



Aplikasi Web Pembelajaran BISINDO Interaktif dengan Deteksi Gestur Real-Time

Surya Adjie Umar Tenda*, Aryanzah Nugrah Maggi, Vicrena Yolita Vanesa Kalangit, Toban T. Pairunan, Anritsu Steven Christian Polii

Program Studi Teknik Informatika, Jurusan Teknik Elektro, Politeknik Negeri Manado, Manado, Indonesia

Email: ^{1,*}suryaadji17@gmail.com, ¹aryanzahmaggi@gmail.com, ¹vicrenakalangit04@gmail.com, ¹pairunantoban546@gmail.com, ¹anritsupolii@polimdo.ac.id

Email Penulis Korespondensi*: suryaadji17@gmail.com

Abstrak—Bahasa Isyarat Indonesia (BISINDO) merupakan bahasa utama komunitas Tuli di Indonesia, namun aksesibilitas pembelajarannya masih terbatas, khususnya di Sulawesi Utara. Penelitian ini bertujuan merancang dan membangun aplikasi web pembelajaran BISINDO interaktif bernama BISINDO.app yang memberikan umpan balik real-time terhadap gerakan isyarat pengguna melalui webcam. Sistem memanfaatkan MediaPipe Hand Landmarker untuk mengekstraksi 21 titik landmark tangan yang diklasifikasikan menggunakan model deep learning low-parameter: Convolutional Neural Network satu dimensi (Conv1D) untuk gestur statis dan Bidirectional Long Short-Term Memory (BiLSTM) untuk gestur dinamis. Seluruh inferensi berjalan di sisi klien melalui TensorFlow.js sehingga aplikasi dapat diakses tanpa instalasi tambahan. Aplikasi dibangun dengan React, Node.js/Express, dan SQLite, dengan autentikasi JWT, serta konten gestur yang divalidasi bersama Pusat Bahasa Isyarat Indonesia (PUSBISINDO) Sulawesi Utara. Hasil pelatihan menunjukkan akurasi pada data validasi (belum diuji pada test set independen) sebesar 97,64% (alfabet, 26 kelas), 98,01% (angka statis, 11 kelas), 87,69% (angka dinamis, 10 kelas), dan 99,26% (kata dasar, 17 kelas), dengan ukuran bobot 165–650 KB per model; evaluasi pada test set independen menjadi agenda pengembangan lanjutan. Algoritma stabilisasi prediksi berbasis ambang keyakinan 0,8, konsistensi tiga tick, dan penghalusan Exponential Moving Average menghasilkan umpan balik yang stabil dan responsif. Pengujian fungsional terhadap 14 skenario menunjukkan seluruh fitur berjalan sesuai spesifikasi, dan evaluasi usability terhadap 13 responden menghasilkan skor SUS rata-rata 83,19 (kategori Excellent). Kontribusi utama penelitian ini adalah integrasi pengenalan gestur real-time berbasis client-side dengan mekanisme umpan balik interaktif dalam satu platform pembelajaran BISINDO berbasis web. Penelitian ini diharapkan meningkatkan aksesibilitas pembelajaran BISINDO dan mendukung inklusi sosial komunitas Tuli di Indonesia.

Kata Kunci: BISINDO; MediaPipe; Conv1D; BiLSTM; Pembelajaran Interaktif

Abstract—Indonesian Sign Language (BISINDO) is the primary language of the Deaf community in Indonesia; however, access to BISINDO learning remains limited, particularly in North Sulawesi. This research develops an interactive web-based BISINDO learning application, BISINDO.app, providing real-time feedback on users' sign gestures via webcam. The system uses MediaPipe Hand Landmarker to extract 21 hand landmarks, classified using low-parameter deep learning models: a one-dimensional Convolutional Neural Network (Conv1D) for static gestures and a Bidirectional Long Short-Term Memory (BiLSTM) network for dynamic gestures. All inference runs client-side via TensorFlow.js, so the application requires no additional installation. The application uses React, Node.js/Express, and SQLite, secured with JSON Web Token (JWT) authentication, with content validated together with North Sulawesi's Indonesian Sign Language Center (PUSBISINDO). Training results show accuracies on the validation set (not yet evaluated on an independent test set) of 97.64% (alphabet, 26 classes), 98.01% (static numbers, 11 classes), 87.69% (dynamic numbers, 10 classes), and 99.26% (basic words, 17 classes), with model weights ranging from 165 KB to 650 KB; evaluation on an independent test set is left for future work. A prediction stabilization algorithm using a 0.8 confidence threshold, three-tick consistency, and Exponential Moving Average smoothing produces stable, responsive feedback. Functional testing across 14 scenarios confirms all features work as specified, and a usability evaluation with 13 respondents yielded an average SUS score of 83.19 (Excellent). The main contribution of this research is the integration of real-time client-side gesture recognition with an interactive feedback mechanism within a single web-based BISINDO learning platform. This research is expected to improve BISINDO learning accessibility and support the Deaf community's social inclusion in Indonesia.

Keywords: BISINDO; MediaPipe; Conv1D; BiLSTM; Interactive Learning

1. PENDAHULUAN

Bahasa Isyarat Indonesia (BISINDO) merupakan bahasa isyarat alami yang berkembang di komunitas Tuli di Indonesia dan digunakan sebagai alat komunikasi utama sehari-hari. Berbeda dengan Sistem Isyarat Bahasa Indonesia (SIBI) yang bersifat buatan, BISINDO tumbuh secara organik di tengah komunitas Tuli dan memuat variasi dialek daerah, sekaligus menjadi penanda identitas komunitas Tuli yang berkaitan erat dengan isu inklusi sosial [1], [2].

Meskipun perannya vital, akses pembelajaran BISINDO bagi masyarakat umum masih terbatas, terutama di Sulawesi Utara, karena kelas tatap muka tidak selalu tersedia dan materi daring yang terstruktur masih sedikit [3]. Kondisi ini membuat keluarga teman Tuli, tenaga pendidik, maupun masyarakat umum kesulitan menemukan media belajar yang terjangkau dan mampu mengoreksi gerakan yang mereka lakukan.

Di sisi lain, perkembangan teknologi *computer vision* dan *deep learning* membuka peluang besar bagi pengenalan gestur bahasa isyarat. Borman dkk. [4] menggunakan Principal Component Analysis (PCA) untuk mengenali alfabet BISINDO pada citra statis, namun terbatas sebagai penerjemah satu arah. Priadiyatna dkk. [5] merancang sarung tangan pintar berbasis IoT untuk menerjemahkan gestur BISINDO, tetapi membutuhkan perangkat keras khusus yang membatasi aksesibilitasnya. Susanty dkk. [6] menerapkan *transfer learning* pada citra BISINDO dan mencapai akurasi tinggi, namun berfokus pada akurasi model tanpa disertai aplikasi pembelajaran. Arisandi dkk. [7] menggunakan *Convolutional Neural*



Network (CNN) untuk klasifikasi alfabet BISINDO, sedangkan Pangestu dkk. [8] menerapkan YOLOv8 untuk deteksi gestur BISINDO secara *real-time* dalam video. Aryananda dan Samopa [9] membandingkan CNN dan RNN untuk pengenalan BISINDO pada platform Android, dan Aljabar dan Suharjo [10] mengombinasikan CNN dan LSTM untuk meningkatkan akurasi pengenalan. Deleviar dkk. [11] mengembangkan sistem konversi suara ke video BISINDO berbasis LSTM, namun sistem tersebut bekerja satu arah dari suara ke isyarat, bukan sebaliknya. Perkembangan terbaru turut mengeksplorasi arsitektur 3D CNN-BiLSTM untuk pengenalan BISINDO tingkat kalimat [12], perbandingan model ringan berbasis landmark MediaPipe seperti MobileNetV3 dan EfficientNetV2 [13], penerapan transfer learning EfficientNet pada perangkat Android [14], deteksi BISINDO pada video menggunakan YOLOv7 [15], ekstraksi fitur skeletal dengan LSTM untuk gestur alfabet [16], pengembangan aplikasi Android untuk pembelajaran BISINDO [17], penerapan deep learning secara umum untuk pengenalan BISINDO [18], serta tinjauan sistematis teknik deep learning untuk pengenalan BISINDO [19]; namun seluruhnya tetap berfokus pada peningkatan akurasi pengenalan atau bersifat satu arah tanpa disertai platform pembelajaran interaktif berumpan balik langsung bagi pemula. Pada tahun yang sama, Kindy dkk. [20] merilis dataset video BISINDO tingkat kata beserta metode baseline-nya, sedangkan Sebastian dkk. [21] mengusulkan pengenalan BISINDO berbasis spatially aware body gesture recognition; kedua penelitian tersebut memperkuat ketersediaan data dan akurasi pengenalan, tetapi keduanya berhenti pada evaluasi model tanpa menyediakan platform pembelajaran berumpan balik langsung.

MediaPipe Hands dari Google menyediakan pelacakan titik kunci tangan secara *real-time* yang ringan untuk dijalankan di *browser* [22], sementara *Long Short-Term Memory* (LSTM) dan variannya berpotensi mengenali gestur dinamis berupa urutan gerakan tangan [23]. Namun, sebagian besar penelitian terdahulu tersebut berfokus sebagai penerjemah satu arah, bukan media pembelajaran interaktif bagi pemula, dan sering hanya menampilkan video contoh tanpa umpan balik otomatis. Padahal, umpan balik langsung (*immediate feedback*) merupakan komponen penting dalam pembelajaran keterampilan motorik, termasuk berbahasa isyarat. Di luar konteks BISINDO, penelitian pengenalan bahasa isyarat lain turut mengeksplorasi arsitektur hybrid deep learning dengan optimasi lanjutan untuk meningkatkan akurasi pengenalan [24], [25], namun pendekatan tersebut juga belum diarahkan pada penyediaan aplikasi pembelajaran interaktif berbasis web dengan umpan balik langsung.

