelatihan, nilai *loss* menurun dan akurasi meningkat seiring model mempelajari pola dasar dari representasi *landmark*. Seiring bertambahnya *epoch*, kurva pelatihan cenderung stabil yang menunjukkan model mendekati konvergensi. Penerapan *Batch Normalization* membantu kestabilan distribusi aktivasi antar layer, sedangkan *Dropout* dan *early stopping* membantu menekan *overfitting*. Kurva *training* dan *validation accuracy* serta *loss* selama pelatihan ditunjukkan pada Gambar 2

Evaluasi model dilakukan pada data uji terpisah sebanyak 1657 sampel untuk menilai kemampuan generalisasi pada data yang tidak pernah dilihat saat pelatihan. Hasil pengujian menunjukkan akurasi pengenalan sebesar 97,77% dengan *macro precision* 97,54%, *macro recall* 97,43%, dan *macro F1-score* 97,47%, yang mengindikasikan performa model tinggi dan relatif merata pada seluruh kelas huruf.

Untuk menganalisis kesalahan prediksi antar huruf, digunakan *confusion matrix* [22]. *Confusion matrix* alfabet BISINDO pada data uji ditunjukkan pada Gambar 3. Secara umum, nilai pada *confusion matrix* terkonsentrasi pada diagonal utama yang menunjukkan sebagian besar sampel berhasil diprediksi dengan benar. Kesalahan lebih sering muncul pada huruf-huruf yang memiliki konfigurasi jari mirip atau perbedaannya sangat halus, sehingga perubahan kecil pada pose, sudut kamera, atau kestabilan *landmark* dapat menyebabkan pertukaran prediksi antar kelas.



**Gambar 2.** Grafik training dan validation accuracy serta loss selama proses pelatihan  
Sumber: Hasil penelitian



**Gambar 3.** Confusion matrix alfabet BISINDO pada data uji.

Sumber: Hasil penelitian

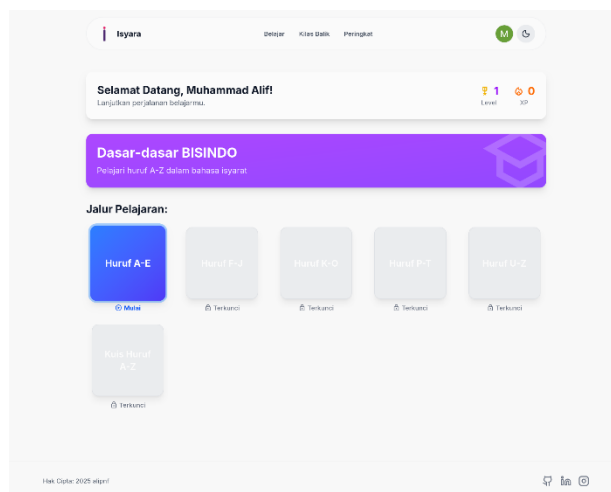
**Tabel 2**

**Precision, recall, dan F1-score per kelas**

| Huruf | Precision | Recall | F1-score | Support |
|-------|-----------|--------|----------|---------|
| A     | 0.96      | 0.98   | 0.97     | 56      |
| B     | 0.94      | 0.94   | 0.94     | 34      |
| C     | 0.98      | 0.98   | 0.98     | 65      |
| D     | 1.00      | 0.94   | 0.97     | 54      |
| E     | 0.96      | 1.00   | 0.98     | 65      |
| F     | 0.98      | 0.95   | 0.96     | 57      |
| G     | 1.00      | 1.00   | 1.00     | 80      |
| H     | 1.00      | 0.96   | 0.98     | 51      |
| I     | 1.00      | 1.00   | 1.00     | 71      |
| J     | 1.00      | 1.00   | 1.00     | 69      |
| K     | 0.97      | 1.00   | 0.99     | 71      |
| L     | 1.00      | 1.00   | 1.00     | 59      |
| M     | 0.89      | 0.89   | 0.89     | 56      |
| N     | 0.92      | 0.92   | 0.92     | 75      |
| O     | 1.00      | 1.00   | 1.00     | 51      |
| P     | 1.00      | 1.00   | 1.00     | 54      |
| Q     | 1.00      | 1.00   | 1.00     | 95      |
| R     | 0.95      | 0.97   | 0.96     | 97      |
| S     | 0.99      | 1.00   | 0.99     | 75      |
| T     | 0.99      | 0.96   | 0.97     | 78      |
| U     | 1.00      | 0.98   | 0.99     | 60      |
| V     | 1.00      | 0.99   | 0.99     | 72      |
| W     | 0.98      | 1.00   | 0.99     | 50      |
| X     | 0.99      | 1.00   | 0.99     | 71      |
| Y     | 0.86      | 0.86   | 0.86     | 35      |

