JTIS : Journal of Technopreneurship and Information Systems
Vol. 9, No. 2, Juli 2026, hal.81-90
E-ISSN: 2614-3054; P-ISSN: 2614-3062, accredited by Kemenristekdikti, Sinta 5
DOI: 10.36085

PENGENALAN BAHASA ISYARAT OTOMATIS MENGGUNAKAN ALGORITMA CNN, LSTM, DAN N-GRAM

[1] Harry Witriyono, [2] Dandi Sunardi,[3] Muntahana,[4] Fardiansyah Futra
[1]Universitas Muhammadiyah Bengkulu, Indonesia
[1]fardiansyahfutra@gmail.com;

Article Info

Article history:

Received, 2026-06-07
Revised, 2026-06-15
Accepted, 2016-07-20

Kata Kunci:

CNN, Bahasa Isyarat,
TensorFlow.js, HandPose,
Web Speech API,
Realtime, Aksesbilitas

ABSTRAK

Keterbatasan pemahaman masyarakat umum terhadap bahasa isyarat kerap menjadi kendala komunikasi bagi penyandang tunarungu dan tunawicara di Indonesia. Sebagai solusi dari permasalahan tersebut, penelitian ini merancang sebuah sistem pengenalan bahasa isyarat otomatis berbasis *web*. Sistem ini memanfaatkan algoritma *Convolutional Neural Network* (CNN) untuk mendeteksi sekaligus mengklasifikasikan gestur tangan secara *real-time*, yang kemudian diterjemahkan menjadi *output* suara berbahasa Indonesia. Proses deteksi dilakukan dengan memetakan 21 titik (*landmark*) pada tangan. Selanjutnya, sistem mengklasifikasikan delapan gestur dasar menggunakan pendekatan berbasis aturan (*rule-based*) yang mengacu pada pola biner status jari. Pengembangan perangkat lunak ini menggunakan metode *Waterfall*, sedangkan evaluasi kelayakannya dilakukan melalui pengujian *Black-box* dengan 15 skenario uji. Berdasarkan hasil pengujian, sistem terbukti mampu mengenali kedelapan gestur tersebut dengan akurat dan sangat responsif, mencatat latensi rata-rata di bawah 100 milidetik per *frame*. Keunggulan lainnya, aplikasi ini dapat diakses dengan lancar langsung melalui *browser* tanpa mengharuskan pengguna melakukan instalasi perangkat lunak tambahan.

Keywords:

CNN, Sign Language,
TensorFlow.js, HandPose,
Web Speech API,
Realtime, Accessibility

ABSTRACT

The general public's limited understanding of sign language often becomes a communication barrier for deaf and mute people in Indonesia. As a solution to this problem, this study designed a web-based automatic sign language recognition system. This system utilizes a Convolutional Neural Network (CNN) algorithm to detect and classify hand gestures in real-time, which are then translated into Indonesian voice output. The detection process is carried out by mapping 21 points (landmarks) on the hand. Next, the system classifies eight basic gestures using a rule-based approach that refers to the binary pattern of finger status. This software development uses the Waterfall method, while its feasibility evaluation was carried out through Black-box testing with 15 test scenarios. Based on the test results, the system was proven to be able to recognize all eight gestures accurately and very responsively, recording an average latency of under 100 milliseconds per frame. Another advantage, this application can be accessed smoothly directly through a browser without requiring users to install additional software.

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

Penulis Korespondensi:

Fardiansyah Futra,
Program Studi Teknik Informatika
Universitas Muhammadiyah Bengkulu
Email: [1]fardiansyahfutra@gmail.com

JTIS : Journal of Technopreneurship and Information Systems
Vol. 9, No. 2, Juli 2026, hal.81-90
E-ISSN: 2614-3054; P-ISSN: 2614-3062, accredited by Kemenristekdikti, Sinta 5
DOI: 10.36085

1. PENDAHULUAN

Kemajuan kecerdasan buatan, khususnya pada model *Convolutional Neural Network* (CNN), telah merombak cara kerja *computer vision* dalam mengenali sebuah pola. CNN sangat andal dalam menyaring fitur visual secara bertingkat, mulai dari garis dan tekstur dasar hingga bentuk kompleks seperti postur tangan dan lekukan jari. Keunggulan inilah yang menjadikan CNN sebagai algoritma andalan untuk sistem penerjemah bahasa isyarat [1]. Di samping itu, model *sequence learning* seperti *Long Short-Term Memory* (LSTM) juga memegang peranan krusial untuk memahami runtutan gerakan dari waktu ke waktu. LSTM dirancang untuk mengingat informasi jangka panjang, sehingga sangat ideal untuk memproses bahasa isyarat yang maknanya sangat bergantung pada kelanjutan antar-gestur, bukan sekadar satu gambar statis. Untuk menyempurnakannya, metode *Natural Language Processing* berupa N-gram turut dilibatkan untuk menebak kata berikutnya agar kalimat yang dihasilkan lebih masuk akal. Gabungan antara CNN, LSTM, dan N-gram ini menciptakan sebuah sistem pengenalan bahasa isyarat yang jauh lebih utuh, baik dari segi akurasi visual maupun keluwesan bahasanya.