Dari perspektif pedagogis, pembelajaran keterampilan motorik seperti bahasa isyarat sangat bergantung pada siklus latihan-koreksi yang cepat; penundaan umpan balik dapat menyebabkan pembentukan kebiasaan gerakan yang keliru dan sulit diperbaiki di kemudian hari. Aplikasi berbasis web dipilih sebagai platform distribusi karena tidak memerlukan proses instalasi maupun spesifikasi perangkat keras khusus, sehingga dapat diakses secara luas oleh keluarga teman Tuli, tenaga pendidik di sekolah inklusi, maupun masyarakat umum menggunakan perangkat yang sudah tersedia sehari-hari seperti laptop atau ponsel berkamera. Pendekatan *client-side inference* turut menjawab kekhawatiran privasi, karena rekaman video pengguna tidak pernah dikirimkan ke server, hal yang cukup relevan mengingat sensitivitas data visual yang melibatkan wajah dan gerakan tubuh pengguna. Selain itu, elemen gamifikasi seperti poin pengalaman, level, dan rentetan hari latihan diyakini dapat meningkatkan motivasi belajar mandiri, khususnya bagi pemula yang belum memiliki akses rutin ke kelas tatap muka maupun pendamping yang fasih berbahasa isyarat.

Berdasarkan kondisi tersebut, diperlukan aplikasi pembelajaran BISINDO berbasis web yang mampu memberikan umpan balik *real-time* terhadap gerakan pengguna melalui webcam. Pendekatan *low-parameter* menjadi krusial agar model cukup ringan dijalankan langsung di *browser* melalui TensorFlow.js tanpa server komputasi berdaya tinggi [26], [27]. Kerja sama dengan Pusat Bahasa Isyarat Indonesia (PUSBISINDO) Sulawesi Utara memastikan dataset dan standar gestur yang digunakan sesuai praktik BISINDO di lapangan, sejalan dengan praktik terbaik pelibatan komunitas Tuli dalam pengembangan dataset bahasa isyarat [28].

Celah penelitian yang teridentifikasi adalah belum adanya aplikasi web pembelajaran BISINDO yang mengintegrasikan pengenalan gestur statis dan dinamis secara *real-time* di sisi klien dengan umpan balik interaktif, algoritma stabilisasi prediksi, serta konten yang tervalidasi oleh komunitas Tuli. Penelitian ini merancang dan membangun aplikasi web pembelajaran BISINDO bernama BISINDO.app yang mengintegrasikan deteksi tangan MediaPipe, model klasifikasi gestur statis (Conv1D) dan dinamis (BiLSTM), serta mekanisme umpan balik interaktif dengan algoritma stabilisasi prediksi, guna mendukung inklusi komunikasi bagi komunitas Tuli di Indonesia. Secara spesifik, penelitian ini bertujuan (1) mengembangkan model pengenalan gerakan BISINDO berbasis *computer vision* dan *deep learning* dengan pendekatan *low-parameter* yang mampu bekerja secara *real-time* melalui webcam; (2) membangun aplikasi web pembelajaran BISINDO yang interaktif dengan umpan balik langsung; (3) memvalidasi akurasi sistem dan kesesuaian materi pembelajaran dengan standar gestur BISINDO bersama PUSBISINDO Sulawesi Utara; serta (4) mengevaluasi tingkat kemudahan penggunaan (*usability*) aplikasi bagi pengguna pemula.

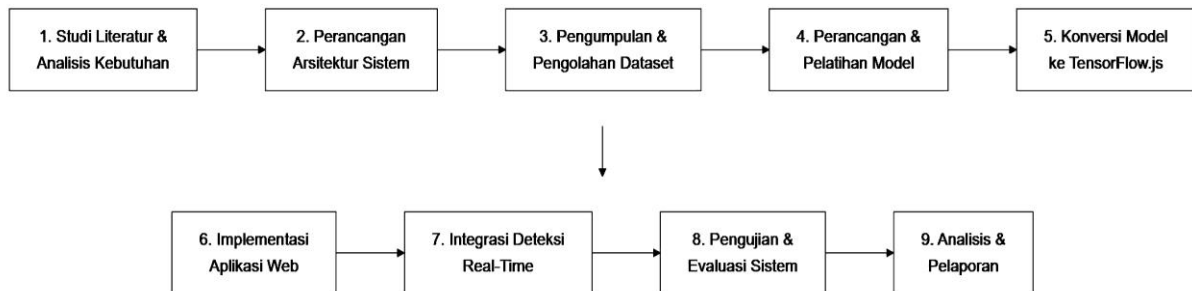
2. METODOLOGI PENELITIAN

Penelitian ini termasuk kategori penelitian terapan (*applied research*) di bidang Teknik Informatika dengan pendekatan rekayasa perangkat lunak dan eksperimen. Aspek rekayasa perangkat lunak tampak pada perancangan dan pembangunan aplikasi web pembelajaran, sedangkan aspek eksperimen tampak pada pengumpulan dataset, pelatihan, pengujian, dan evaluasi model pengenalan gestur. Metode pengembangan perangkat lunak mengadaptasi model *prototyping* dengan iterasi bertahap: kebutuhan dianalisis, prototipe dibangun, diuji bersama mitra PUSBISINDO Sulawesi Utara, kemudian

diperbaiki pada iterasi berikutnya. Penelitian dilaksanakan di Politeknik Negeri Manado dari bulan Februari sampai dengan Juni 2026.

2.1 Tahapan Penelitian

Penelitian dilakukan dalam sembilan tahap sebagaimana ditunjukkan pada Gambar 1: (1) studi literatur dan analisis kebutuhan; (2) perancangan arsitektur sistem; (3) pengumpulan dan pengolahan dataset; (4) perancangan dan pelatihan model; (5) konversi model ke TensorFlow.js; (6) implementasi aplikasi web; (7) integrasi deteksi *real-time*; (8) pengujian dan evaluasi sistem; serta (9) analisis dan pelaporan. Tahap analisis kebutuhan menghasilkan sepuluh kebutuhan fungsional, di antaranya autentikasi JWT, penyediaan modul pembelajaran (Alfabet, Angka, Kata Dasar, Salam, dan Frasa Umum), deteksi tangan dan klasifikasi gestur di sisi klien, algoritma stabilisasi prediksi, pencatatan sesi latihan, kamus gestur tanpa login, serta mode latihan mandiri; dan enam kebutuhan non-fungsional yang menekankan inferensi di sisi klien untuk menjaga privasi, ukuran model di bawah 1 MB, interval inferensi 100 ms, serta kompatibilitas dengan *browser* modern. Rangkaian sembilan tahap penelitian beserta urutan pelaksanaannya digambarkan pada Gambar 1.



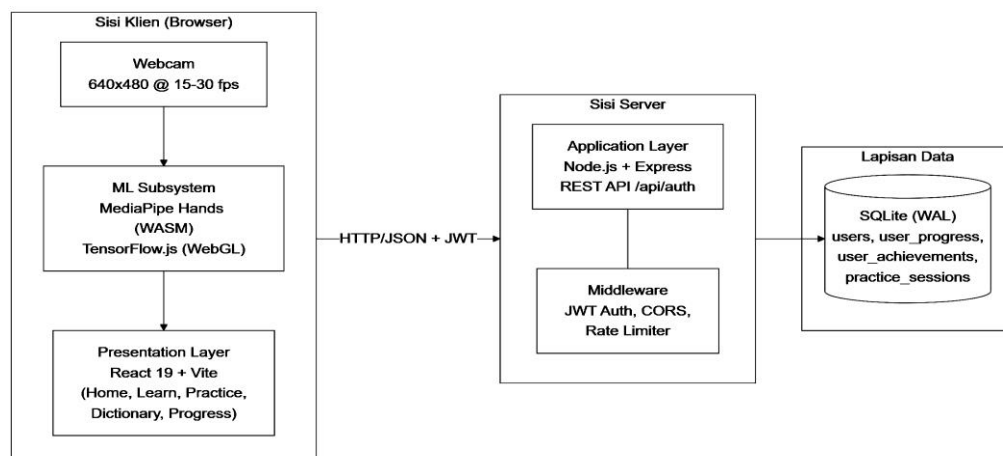
Gambar 1. Tahapan Penelitian

Gambar 1 menyusun sembilan tahap penelitian ke dalam dua baris alur yang saling berurutan. Baris pertama pada Gambar 1 memuat tahap konseptual dan pemodelan, yaitu studi literatur dan analisis kebutuhan, perancangan arsitektur sistem, pengumpulan dan pengolahan dataset, perancangan dan pelatihan model, serta konversi model ke format TensorFlow.js. Baris kedua pada Gambar 1 memuat tahap realisasi perangkat lunak, mulai dari implementasi aplikasi web, integrasi deteksi *real-time*, pengujian dan evaluasi sistem, sampai analisis dan pelaporan. Urutan pada Gambar 1 menegaskan bahwa hasil konversi model menjadi masukan langsung bagi tahap implementasi aplikasi, sehingga pengembangan model dan pengembangan perangkat lunak tidak berjalan sebagai dua kegiatan yang terpisah.

2.2 Arsitektur Sistem

Sistem dirancang dengan arsitektur yang memisahkan empat bagian fungsional, sebagaimana ditunjukkan pada Gambar 2: *Presentation Layer* berupa antarmuka pengguna React yang berjalan di *browser*; *ML Subsystem* yang juga berjalan di sisi klien dan terdiri dari MediaPipe Hands (WASM) serta model klasifikasi TensorFlow.js; *Application Layer* berupa layanan REST API berbasis Node.js/Express yang menangani autentikasi dan pencatatan progres; serta *Data Layer* berupa basis data SQLite dengan mode WAL. Pemisahan ini memberikan dua keuntungan: seluruh komputasi pengenalan gestur dilakukan di perangkat pengguna sehingga server tidak memerlukan GPU dan aliran video tidak pernah meninggalkan perangkat; dan server hanya menangani operasi ringan sehingga dapat dilayani basis data berkas tunggal tanpa konfigurasi tambahan.