| Huruf        | Precision | Recall | F1-score | Support |
|--------------|-----------|--------|----------|---------|
| Z            | 1.00      | 1.00   | 1.00     | 56      |
| Macro Avg    | 0.98      | 0.97   | 0.97     | 1657    |
| Weighted Avg | 0.98      | 0.98   | 0.98     | 1657    |

Sumber: Hasil Penelitian

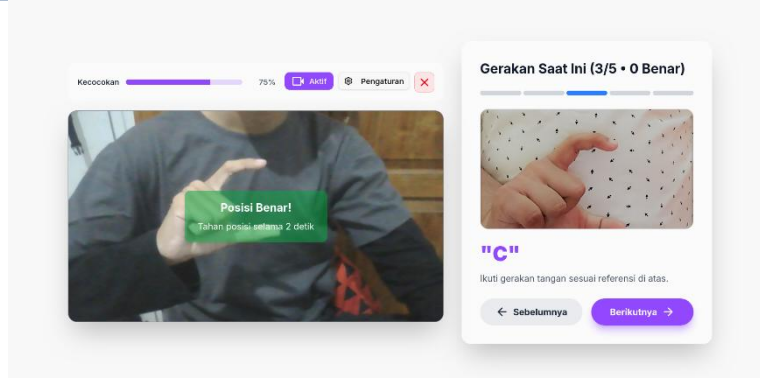


**Gambar 4.** Tampilan halaman progress pengguna  
Sumber: Hasil penelitian

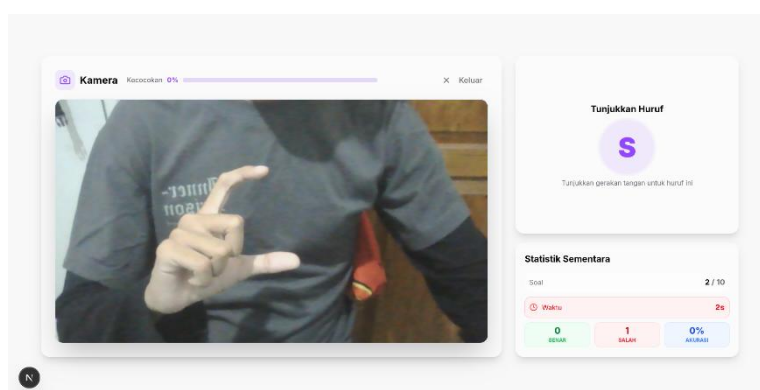
Selain akurasi, evaluasi per kelas menggunakan *precision*, *recall*, dan *F1-score* memberikan gambaran huruf mana yang paling kuat dikenali dan huruf mana yang masih menantang. Umumnya, huruf dengan bentuk gestur yang unik menghasilkan nilai metrik tinggi karena pola *landmark* konsisten. Sebaliknya, huruf yang gesturnya mirip dengan kelas lain atau memiliki variasi pose lebih besar cenderung memiliki nilai *recall* atau *F1-score* lebih rendah. Rekap metrik per huruf disajikan pada Tabel 2 sebagai dasar analisis performa detail per kelas.