Susunan lapisan sistem beserta pembagian tanggung jawab antara sisi klien dan sisi server digambarkan pada Gambar 2.



Gambar 2. Arsitektur Sistem BISINDO.app



Gambar 2 memperlihatkan penempatan seluruh beban komputasi pengenalan gestur pada sisi klien. Blok sisi klien pada Gambar 2 memuat webcam beresolusi 640x480 piksel, subsistem ML yang terdiri atas MediaPipe Hands berbasis WASM dan TensorFlow.js berbasis WebGL, serta lapisan presentasi React 19. Pertukaran data antara klien dan server pada Gambar 2 terbatas pada HTTP/JSON yang diamankan JWT untuk keperluan autentikasi dan pencatatan progres, sedangkan lapisan data diwakili SQLite bermode WAL dengan empat tabel utama. Pemisahan tanggung jawab pada Gambar 2 menjelaskan alasan server tidak memerlukan GPU dan alasan aliran video pengguna tidak pernah meninggalkan perangkat.

2.3 Pengumpulan Dataset dan Perancangan Model

Dataset dikumpulkan secara mandiri melalui perekaman webcam dengan rujukan gestur dari narasumber PUSBISINDO Sulawesi Utara pada berbagai kondisi pencahayaan dan sudut pandang. Perekaman melibatkan 3 narasumber/subjek dengan variasi kondisi pencahayaan (dalam ruangan/luar ruangan), sudut pandang kamera, dan latar belakang yang sengaja divariasikan untuk mengurangi risiko bias; pembagian data latih dan validasi dilakukan secara acak per-sampel sehingga berpotensi terjadi kebocoran data antar keduanya. Skema pembagian ini dipilih karena keterbatasan jumlah subjek pada tahap pengumpulan data awal. Pada pengembangan lanjutan, split akan dilakukan berbasis subjek (subject-independent split) dan dievaluasi pada test set yang benar-benar terpisah dari proses pelatihan, sehingga akurasi yang dilaporkan pada bagian hasil perlu dibaca sebagai indikasi performa pada kondisi validasi internal, bukan generalisasi penuh terhadap pengguna baru, mengingat jumlah dan keragaman demografis subjek pada tahap pengumpulan data ini masih terbatas dan perlu diperluas pada pengembangan selanjutnya. Untuk gestur statis (alfabet A–Z dan angka 0–10), setiap citra diproses MediaPipe menjadi vektor 126 fitur ($2 \text{ tangan} \times 21 \text{ landmark} \times 3 \text{ koordinat}$). Untuk gestur dinamis (angka 11–20 dan kata dasar), direkam sekuens 60 frame per sampel; khusus isyarat kata digunakan MediaPipe Holistic Landmarker sehingga tiap frame menghasilkan 167 fitur yang mencakup landmark kedua tangan dan pose tubuh bagian atas. Augmentasi diterapkan pada data latih meliputi derau Gaussian ($\sigma=0,005$), penskalaan acak ($\pm 5\%$), dan translasi acak ($\pm 3\%$) untuk gestur statis; serta *temporal jitter* (± 2 frame) dan variasi kecepatan (*resampling* 0,85–1,15) untuk gestur dinamis. Data dibagi menjadi data latih dan validasi dengan rasio sekitar 80:20. Komposisi akhir dataset dirangkum pada Tabel 1. Pendekatan augmentasi berbasis koordinat ini sejalan dengan strategi mengatasi keterbatasan data pada model penerjemah BISINDO yang telah dilaporkan sebelumnya [29].

Rincian jumlah kelas, jumlah data latih, jumlah data validasi, dan bentuk fitur pada setiap kelompok gestur disajikan pada Tabel 1.

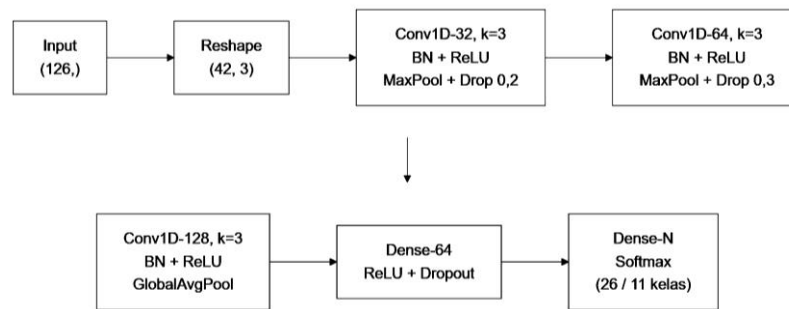
Tabel 1. Komposisi Dataset Penelitian

Kelompok Gestur	Jml Kelas	Data Latih	Data Validasi	Bentuk Fitur
Alfabet A-Z (statis)	26	7.558	1.910	vektor (126,)
Angka 0-10 (statis)	11	582	151	vektor (126,)
Angka 11-20 (dinamis)	10	± 200 sekuens	± 65 sekuens	sekuens (60,126)
Kata dasar (dinamis)	17	± 540 sekuens	± 136 sekuens	sekuens (60,167)

Tabel 1 memperlihatkan ketimpangan jumlah sampel antarkelompok gestur yang perlu diperhatikan saat menafsirkan hasil pelatihan. Kelompok alfabet pada Tabel 1 memiliki 7.558 sampel latih dan 1.910 sampel validasi, jauh lebih besar dibandingkan kelompok angka statis yang hanya memiliki 582 sampel latih dan 151 sampel validasi. Kelompok gestur dinamis pada Tabel 1 direpresentasikan sebagai sekuens 60 frame, dengan angka 11 sampai 20 hanya memiliki sekitar 200 sekuens latih, sedangkan kata dasar mencapai sekitar 540 sekuens latih. Perbedaan bentuk fitur pada Tabel 1, yaitu vektor 126 dimensi untuk gestur statis dan sekuens 60x167 untuk isyarat kata, menjadi dasar pemilihan dua arsitektur model yang berbeda pada penelitian ini.

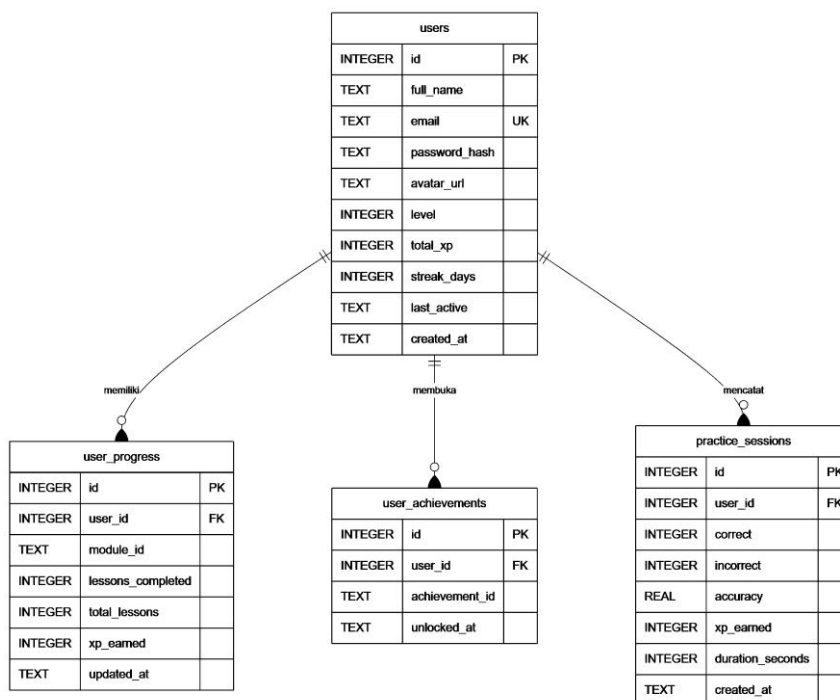
Dua arsitektur model dirancang sesuai karakteristik gestur, ditunjukkan pada Gambar 3. Model Conv1D untuk gestur statis menerima vektor 126 fitur yang disusun ulang menjadi matriks 42×3 , diproses tiga blok konvolusi (32, 64, 128 filter, kernel 3) masing-masing dengan *batch normalization*, ReLU, *max pooling*, dan *dropout* (0,2–0,3), diikuti *global average pooling*, Dense-64, dan lapisan keluaran *softmax*, mengikuti praktik umum desain arsitektur jaringan konvolusi ringan [30]. Total parameter model hanya sekitar 42 ribu (bobot TensorFlow.js ≈ 168 KB). Model BiLSTM untuk gestur dinamis menerima sekuens 60 frame, diproses dua blok BiLSTM bertumpuk (64 dan 32 unit per arah) dengan *dropout* 0,3, diikuti Dense-32 dan lapisan keluaran *softmax*. Total parameter berkisar 141–163 ribu (bobot ≈ 560 –650 KB), tetap memenuhi kriteria *low-parameter*. Pelatihan menggunakan optimizer Adam (laju pembelajaran awal 0,001), *sparse categorical crossentropy*, *batch size* 64, serta *callback early stopping* (patience 15), *reduce learning rate on plateau*, dan *model checkpoint*. Susunan lapisan model Conv1D yang digunakan untuk klasifikasi gestur statis digambarkan pada Gambar 3.

Gambar 3 menampilkan aliran tensor pada model Conv1D, dimulai dari vektor masukan 126 fitur yang disusun ulang menjadi matriks 42×3 . Tiga blok konvolusi pada Gambar 3 menggunakan 32, 64, dan 128 filter dengan kernel berukuran 3, masing-masing dipadukan dengan *batch normalization*, aktivasi ReLU, *max pooling*, serta *dropout* 0,2 sampai 0,3. Bagian akhir Gambar 3 memperlihatkan *global average pooling* yang menekan jumlah parameter sebelum lapisan Dense-64 dan lapisan keluaran *softmax* dengan 26 kelas untuk alfabet atau 11 kelas untuk angka statis. Susunan ringkas pada Gambar 3 menjadi alasan utama ukuran bobot model tetap berada di bawah 200 KB sehingga layak dijalankan pada peramban.



Gambar 3. Arsitektur Model Conv1D untuk Gestur Statis

Basis data dirancang dengan empat tabel utama sebagaimana ditunjukkan pada Gambar 4: *users* untuk data akun dan atribut gamifikasi (*level*, *total_xp*, *streak*), *user_progress* untuk progres per modul, *user_achievements* untuk pencapaian, dan *practice_sessions* untuk riwayat sesi latihan. Pengenalan *real-time* dijadwalkan setiap 100 ms pada resolusi kamera 640×480 piksel (15–30 fps); probabilitas keluaran dihaluskan dengan *Exponential Moving Average* ($\alpha=0,6$) sebelum diteruskan ke algoritma stabilisasi berbasis ambang keyakinan 0,8 dan konsistensi tiga *tick*. Struktur tabel basis data beserta relasi antartentitasnya digambarkan pada Gambar 4.



Gambar 4. Rancangan Skema Basis Data (ERD)

Gambar 4 memetakan relasi antartabel yang menopang pencatatan progres dan fitur gamifikasi. Tabel *users* pada Gambar 4 berperan sebagai entitas induk yang menyimpan identitas akun beserta atribut *level*, *total_xp*, dan *streak_days*. Relasi satu-ke-banyak pada Gambar 4 menghubungkan tabel *users* dengan *user_progress*, *user_achievements*, dan *practice_sessions*, sehingga setiap sesi latihan tercatat lengkap dengan jumlah jawaban benar, jumlah jawaban salah, akurasi, XP yang diperoleh, dan durasi sesi. Rancangan pada Gambar 4 memungkinkan perhitungan kemajuan belajar dilakukan melalui kueri agregasi sederhana tanpa denormalisasi tambahan.

Antarmuka dibangun menggunakan React 19 dengan Vite, React Router, dan Framer Motion. Sisi server dibangun dengan Node.js dan Express, menggunakan *better-sqlite3* sebagai penggerak basis data, *bcrypt* untuk *hashing* kata sandi, *jsonwebtoken* untuk JWT, serta *express-rate-limit* untuk pembatasan laju permintaan (maksimal 30 permintaan per 15 menit pada endpoint autentikasi). Model hasil pelatihan dikonversi dari arsip .keras ke format TensorFlow.js LayersModel menggunakan skrip konversi mandiri yang membaca bobot secara langsung tanpa bergantung pada pustaka *tensorflowjs* Python yang berukuran besar, sehingga proses konversi tetap terkendali dan deterministik.

Pengujian sistem mencakup empat jenis: (1) pengujian fungsional (*black-box*) untuk memastikan seluruh fitur berjalan sesuai kebutuhan; (2) pengujian kinerja model untuk mengukur akurasi latih dan validasi; (3) validasi mitra bersama PUSBISINDO Sulawesi Utara untuk menguji kesesuaian gestur dengan standar BISINDO; serta (4) pengujian *usability* menggunakan instrumen System Usability Scale (SUS) [31], [32] yang melibatkan 13 responden (8 mahasiswa Teknik Informatika Politeknik Negeri Manado dan 5 orang teman terdekat/masyarakat umum). Kuesioner SUS terdiri

dari 10 butir pernyataan berskala Likert 1–5; skor akhir dihitung dengan menjumlahkan kontribusi tiap butir (bernomor ganjil: nilai–1; bernomor genap: 5–nilai) lalu dikalikan 2,5 untuk memperoleh nilai 0–100.

3. HASIL DAN PEMBAHASAN

3.1 Antarmuka, Fitur, dan Layanan API

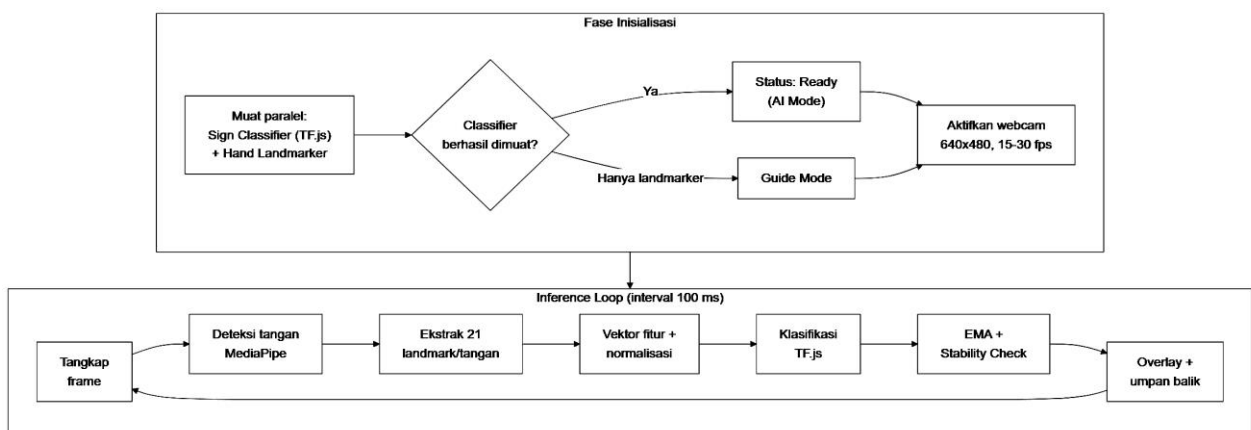
Aplikasi hasil penelitian ini diberi nama BISINDO.app. Halaman beranda menampilkan navigasi yang bersih dan intuitif dengan lima area utama: Beranda, Belajar (akses modul pembelajaran terstruktur), Latihan (sesi praktik dengan umpan balik *real-time*), Kamus (dapat diakses tanpa login), dan Progress (pemantauan kemajuan belajar). Fitur gamifikasi diimplementasikan melalui sistem XP, level, dan *streak days* yang mendorong konsistensi belajar; nilai-nilai ini disimpan pada tabel *users* dan diperbarui melalui REST API setiap sesi latihan selesai. Struktur komponen mengikuti pola *Single Page Application* React: komponen publik (Home, Login, Register, Dictionary) dapat diakses bebas, sedangkan komponen terproteksi (Learn, Practice, Progress) dibungkus komponen *ProtectedRoute* yang memeriksa token JWT di penyimpanan lokal dan memvalidasinya ke server melalui *endpoint* GET */api/auth/me* sebelum mengizinkan akses.

Layanan API diimplementasikan dengan Node.js dan Express pada port 3001, dilengkapi CORS, *parser* JSON, dan *rate limiter* pada *endpoint* autentikasi. Protokol autentikasi mencakup empat skenario: registrasi (validasi input, pengecekan keunikan email, *hashing* kata sandi dengan *bcrypt*, penerbitan JWT), login (verifikasi kata sandi dan penerbitan JWT), pemeriksaan sesi melalui header *Authorization Bearer*, dan *logout* yang cukup menghapus token dari sisi klien sesuai karakteristik *stateless* JWT.

3.2 Pipeline Pengenalan Gestur dan Algoritma Stabilisasi

Pipeline pengenalan gestur *real-time* terdiri dari fase inisialisasi dan *inference loop*, sebagaimana ditunjukkan pada Gambar 5. Pada fase inisialisasi, sistem memuat model klasifikasi (TensorFlow.js) dan Hand Landmarker (MediaPipe) secara paralel; jika model klasifikasi berhasil dimuat, sistem masuk ke *AI Mode*, sedangkan jika hanya *landmarker* yang tersedia, sistem masuk ke *Guide Mode* yang menampilkan visualisasi landmark tanpa klasifikasi. Fase *inference loop* berjalan tiap 100 ms: menangkap frame, mendeteksi tangan, mengekstraksi 21 titik landmark per tangan, menyusun vektor 126 fitur, menormalisasi dengan parameter min-max hasil pelatihan, lalu mengklasifikasikan dengan model Conv1D. Tiga teknik optimasi diterapkan agar inferensi tetap berada dalam interval 100 ms: pra-alokasi *buffer* Float32Array tanpa alokasi ulang untuk menghindari *garbage collection* berulang; kalibrasi temperatur probabilitas ($T=1,5$) agar keluaran *softmax* tidak terlalu percaya diri; dan *fallback* otomatis dari backend WebGL ke CPU jika akselerasi GPU tidak tersedia. Pada perangkat dengan GPU terintegrasi Intel UHD 620, inferensi WebGL memerlukan rata-rata 8–12 ms per prediksi.

Urutan proses pengenalan gestur mulai dari fase inisialisasi sampai perulangan inferensi digambarkan pada Gambar 5.