### Implementasi Sistem Real-Time pada Browser

Implementasi sistem real-time dilakukan melalui aplikasi web dengan memanfaatkan kamera perangkat setiap frame video diproses menggunakan *MediaPipe Hands* untuk menghasilkan koordinat *landmark*. Koordinat tersebut kemudian diproses membentuk vektor fitur berdimensi 126 yang konsisten. Vektor fitur selanjutnya diprediksi oleh model MLP yang telah diekspor ke TensorFlow.js sehingga inferensi berjalan sepenuhnya di sisi klien tanpa memerlukan server. Hasil prediksi huruf ditampilkan langsung pada antarmuka web sebagai umpan balik sehingga pengguna dapat menyesuaikan gestur secara interaktif. Contoh tampilan halaman *progress*, belajar, dan kuis ditunjukkan pada Gambar 4, Gambar 5, dan Gambar 6.



**Gambar 5.** Tampilan halaman belajar  
Sumber: Hasil penelitian



**Gambar 6.** Tampilan halaman quiz  
Sumber: Hasil penelitian

### Uji Kegunaan Sistem Menggunakan SUS

Uji kegunaan sistem dievaluasi memakai *System Usability Scale* (SUS) yang memuat 10 butir pernyataan untuk menilai persepsi pengguna terhadap kemudahan penggunaan aplikasi. Responden terlebih dahulu mencoba aplikasi Isyara, kemudian memberikan skor untuk setiap butir menggunakan skala *Likert* 1–5 (1 = sangat tidak setuju, 5 = sangat setuju). Rangkuman 10 pernyataan SUS beserta rata-rata penilaian responden ditampilkan pada Tabel 3. 10 pernyataan seperti pada Tabel 3. Pernyataan bernomor ganjil (1, 3, 5, 7, 9) bersifat positif, sedangkan pernyataan bernomor genap (2, 4, 6, 8, 10) bersifat negatif, sehingga diperlukan proses konversi skor sebelum dihitung menjadi nilai SUS akhir. Skor SUS dihitung dengan mengonversi jawaban item ganjil menjadi (jawaban – 1) dan item genap menjadi (5 – jawaban). Seluruh kontribusi kemudian dijumlahkan (rentang 0–40) dan dikalikan 2,5 untuk memperoleh skor SUS pada rentang 0–100. Skor yang lebih tinggi menunjukkan tingkat kegunaan (*usability*) yang lebih baik. Skor SUS rata-rata yang diperoleh adalah 70,53 dari 52 responden, dengan rentang skor 27,50–100,00. Hasil ini menunjukkan tingkat kegunaan aplikasi berada pada kategori cukup baik

**Tabel 3**  
**Pertanyaan System Usability Scale (SUS)**

| No | Pernyataan                                                                                       | Rata-rata |
|----|--------------------------------------------------------------------------------------------------|-----------|
| 1  | Saya kemungkinan akan memakai aplikasi ini lagi di lain waktu.                                   | 3,75      |
| 2  | Menurut saya, penggunaan aplikasi ini terasa berbelit dan bisa dibuat lebih simpel.              | 2,33      |
| 3  | Saya menilai aplikasi ini mudah dioperasikan.                                                    | 4,19      |
| 4  | Saya rasa saya perlu bantuan orang yang lebih paham teknis untuk menjalankannya.                 | 2,50      |
| 5  | Fitur-fitur yang tersedia terasa menyatu dan bekerja dengan baik satu sama lain.                 | 4,04      |
| 6  | Saya menemukan beberapa bagian yang kurang konsisten saat digunakan.                             | 2,31      |
| 7  | Sebagian besar orang seharusnya bisa cepat memahami cara memakai aplikasi ini.                   | 4,10      |
| 8  | Saya sempat merasa kerepotan atau bingung ketika menggunakan aplikasi ini.                       | 2,44      |
| 9  | Saya merasa yakin dan nyaman saat menggunakan aplikasi ini.                                      | 4,10      |
| 10 | Saya perlu belajar banyak hal terlebih dulu sebelum bisa menggunakan aplikasi ini dengan lancar. | 2,38      |