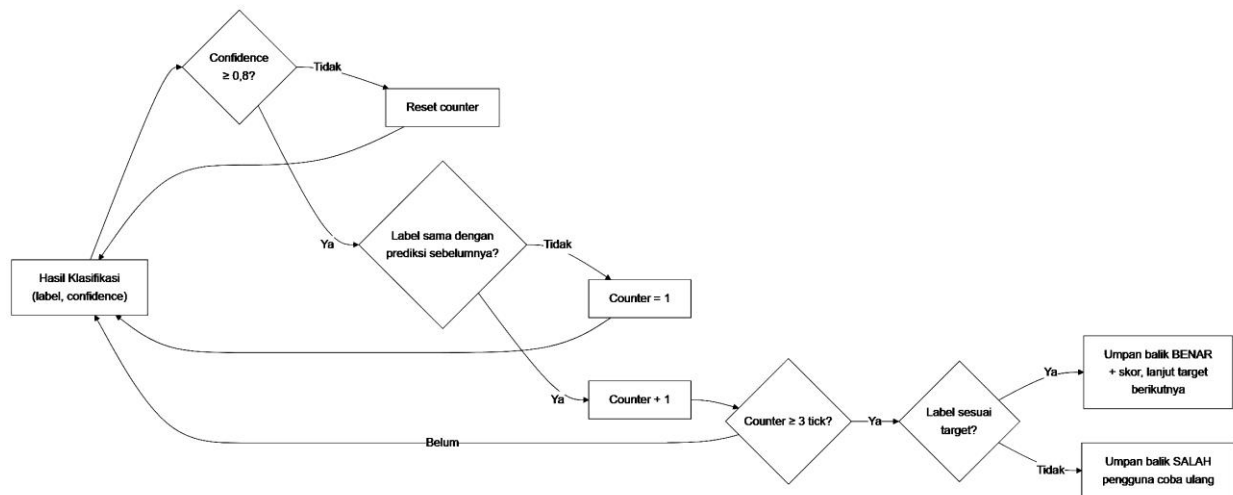
Gambar 5. Pipeline Pengenalan Gestur Real-Time BISINDO.app

Gambar 5 membagi proses pengenalan gestur menjadi fase inisialisasi dan inference loop. Fase inisialisasi pada Gambar 5 memuat model klasifikasi TensorFlow.js dan Hand Landmarker secara paralel, lalu menentukan mode operasi, yaitu AI Mode ketika model klasifikasi berhasil dimuat dan Guide Mode ketika hanya landmarker yang tersedia. Inference loop pada Gambar 5 berjalan pada interval 100 ms dengan urutan tetap, yaitu penangkapan frame, deteksi tangan, ekstraksi 21 titik landmark per tangan, penyusunan dan normalisasi vektor fitur, klasifikasi TensorFlow.js, penghalusan EMA yang disertai pemeriksaan stabilitas, serta penggambaran overlay umpan balik. Perulangan tertutup pada Gambar 5 menjaga latensi setiap siklus tetap terprediksi karena tidak ada proses tambahan di luar rangkaian tersebut.

Salah satu kontribusi teknis penting penelitian ini adalah algoritma stabilisasi prediksi untuk mengatasi masalah *flickering* (kedipan label tidak stabil), sebagaimana ditunjukkan pada Gambar 6. Setiap hasil klasifikasi diperiksa terhadap ambang keyakinan 0,8; jika tidak terpenuhi, pencacah direset. Jika keyakinan terpenuhi dan label sama dengan prediksi sebelumnya, pencacah *tick* ditambah; jika berbeda, direset ke satu. Ketika label cocok dengan target dan stabil selama



tiga *tick* berurutan, sistem memberikan umpan balik benar, menambah skor dan XP, lalu otomatis melanjutkan ke target berikutnya setelah jeda 900 ms; jika tidak cocok selama tiga *tick*, sistem memberikan umpan balik salah dan mengizinkan percobaan ulang. Sebelum pemeriksaan *tick*, probabilitas kelas dihaluskan dengan *Exponential Moving Average* ($\alpha=0,6$) sehingga *jitter* antarframe teredam tanpa mengorbankan responsivitas. Pendekatan ini lebih efisien secara komputasi dibandingkan *temporal filtering* konvensional seperti *moving average* berjendela panjang atau *Kalman filter*, karena tidak memerlukan penyimpanan riwayat prediksi yang panjang. Nilai tiga *tick* dipilih melalui pengujian empiris terhadap rentang 1-5 *tick* pada interval 100 ms per *tick*: nilai di bawah tiga (1-2 *tick*, ≤ 200 ms) menghasilkan lebih banyak umpan balik salah akibat prediksi sesaat yang belum stabil, sedangkan nilai di atas tiga (4-5 *tick*, ≥ 400 ms) menambah latensi yang terasa oleh pengguna tanpa peningkatan akurasi umpan balik yang berarti; tiga *tick* (~ 300 ms) memberikan keseimbangan terbaik antara stabilitas dan responsivitas pada pengujian ini. Alur pengambilan keputusan pada algoritma stabilisasi prediksi digambarkan pada Gambar 6.



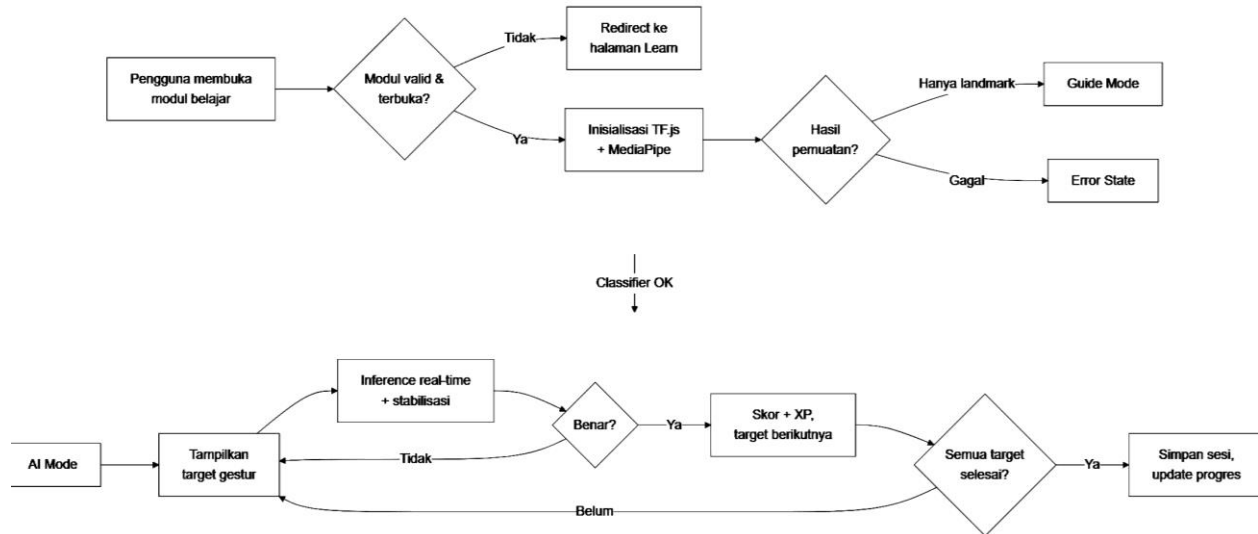
Gambar 6. Algoritma Stabilisasi Prediksi Gestur BISINDO.app

Gambar 6 merinci logika keputusan yang menyaring hasil klasifikasi sebelum ditampilkan sebagai umpan balik. Cabang pertama pada Gambar 6 memeriksa nilai keyakinan terhadap ambang 0,8, dan hasil di bawah ambang tersebut membuat pencacah direset sehingga prediksi yang belum stabil tidak pernah sampai ke pengguna. Cabang kedua pada Gambar 6 membandingkan label saat ini dengan prediksi sebelumnya untuk menentukan penambahan pencacah atau pengaturan ulang pencacah ke nilai satu. Setelah pencacah mencapai tiga *tick*, percabangan akhir pada Gambar 6 memberikan umpan balik benar beserta penambahan skor dan XP ketika label sesuai target, atau memberikan umpan balik salah dengan kesempatan mencoba ulang ketika label tidak sesuai target.

Siklus lengkap sesi pembelajaran (*learning session lifecycle*) ditunjukkan pada Gambar 7. Sistem memeriksa validitas dan status keterbukaan modul, menginisialisasi runtime TensorFlow.js dan MediaPipe, lalu masuk ke salah satu dari tiga mode: *AI Mode*, *Guide Mode*, atau *Error State*. Dalam *AI Mode*, sesi berlangsung dalam perulangan: menampilkan target gestur, menjalankan inferensi dengan stabilisasi, mencatat hasil, menambah skor jika benar, dan melanjutkan hingga seluruh target selesai. Di akhir sesi, ringkasan hasil (benar, salah, akurasi, durasi, XP) dikirim ke server dan mencatatnya pada tabel *practice_sessions* sekaligus memperbarui *user_progress*. Rangkaian status dan proses sepanjang satu sesi pembelajaran digambarkan pada Gambar 7.

Gambar 7 menyatukan pemeriksaan prasyarat, inisialisasi runtime, dan perulangan sesi latihan dalam satu alur. Bagian awal Gambar 7 memvalidasi status modul, yaitu modul yang tidak valid atau masih terkunci mengarahkan pengguna kembali ke halaman Learn, sedangkan modul yang valid memicu inisialisasi TensorFlow.js dan MediaPipe. Hasil pemuatan pada Gambar 7 menentukan tiga kemungkinan status, yaitu *AI Mode*, *Guide Mode*, dan *Error State*. Perulangan sesi pada Gambar 7 menampilkan target gestur, menjalankan inferensi berstabilisasi, menambah skor dan XP untuk jawaban benar, lalu menyimpan ringkasan sesi dan memperbarui progres pengguna ketika seluruh target selesai dikerjakan.

Aplikasi ini menyediakan lima modul pembelajaran, masing-masing dengan status implementasi yang berbeda. Modul Alfabet (26 target) sudah beroperasi penuh dalam *AI Mode* dengan model klasifikasi terintegrasi. Modul Angka (21 target) sudah aktif dengan panduan landmark; model angka statis dan dinamisnya sendiri sudah selesai dilatih dan tinggal diintegrasikan sebagai bundel TensorFlow.js. Modul Kata Dasar (12 target) dan Salam & Sapaan (5 target) untuk saat ini masih berjalan dalam *Guide Mode*; model katanya yang berbasis BiLSTM sudah dilatih dan diekspor, hanya saja membutuhkan alur kerja sekuensial (*buffer* 60 frame) yang berbeda dari *pipeline* klasifikasi statis, sehingga integrasinya menjadi pekerjaan lanjutan. Modul Frasa Umum sendiri masih terkunci dan direncanakan untuk pengembangan berikutnya. Karena seluruh modul berbagi infrastruktur yang sama (MediaPipe dan TensorFlow.js) lewat *hook useBisindoPractice*, penambahan modul baru cukup dilakukan dengan menyediakan bundel model dan daftar target tanpa perlu mengubah alur aplikasi.



Gambar 7. Diagram Siklus Sesi Pembelajaran BISINDO.app

3.3 Hasil Pelatihan dan Kinerja Model

Empat model klasifikasi telah dilatih sesuai rancangan pada Bagian 2. Ringkasan data pelatihan dan hasil akurasi disajikan pada Tabel 2.

Tabel 2. Data Pelatihan dan Hasil Akurasi Model BISINDO.app

Model	Kelas	Arsitektur	Epoch	Akurasi Latih	Akurasi Validasi
Alfabet A-Z	26	Conv1D	100	93,50%	97,64%
Angka 0-10 (statis)	11	Conv1D	58	90,51%	98,01%
Angka 11-20 (dinamis)	10	BiLSTM	85	99,31%	87,69%
Kata dasar	17	BiLSTM	41	99,94%	99,26%

Tabel 2 merangkum konfigurasi pelatihan dan capaian akurasi keempat model klasifikasi. Baris alfabet dan baris angka statis pada Tabel 2 menunjukkan akurasi validasi yang lebih tinggi daripada akurasi latih, yaitu 97,64% berbanding 93,50% dan 98,01% berbanding 90,51%. Sebaliknya, baris angka dinamis pada Tabel 2 memperlihatkan selisih terbalik sebesar 11,62 poin persentase antara akurasi latih 99,31% dan akurasi validasi 87,69%. Jumlah epoch pada Tabel 2 bervariasi antara 41 sampai 100 karena mekanisme early stopping menghentikan pelatihan ketika kinerja pada data validasi berhenti membaik.

Model alfabet mencapai akurasi validasi 97,64% pada 26 kelas dengan 9.468 sampel total, menunjukkan bahwa representasi koordinat landmark cukup diskriminatif untuk membedakan bentuk tangan alfabet BISINDO, termasuk isyarat dua tangan. Akurasi validasi yang lebih tinggi daripada akurasi latih merupakan efek *dropout* dan augmentasi yang hanya aktif saat pelatihan. Model angka statis mencapai 98,01% meskipun jumlah sampel per kelas lebih kecil, terbantu oleh jumlah kelas yang lebih sedikit. Model kata dasar mencapai akurasi validasi tertinggi (99,26%), mengindikasikan pola temporal antar kata yang cukup berbeda pada fitur holistik 167 dimensi. Model angka dinamis mencapai akurasi validasi 87,69%, terendah di antara keempat model, dengan selisih cukup besar terhadap akurasi latihnya (99,31%), mengindikasikan *overfitting* ringan akibat keterbatasan jumlah sekuens per kelas serta kemiripan pola gerakan antar angka belasan. Akar masalah ini dapat ditelusuri pada karakteristik temporal datanya: angka-angka belasan berbagi lintasan tangan yang hampir identik dengan angka satuan yang bersesuaian pada awal gerakannya, sehingga fitur yang ditangkap BiLSTM pada tahap-tahap awal sekuens menjadi kurang diskriminatif untuk memisahkan kelas-kelas yang secara visual mirip tersebut. Akibatnya, alih-alih mempelajari pola gerak yang tergeneralisasi, model cenderung mengandalkan variasi spesifik pada data latih untuk membedakan kelas, kecenderungan yang semakin kuat mengingat jumlah data yang tersedia sangat terbatas, yaitu rata-rata hanya sekitar 20 sekuens per kelas dari total ± 200 sekuens data latih, jauh lebih sedikit dibandingkan ± 540 sekuens yang tersedia untuk kata dasar. Penambahan sampel dari lebih banyak subjek serta variasi kecepatan gerak yang lebih luas merupakan langkah perbaikan yang direncanakan.

Evaluasi lebih lanjut pada *validation set* menunjukkan bahwa model Alfabet mencapai *precision*, *recall*, dan *F1-score* masing-masing sebesar 0,979, 0,979, dan 0,979 (akurasi 97,85% pada 1.910 sampel); model Angka Statis mencapai 0,987 untuk ketiga metrik tersebut (akurasi 98,68% pada 151 sampel); model Angka Dinamis mencapai *precision* 0,913, *recall* 0,908, dan *F1-score* 0,904 (akurasi 90,77% pada 65 sampel), konsisten dengan akurasi yang relatif lebih rendah akibat kompleksitas trajektori gestur dinamis; sementara model Kata Dasar (17 kelas) mencapai *precision* 0,994, *recall* 0,991, dan *F1-score* 0,992 pada 136 sampel sekuens validasi (akurasi 99,26%). Nilai metrik yang seimbang dan tinggi pada model Kata Dasar membuktikan bahwa pemanfaatan 167 fitur holistik (kombinasi landmark kedua tangan dan pose tubuh bagian atas) sangat efektif dalam membedakan trajektori spasio-temporal antar-



isyarat kata. Perlu dicatat bahwa evaluasi ini menggunakan data validasi internal, bukan *test set* independen yang terpisah sepenuhnya dari proses pelatihan, sejalan dengan batasan penelitian yang diuraikan pada Bagian 2.3.

Dari sisi kompleksitas, seluruh model memenuhi kriteria *low-parameter* yang ditetapkan, sebagaimana dirangkum pada Tabel 3. Sebagai perbandingan, model CNN citra penuh untuk tugas serupa umumnya berukuran jutaan parameter dengan berkas puluhan megabita. Dengan bobot di bawah 1 MB, model-model BISINDO.app dapat diunduh dalam hitungan detik pada koneksi internet biasa dan dieksekusi di bawah interval inferensi 100 ms pada perangkat tanpa GPU diskrit.

Perbandingan jumlah parameter, ukuran bobot TensorFlow.js, dan bentuk masukan untuk keempat model disajikan pada Tabel 3.

Tabel 3. Perbandingan Ukuran dan Parameter Model

Model	Jumlah Parameter	Ukuran Bobot TF.js	Bentuk Masukan
Alfabet (Conv1D)	± 42 ribu	± 168 KB	(126,)
Angka statis (Conv1D)	± 41 ribu	± 165 KB	(126,)
Angka dinamis (BiLSTM)	± 141 ribu	± 560 KB	(60, 126)
Kata dasar (BiLSTM)	± 163 ribu	± 650 KB	(60, 167)

Tabel 3 menegaskan bahwa keempat model tetap berada dalam kategori *low-parameter*. Dua model Conv1D pada Tabel 3 hanya memerlukan sekitar 41 ribu sampai 42 ribu parameter dengan ukuran bobot 165 KB sampai 168 KB, sedangkan dua model BiLSTM memerlukan 141 ribu sampai 163 ribu parameter dengan ukuran bobot 560 KB sampai 650 KB. Selisih ukuran pada Tabel 3 muncul karena masukan berupa sekuens 60 frame menuntut kapasitas pemodelan temporal yang lebih besar daripada masukan berupa vektor tunggal. Total bobot seluruh model pada Tabel 3 masih berada di bawah 1,6 MB, sehingga seluruh berkas model dapat diunduh peramban dalam hitungan detik pada koneksi internet rumahan.

Augmentasi data ternyata cukup krusial untuk mencapai akurasi tinggi dengan dataset yang terbatas. Untuk gestur statis, augmentasi berbasis koordinat (derau Gaussian, penskalaan, translasi) jauh lebih efisien dibandingkan augmentasi berbasis citra, sebab hanya mengolah 126 angka per sampel, bukan ribuan piksel. Temuan ini sejalan dengan Fadillah dkk [29], yang juga menekankan peran krusial augmentasi data dalam mengatasi keterbatasan dataset pada model penerjemah BISINDO. Pada tahap pengumpulan dataset sendiri, diterapkan strategi deteksi *multi-pass* untuk memaksimalkan cakupan deteksi tangan: lintasan pertama memproses citra asli, lintasan kedua menerapkan CLAHE (*Contrast Limited Adaptive Histogram Equalization*) jika tangan belum terdeteksi, dan lintasan ketiga menaikkan kecerahan sebesar 30% bila masih gagal juga. Strategi tiga lapis ini cukup membantu meningkatkan keberhasilan deteksi pada kondisi pencahayaan yang kurang ideal, situasi yang cukup sering ditemui di lingkungan domestik pengguna akhir.

3.4 Pengujian Fungsional dan Evaluasi Usability

Pengujian fungsional dilaksanakan dengan metode *black-box* pada *browser* Chrome, Firefox, dan Edge versi terkini. Seluruh 14 skenario pengujian mencakup registrasi, login, persistensi sesi, pemuatan model TensorFlow.js, aktivasi webcam, deteksi tangan dan *overlay* landmark, klasifikasi gestur, stabilisasi prediksi, penyimpanan hasil sesi, akses kamus tanpa autentikasi, *Guide Mode*, dan *rate limiter* menunjukkan hasil sesuai ekspektasi, sehingga seluruh kebutuhan fungsional F-01 sampai F-10 dinyatakan terpenuhi.

Evaluasi *usability* menggunakan instrumen SUS melibatkan 13 responden: 8 mahasiswa Teknik Informatika Politeknik Negeri Manado dan 5 orang teman terdekat/masyarakat umum yang sama sekali belum pernah mempelajari BISINDO. Setiap responden menyelesaikan satu sesi penuh modul Alfabet sebelum mengisi kuesioner. Hasil pengujian dirangkum pada Tabel 4.

Tabel 4. Hasil Pengujian System Usability Scale (n=13)

Kelompok Responden	Jumlah	Rata-rata SUS	Kategori
Mahasiswa Teknik Informatika	8	84,25	Excellent
Teman Terdekat/Masyarakat Umum	5	81,50	Excellent
Keseluruhan	13	83,19	Excellent (Grade A)

Tabel 4 menunjukkan skor SUS yang konsisten tinggi pada kedua kelompok responden. Kelompok mahasiswa Teknik Informatika pada Tabel 4 memperoleh rata-rata 84,25, sedangkan kelompok teman terdekat dan masyarakat umum memperoleh rata-rata 81,50, sehingga selisih antarkelompok hanya 2,75 poin dan keduanya sama-sama berada pada kategori Excellent. Nilai keseluruhan pada Tabel 4 sebesar 83,19 dari 13 responden melampaui ambang Grade A sebesar 80,3, yang menandakan aplikasi dapat digunakan tanpa pelatihan khusus.