Sumber: Hasil Penelitian

## SIMPULAN

Penelitian ini mengembangkan aplikasi web pembelajaran alfabet BISINDO yang mampu mengenali 26 huruf secara real-time menggunakan *MediaPipe Hands* untuk ekstraksi *hand landmark* dan model *Multilayer Perceptron* (MLP) untuk klasifikasi. Melalui pra-pemrosesan berupa normalisasi relatif terhadap *wrist*, penskalaan berbasis jarak maksimum, pembentukan input 126 dimensi dengan *padding*, serta *augmentasi horizontal flip*, rotasi acak, dan *Gaussian noise* 0,02, model mencapai akurasi 97,77% pada data uji. Model diekspor ke TensorFlow.js sehingga inferensi berjalan sepenuhnya di sisi klien tanpa memerlukan server inferensi.

Berdasarkan uji kegunaan menggunakan SUS, diperoleh skor rata-rata 70,53 dengan rentang 27,50–100,00, yang menunjukkan tingkat kegunaan aplikasi berada pada kategori cukup baik. Pengembangan selanjutnya dapat difokuskan pada perluasan variasi data (lebih banyak pengguna dan kondisi lingkungan), serta pengembangan pengenalan gestur dinamis untuk membentuk kata atau kalimat dalam BISINDO.

## DAFTAR PUSTAKA

- [1] G. Gumelar, H. Hafiar, dan P. Subekti, "KONSTRUKSI MAKNA BISINDO SEBAGAI BUDAYA TULI BAGI ANGGOTA GERKATIN," *INFORMASI*, vol. 48, no. 1, hlm. 65, Jul 2018, doi: 10.21831/informasi.v48i1.17727.
- [2] B. Asriandhini dan C. H. Rahmawati, "BAHASA ISYARAT INDONESIA SEBAGAI KONSTRUKSI IDENTITAS DAN CITRA SOSIAL TULI DI PURWOKERTO," *JRK (Jurnal Riset Komunikasi)*, vol. 12, no. 1, Jun 2021, doi: 10.31506/jrk.v12i1.10059.