SUS rata-rata sebesar 83,19 dari skala 0–100 masuk kategori *Excellent* (Grade A, $\geq 80,3$) berdasarkan skala adjektif Bangor dkk. [32] dan berada di atas persentil ke-85 dari data normatif 500+ studi *usability*. Kelompok mahasiswa Politeknik Negeri Manado memberikan skor rata-rata 84,25, sedikit lebih tinggi dibandingkan kelompok teman terdekat/masyarakat umum (81,50), kemungkinan karena keterbiasaan menggunakan aplikasi berbasis web dan webcam turut memengaruhi kemudahan adaptasi terhadap antarmuka BISINDO.app, meski kedua kelompok sama-sama belum pernah mempelajari BISINDO sebelumnya, mencerminkan bahwa pengguna dengan latar belakang teknis merasakan manfaat umpan balik *real-time* secara lebih langsung. Analisis per butir menunjukkan pernyataan “sistem mudah



digunakan” memperoleh rata-rata tertinggi (4,54/5), sedangkan pernyataan “membutuhkan bantuan teknis” memperoleh rata-rata terendah (1,62/5), mengonfirmasi bahwa BISINDO.app berhasil menurunkan hambatan teknis penggunaan bagi kedua kelompok responden.

3.5 Perbandingan dengan Penelitian Terdahulu

Tabel 5 menyajikan perbandingan BISINDO.app dengan penelitian-penelitian terdahulu yang relevan dalam domain pengenalan dan pembelajaran BISINDO. Salah satu penelitian yang paling relevan adalah integrasi model CNN dengan platform web untuk pengenalan BISINDO [33], meskipun penelitian tersebut belum menyertakan mekanisme umpan balik interaktif maupun algoritma stabilisasi prediksi seperti yang diterapkan pada BISINDO.app.

Rangkuman perbandingan antara BISINDO.app dan penelitian terdahulu pada lima aspek pembandingan disajikan pada Tabel 5.

Tabel 5. Perbandingan BISINDO.app dengan Penelitian Terdahulu

Penelitian	Platform	Umpan Balik RT	Client-Side	Gestur Statis	Gestur Dinamis
Borman dkk. (2018) [4]	Desktop	Tidak	Tidak	Ya	Tidak
Priadiyatna dkk. (2020) [5]	Perangkat keras	Tidak	Tidak	Ya	Sebagian
Susanty dkk. (2021) [6]	Studi model	Tidak	Tidak	Ya	Tidak
Pangestu dkk. (2024) [8]	Desktop	Sebagian	Tidak	Ya	Tidak
Aryananda & Samopa (2024) [9]	Android	Tidak	Ya	Ya	Ya
Deleviar dkk. (2025) [11]	Web	Tidak	Tidak	Tidak	Ya
BISINDO.app (penelitian ini)	Web (browser)	Ya	Ya	Ya	Ya

Tabel 5 memetakan posisi BISINDO.app terhadap enam penelitian terdahulu pada lima aspek pembandingan. Kolom umpan balik real-time pada Tabel 5 memperlihatkan bahwa hanya penelitian ini yang menyediakan koreksi langsung terhadap gerakan pengguna, sedangkan Pangestu dkk. [8] baru mencapai deteksi real-time tanpa mekanisme koreksi untuk keperluan pembelajaran. Kolom client-side pada Tabel 5 menempatkan BISINDO.app bersama Aryananda dan Samopa [9] sebagai penelitian yang menjalankan inferensi pada perangkat pengguna, namun pendekatan berbasis Android pada penelitian tersebut tetap menuntut proses instalasi. Baris terakhir Tabel 5 menunjukkan bahwa cakupan gestur statis sekaligus gestur dinamis pada satu platform web belum ditemukan pada penelitian pembandingan lainnya.

Dari analisis perbandingan tersebut terlihat bahwa penelitian-penelitian sebelumnya umumnya berfokus pada akurasi pengenalan atau penerjemahan gestur tanpa menyediakan mekanisme umpan balik *real-time* yang interaktif maupun platform web yang dapat diakses tanpa instalasi. BISINDO.app mengisi celah ini dengan mengintegrasikan pengenalan gestur yang akurat dan platform pembelajaran interaktif berbasis web dalam satu sistem yang kohesif. Penelitian ini memberikan tiga kontribusi kebaruan utama. Pertama, integrasi model Conv1D (gestur statis) dan BiLSTM (gestur dinamis) dalam satu platform web pembelajaran BISINDO yang berjalan sepenuhnya di sisi klien (*client-side inference*) merupakan pendekatan yang belum ditemukan dalam literatur BISINDO sebelumnya. Kedua, algoritma stabilisasi prediksi berbasis pemeriksaan ganda (ambang keyakinan 0,8, konsistensi tiga *tick*, dan penghalusan EMA) memberikan keseimbangan optimal antara respons cepat dan stabilitas umpan balik. Ketiga, kolaborasi langsung dengan PUSBISINDO Sulawesi Utara dalam validasi konten memastikan keaslian representasi BISINDO, aspek yang sering diabaikan dalam penelitian yang murni teknis.

Perbandingan dengan penelitian pada Tabel 5 juga menunjukkan bahwa pendekatan client-side yang diterapkan BISINDO.app memberikan keunggulan pada aspek privasi dan skalabilitas, karena tidak memerlukan infrastruktur server untuk inferensi model, berbeda dengan sebagian penelitian terdahulu yang mengandalkan pemrosesan di sisi server. Namun, trade-off dari pendekatan ini adalah ketergantungan pada kemampuan komputasi perangkat pengguna, sehingga performa pada perangkat dengan spesifikasi rendah atau tanpa akselerasi GPU perlu menjadi perhatian pada pengembangan selanjutnya. Dibandingkan penelitian yang berfokus murni pada akurasi model penerjemah satu arah, BISINDO.app memosisikan diri sebagai media pembelajaran dua arah: pengguna tidak hanya menerima informasi gestur, tetapi juga mendapat umpan balik langsung atas gerakan yang mereka lakukan sendiri, sejalan dengan prinsip pedagogis latihan-koreksi cepat pada pembelajaran keterampilan motorik. Aspek kolaborasi dengan PUSBISINDO Sulawesi Utara turut membedakan penelitian ini dari mayoritas studi BISINDO sebelumnya yang jarang melibatkan komunitas Tuli secara langsung dalam proses validasi konten, sehingga akurasi teknis yang tinggi dapat diimbangi dengan keaslian representasi bahasa isyarat yang diajarkan.

Dari sisi dampak sosial, BISINDO.app berpotensi meningkatkan aksesibilitas pembelajaran BISINDO bagi keluarga teman Tuli, tenaga pendidik di sekolah inklusi, dan masyarakat umum yang sebelumnya tidak memiliki akses ke kelas tatap muka BISINDO di Sulawesi Utara. Keterbatasan yang teridentifikasi meliputi akurasi model angka dinamis yang masih di bawah 90% akibat keterbatasan variasi subjek dalam dataset, modul kata dan salam yang masih berjalan dalam *Guide Mode* menunggu integrasi alur kerja sekuensial, serta cakupan pengujian *usability* yang terbatas pada 13 responden.

Meskipun demikian, hasil pengujian *usability* yang konsisten tinggi pada kedua kelompok responden mengindikasikan bahwa desain antarmuka dan mekanisme umpan balik BISINDO.app cukup intuitif untuk digunakan tanpa pelatihan khusus, sehingga berpotensi diadopsi sebagai pelengkap materi pembelajaran BISINDO formal di



lingkungan pendidikan inklusif maupun sebagai alat bantu mandiri bagi masyarakat umum yang ingin mempelajari BISINDO secara swadaya. Skala penelitian yang masih terbatas pada satu politeknik dan satu komunitas mitra juga membuka peluang replikasi pengujian pada kelompok pengguna yang lebih beragam di wilayah lain untuk memperkuat generalisasi temuan usability ini.

4. KESIMPULAN

Penelitian ini berhasil mengembangkan BISINDO.app, aplikasi web pembelajaran BISINDO interaktif yang mengintegrasikan MediaPipe Hand Landmarker dengan model klasifikasi low-parameter (Conv1D untuk gestur statis, BiLSTM untuk gestur dinamis) yang berjalan sepenuhnya di sisi klien melalui TensorFlow.js. Algoritma stabilisasi prediksi berbasis ambang keyakinan, konsistensi tick, dan penghalusan EMA terbukti menghasilkan umpan balik yang stabil dan responsif, sementara pengujian usability terhadap 13 responden menghasilkan skor SUS rata-rata 83,19 (kategori Excellent). Kontribusi utama penelitian ini adalah integrasi model client-side untuk gestur statis dan dinamis dalam satu platform pembelajaran, algoritma stabilisasi berbasis pemeriksaan ganda, serta validasi konten bersama PUSBISINDO Sulawesi Utara. Keterbatasan yang teridentifikasi meliputi akurasi model angka dinamis yang masih di bawah 90%, kebutuhan integrasi inferensi sekuensial penuh pada modul kata dinamis di antarmuka web, skema pembagian data yang belum sepenuhnya subject-independent, serta cakupan pengujian usability yang masih terbatas pada satu institusi dan satu komunitas mitra. Pengembangan selanjutnya diarahkan pada perluasan dataset dengan lebih banyak subjek, evaluasi pada test set independen, integrasi alur kerja inferensi sekuensial penuh pada peramban web untuk seluruh modul dinamis, serta replikasi pendekatan ini untuk bahasa isyarat daerah lain di Indonesia.

REFERENCES

- [1] Z. R. Jannah, "Perbedaan Bahasa Isyarat: BISINDO VS SIBI | Lembaga Bahasa Internasional FIB UI," Lembaga Bahasa Internasional FIB UI. Accessed: Aug. 18, 2026. [Online]. Available: <https://lbifib.ui.ac.id/id/blog/artikel/perbedaan-bahasa-isyarat-bisindo-vs-sibi>
- [2] B. Asriandhini and C. H. Rahmawati, "Bahasa Isyarat Indonesia Sebagai Konstruksi Identitas Dan Citra Sosial Tuli Di Purwokerto," *JRK (Jurnal Riset Komunikasi)*, vol. 12, no. 1, 2021, doi: 10.31506/jrk.v12i1.10059.
- [3] Direktorat Jenderal Pendidikan Vokasi, "Jadi Alat Komunikasi Kaum Tunarungu, Yuk Mengenal Bahasa Isyarat di Indonesia | Direktorat Jenderal Pendidikan Vokasi PKPLK Kemendikdasmen," Kementerian Pendidikan Dasar dan Menengah. Accessed: Aug. 18, 2026. [Online]. Available: <https://vokasi.kemendikdasmen.go.id/read/b/jadi-alat-komunikasi-kaum-tunarungu-yuk-mengenal-bahasa-isyarat-di-indonesia>
- [4] R. I. Borman and B. Priyopradono, "Implementasi Penerjemah Bahasa Isyarat Pada Bahasa Isyarat Indonesia (BISINDO) Dengan Metode Principal Component Analysis (PCA)," *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 3, no. 1, 2018, doi: 10.30591/jpit.v3i1.631.
- [5] W. Ahmad priadiyatna, H. Hudiono, and A. Rasyid, "Rancang Bangun Sarung Tangan Pintar Penerjemah Bahasa Isyarat Indonesia (Bisindo) Berbasis Iot," *Jurnal Jartel: Jurnal Jaringan Telekomunikasi*, vol. 10, no. 4, 2020, doi: 10.33795/jartel.v10i4.21.
- [6] M. Susanty, R. Z. Fadillah, and A. Irawan, "Model Penerjemah Bahasa Isyarat Indonesia (BISINDO) Menggunakan Pendekatan Transfer Learning," *PETIR*, vol. 15, no. 1, pp. 1–9, Dec. 2021, doi: 10.33322/PETIR.V15I1.1289.
- [7] L. Arisandi and B. Satya, "Sistem Klarifikasi Bahasa Isyarat Indonesia (Bisindo) Dengan Menggunakan Algoritma Convolutional Neural Network," *Jurnal Sistem Cerdas*, vol. 5, no. 3, 2022, doi: 10.37396/jsc.v5i3.262.
- [8] A. Bayu Pangestu, M. Rafi Muttaqin, and M. Agus Sunandar, "Sistem Deteksi Bahasa Isyarat Indonesia (Bisindo) Menggunakan Algoritma You Only Look Once (YOLO)v8," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 5, 2024, doi: 10.36040/jati.v8i5.10833.
- [9] I. G. A. O. Aryananda and F. Samopa, "Comparison of the Accuracy of The Bahasa Isyarat Indonesia (BISINDO) Detection System Using CNN and RNN Algorithm for Implementation on Android," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 3, 2024, doi: 10.57152/malcom.v4i3.1465.
- [10] A. Aljabar and Suhartito, "BISINDO (Bahasa isyarat indonesia) sign language recognition using CNN and LSTM," *Advances in Science, Technology and Engineering Systems*, vol. 5, no. 5, 2020, doi: 10.25046/AJ050535.
- [11] A. F. Deleviar, I. Oktaviani, and H. 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, 2025, doi: 10.29408/jit.v8i1.26117.
- [12] K. L. Wiguna and Rojali, "Sentence-Level Indonesian Sign Language (BISINDO) Recognition Using 3D CNN-LSTM and 3D CNN-BiLSTM Models," *International Journal of Advanced Computer Science and Applications*, vol. 16, no. 9, 2025, doi: 10.14569/IJACSA.2025.0160941.
- [13] A. K. Fadzli and M. Rahardi, "Comparative Analysis of MobileNetV3 and EfficientNetv2B0 in BISINDO Hand Sign Recognition Using MediaPipe Landmarks," *Journal of Applied Informatics and Computing*, vol. 10, no. 1, pp. 737–746, Feb. 2026, doi: 10.30871/JAIC.V10I1.11878.
- [14] I. Rizka Fadhillah, M. Muharrom Al Haromainy, and H. Maulana, "Implementasi Model Transfer Learning Efficientnet Untuk Pendeteksian Bahasa Isyarat Indonesia (BISINDO) Pada Perangkat Android," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 4, 2024, doi: 10.36040/jati.v8i4.10463.
- [15] N. Renaningtias, F. Putra Utama, and A. N. A. Sobri, "Detection System Indonesian Sign Language (BISINDO) in Video with YOLOv7," *JSAT (Journal Scientific and Applied Informatics)*, vol. 8, no. 1, pp. 1–8, Jan. 2025, doi: 10.36085/JSAT.V8I1.7067.



- [16] T. M. A. Afwan, R. Gernowo, and H. A. Wibawa, "Deep Learning-Based Recognition of Indonesian Sign Language (BISINDO) Alphabetic Gestures Using Skeletal Feature Extraction and LSTM," *Jurnal Teknik Informatika (Jutif)*, vol. 7, no. 2, pp. 903–925, Apr. 2026, doi: 10.52436/1.JUTIF.2026.7.2.5337.
- [17] S. A. Nurpiena, E. S. Wihidayat, and A. Budianto, "Developing Indonesia Sign Language (BISINDO) Application with Android Based for Learning Sign Language," *Journal of Informatics and Vocational Education*, vol. 4, no. 1, pp. 1–11, Feb. 2021, doi: 10.20961/JOIVE.V4I1.48629.
- [18] R. Setiawan, Y. Yunita, F. F. Rahman, and H. Fahmi, "BISINDO (Bahasa Isyarat Indonesia) Sign Language Recognition Using Deep Learning," *IT for Society*, vol. 9, no. 1, Mar. 2024, doi: 10.33021/ITFS.V9I1.5076.
- [19] F. I. Hadinata and S. A. Sanjaya, "BISINDO Sign Language Recognition: A Systematic Literature Review of Deep Learning Techniques for Image Processing," *The Indonesian Journal of Computer Science*, vol. 12, no. 6, p. 3281, Dec. 2023, doi: 10.33022/IJCS.V12I6.3539.
- [20] G. O. Kindy, G. Leonali, and H. Lucky, "Word-Level BISINDO: A Novel Video Indonesian Sign Language Dataset and Baseline Methods," *Procedia Comput. Sci.*, vol. 269, pp. 249–258, Jan. 2025, doi: 10.1016/J.PROCS.2025.08.277.
- [21] C. Sebastian, J. Limanza, L. Laurentia, J. Harefa, and K. Jingga, "Indonesian Sign Language (BISINDO) Recognition Using Spatially Aware Body Gesture Recognition," *Procedia Comput. Sci.*, vol. 269, pp. 1002–1011, Jan. 2025, doi: 10.1016/J.PROCS.2025.09.042.
- [22] A. Vakunov, C.-L. Chang, F. Zhang, G. Sung, M. Grundmann, and V. Bazarevsky, "MediaPipe Hands: On-device Real-time Hand Tracking," 2020. Accessed: Aug. 18, 2026. [Online]. Available: <https://research.google/pubs/mediapipe-hands-on-device-real-time-hand-tracking/>
- [23] I. D. A. Rachmawati, R. Yunanda, M. F. Hidayat, and P. Wicaksono, "Deep Transfer Learning for Sign Language Image Classification: A Bisindo Dataset Study," *Engineering, Mathematics and Computer Science Journal (EMACS)*, vol. 5, no. 3, 2023, doi: 10.21512/emacsjournal.v5i3.10621.
- [24] A. Almjjally, S. A. Algamdi, N. Aljohani, and M. K. Nour, "Harnessing attention-driven hybrid deep learning with combined feature representation for precise sign language recognition to aid deaf and speech-impaired people," *Sci. Rep.*, vol. 15, no. 1, 2025, doi: 10.1038/s41598-025-15109-2.
- [25] A. Baihan, A. I. Alutaibi, M. Alshehri, and S. K. Sharma, "Sign language recognition using modified deep learning network and hybrid optimization: a hybrid optimizer (HO) based optimized CNNs-LSTM approach," *Sci. Rep.*, vol. 14, no. 1, 2024, doi: 10.1038/s41598-024-76174-7.
- [26] A. Yuan and A. Vakunov, "Face and hand tracking in the browser with MediaPipe and TensorFlow.js The TensorFlow Blog," The TensorFlow Blog. Accessed: Aug. 18, 2026. [Online]. Available: <https://blog.tensorflow.org/2020/03/face-and-hand-tracking-in-browser-with-mediapipe-and-tensorflowjs.html>
- [27] Google AI Edge, "Gesture recognition task guide | Google AI Edge | Google for Developers," Google for Developers. Accessed: Aug. 18, 2026. [Online]. Available: https://developers.google.com/edge/mediapipe/solutions/vision/gesture_recognizer
- [28] A. Desai *et al.*, "ASL Citizen: A Community-Sourced Dataset for Advancing Isolated Sign Language Recognition," *Adv. Neural Inf. Process. Syst.*, vol. 36, pp. 76893–76907, Dec. 2023, doi: 10.52202/075280-3360.
- [29] R. Z. Fadillah, A. Irawan, and M. Susanty, "Data Augmentasi Untuk Mengatasi Keterbatasan Data Pada Model Penerjemah Bahasa Isyarat Indonesia (BISINDO)," *Jurnal Informatika*, vol. 8, no. 2, pp. 208–214, Sep. 2021, doi: 10.31294/JI.V8I2.10768.
- [30] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [31] J. Brooke, "SUS: A quick and dirty usability scale," *Usability Eval. Ind.*, vol. 189, pp. 207–212, Nov. 1996.
- [32] A. Bangor, P. Kortum, and J. Miller, "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale," *J. Usability Stud.*, vol. 4, pp. 114–123, Apr. 2009.
- [33] E. L. Kelana, M. R. Anshori Prasetya, Mambang, and M. Zulfadhilah, "Integrating the CNN Model with the Web for Indonesian Sign Language (BISINDO) Recognition," *Journal of Applied Informatics and Computing*, vol. 9, no. 3, 2025, doi: 10.30871/jaic.v9i3.9345.