- [3] B. R. Fajri, A. D. Samala, dan F. Ranuharja, "MEDIA INTERAKTIF PENGENALAN BAHASA ISYARAT BISINDO," *Jurnal Teknologi Informasi dan Pendidikan*, vol. 13, no. 1, hlm. 35–44, Mar 2020, doi: 10.24036/tip.v13i1.293.
- [4] Z. Nikolawatin, P. Setyosari, dan S. Ulfa, "PENGEMBANGAN MEDIA TUTORIAL BAHASA ISYARAT UNTUK SISWA TUNARUNGU SLB BC KEPANJEN," *JINOTEP (Jurnal Inovasi dan Teknologi Pembelajaran) Kajian dan Riset dalam Teknologi Pembelajaran*, vol. 6, no. 1, hlm. 15–22, Jul 2019, doi: 10.17977/um031v6i12019p015.
- [5] M. Iqbal, E. Supriyati, dan T. Listyorini, "IMPLEMETASI OFFLINE PENGENALAN SISTEM ISYARAT BAHASA INDONESIA MENGGUNAKAN METODE DYNAMIC TIME WARPING PADA PERANGKAT ANDROID," *Simetris : Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, vol. 6, no. 2, hlm. 391, Nov 2015, doi: 10.24176/simet.v6i2.477.
- [6] S. Anugerah, S. Ulfa, dan A. Husna, "Pengembangan Video Pembelajaran Bahasa Isyarat Indonesia (Bisindo) Untuk Siswa Tunarungu Di Sekolah Dasar," *JINOTEP (Jurnal Inovasi dan Teknologi Pembelajaran): Kajian dan Riset Dalam Teknologi Pembelajaran*, vol. 7, no. 2, hlm. 76–85, Okt 2020, doi: 10.17977/um031v7i22020p076.
- [7] M. Iqbal, E. Supriyati, dan T. Listyorini, "Pengembangan Sarung Tangan Wireless Untuk Pengenalan Sistem Isyarat Bahasa Indonesia Berbasis Modul Bluetooth Hc-05," Seminar Nasional Teknologi dan Informatika 2015, Des 2015.
- [8] A. N. Sihananto, M. M. Al Haromainy, dan A. P. Sari, "PEMILAHAN JENIS SAMPAH MENGGUNAKAN ALGORITMA CNN," *Scan: Jurnal Teknologi Informasi dan Komunikasi*, vol. 17, no. 3, Jan 2023, doi: 10.33005/scan.v17i3.3523.
- [9] A. A. Q. Mohammed, Y. Gao, Z. Ji, J. Lv, M. Sajjatul Islam, dan Y. Sang, "Automatic 3D Skeleton-based Dynamic Hand Gesture Recognition Using Multi-Layer Convolutional LSTM," dalam *2021 7th International Conference on Robotics and Artificial Intelligence*, New York, NY, USA: ACM, Nov 2021, hlm. 8–14. doi: 10.1145/3505688.3505690.
- [10] C. Lugaresi dkk., "MediaPipe: A Framework for Building Perception Pipelines," Jun 2019.
- [11] J. R. Lewis, "The System Usability Scale: Past, Present, and Future," *Int. J. Hum. Comput. Interact.*, vol. 34, no. 7, hlm. 577–590, Jul 2018, doi: 10.1080/10447318.2018.1455307.
- [12] M. Hyzy dkk., "System Usability Scale Benchmarking for Digital Health Apps: Meta-analysis," *JMIR Mhealth Uhealth*, vol. 10, no. 8, hlm. e37290, Agu 2022, doi: 10.2196/37290.
- [13] C. Lugaresi dkk., "MediaPipe: A Framework for Building Perception Pipelines," Jun 2019.
- [14] F. Zhang dkk., "MediaPipe Hands: On-device Real-time Hand Tracking," Jun 2020.
- [15] D. Smilkov dkk., "TensorFlow.js: Machine Learning for the Web and Beyond," Feb 2019.

- [16] C. Shorten dan T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J. Big Data*, vol. 6, no. 1, hlm. 60, Des 2019, doi: 10.1186/s40537-019-0197-0.
- [17] J. Warchocki, M. Vlasenko, dan Y. B. Eisma, "GRLib: An Open-Source Hand Gesture Detection and Recognition Python Library," Okt 2023.
- [18] D. A. Supangkat, F. N. Alibasyah, M. R. Dzulkornain, M. Hilal, M. A. N. Nashrulloh, dan F. T. Anggraeny, "PERBANDINGAN KOMBINASI METODE TEMPLATE MATCHING DAN ALGORITMA FEATURE MATCHING PADA PENGENALAN MATA UANG INDIA," *SCAN - Jurnal Teknologi Informasi dan Komunikasi*, vol. 16, no. 3, Des 2021, doi: 10.33005/scan.v16i3.2636.
- [19] S. Barman dan S. Majumdar, "Person-Independent Hand Gesture Recognition Using MediaPipe and Multi-layer Perceptron," 2025, hlm. 3–16. doi: 10.1007/978-981-96-1348-9\_1.
- [20] M. Vilares Ferro, Y. Doval Mosquera, F. J. Ribadas Pena, dan V. M. Darriba Bilbao, "Early stopping by correlating online indicators in neural networks," *Neural Networks*, vol. 159, hlm. 109–124, Feb 2023, doi: 10.1016/j.neunet.2022.11.035.
- [21] S. J. Reddi, S. Kale, dan S. Kumar, "On the Convergence of Adam and Beyond," Apr 2019.
- [22] A. Tharwat, "Classification assessment methods," *Applied Computing and Informatics*, vol. 17, no. 1, hlm. 168–192, Jan 2021, doi: 10.1016/j.aci.2018.08.003.