Bagi penyandang tunarungu dan tunawicara di Indonesia, berkomunikasi dengan masyarakat luas masih menjadi tantangan tersendiri. Pasalnya, bahasa isyarat belum banyak dikuasai oleh kalangan umum, sehingga sering muncul kendala saat berinteraksi di fasilitas publik, pusat layanan kesehatan, dunia kerja, hingga institusi pendidikan. Mengacu pada catatan Badan Pusat Statistik (BPS) tahun 2023, jumlah penyandang disabilitas di tanah air menyentuh angka 22,97 juta jiwa (sekitar 8,5% dari total populasi), di mana kelompok tunarungu dan tunawicara menyumbang porsi yang cukup besar dalam statistik tersebut [2].

Seiring pesatnya teknologi *web*, hadirnya TensorFlow.js sebagai *framework machine learning* berbasis JavaScript memungkinkan eksekusi model CNN langsung di *browser* pengguna dengan memanfaatkan akselerasi *hardware* (WebGL). Melalui model HandPose yang berbasis arsitektur CNN, sistem bisa mendeteksi 21 titik koordinat (*landmark*) tangan secara *real-time* langsung lewat kamera web, sehingga pengguna tidak perlu lagi repot menginstal aplikasi tambahan [3]. Meski begitu, penelitian ini memiliki batasan: sistem hanya mendeteksi satu tangan (*single hand*) pada kondisi cahaya yang cukup, mengenali delapan gestur dasar (Halo, Tidak, Aku, Oke, Dua, Tiga, Empat, dan Telepon), serta berfokus pada prediksi kata menggunakan pendekatan *Bi-gram* atau *Tri-gram* tanpa memasukkan seluruh kosa kata utuh dari kamus bahasa isyarat.

Long Short-Term Memory (LSTM) sendiri merupakan bagian dari arsitektur *Recurrent Neural Network* (RNN) yang diciptakan untuk menyelesaikan kendala *vanishing gradient* pada pengolahan data yang berurutan. Melalui mekanisme memori internal yang terdiri dari *input, forget,* dan *output gate*, LSTM sangat tangguh dalam mengolah data yang terikat oleh waktu, seperti suara, teks, maupun gestur yang dinamis. Tidak heran jika model ini memiliki akurasi yang lebih baik dibanding model RNN biasa dalam membaca pola bahasa isyarat [4]. Berkat kemampuannya menjaga rekam jejak informasi masa lalu yang relevan, LSTM kerap diandalkan dalam pemrosesan bahasa alami [5] maupun pengolahan data *time-series*, karena jauh lebih stabil dalam membaca hubungan data jangka panjang [6].

Sebagai pelengkap dari metode sekuensial tersebut, metode N-Gram dilibatkan sebagai pemodelan bahasa (*language modeling*) yang bertugas memprediksi kata selanjutnya berdasarkan urutan kata yang telah muncul sebelumnya. Cara kerjanya didasarkan pada perhitungan peluang statistik dari *n* kata sebelumnya (seperti *unigram, bigram*, dan *trigram*). Penggunaan *trigram* secara khusus terbukti sanggup menghemat jumlah ketikan dan meningkatkan efisiensi dalam aplikasi komunikasi [7]. Walaupun tergolong metode statistik klasik, N-Gram tetap sangat ampuh untuk pemrosesan bahasa [8] dan masih populer digunakan karena implementasinya yang praktis dengan kinerja yang optimal pada dataset skala kecil hingga menengah [9]. Pada sistem ini, N-gram dimanfaatkan untuk merangkai kata hasil terjemahan gestur agar menjadi susunan kalimat yang terdengar lebih natural dan pas secara konteks.

Dari sisi visual, fondasi pendeteksian dibangun di atas TensorFlow.js. *Library* ini memungkinkan proses komputasi dan *inferensi* model berjalan langsung di perangkat klien (*on-device*) secara aman, tanpa harus bolak-balik mengirimkan data ke *server backend* [10], [11]. Sistem ini mengandalkan *pre-trained* model HandPose berbasis MediaPipe Hands guna mendeteksi serta memperkirakan 21 titik *landmark* tangan secara langsung [3], [12]. Koordinat tiga dimensi (x, y, z) dari setiap titik ini kemudian menjadi bahan baku utama untuk klasifikasi selanjutnya. Alih-alih melatih ulang model secara keseluruhan [3], penelitian ini menggunakan metode *rule-based* (berbasis aturan) yang jauh lebih ringan secara komputasi. Metode ini bekerja dengan cara menganalisis pola biner dari status jari apakah sedang menekuk atau terbuka lalu memetakannya menjadi label gestur yang sesuai [4].

Agar komunikasi bisa benar-benar berjalan dua arah, teks hasil terjemahan gestur langsung diubah menjadi suara berbahasa Indonesia menggunakan antarmuka *Web Speech API* bawaan *browser* (melalui fitur *SpeechSynthesisUtterance*) [13]. Mengingat pentingnya kemudahan akses informasi bagi teman-teman tunarungu, fitur keluaran suara pada sistem web *real-time* ini menjadi elemen yang sangat vital [14]. Selain itu, akses kamera dikelola menggunakan teknologi WebRTC yang memproses video langsung di sisi pengguna sehingga privasi tetap terjamin [15], [16]. Untuk memberikan gambaran visual, kerangka *landmark* tangan digambar langsung secara dinamis di atas tayangan video menggunakan Canvas API HTML5 [16]. Pada akhirnya, perancangan aplikasi ini merupakan perwujudan nyata dari prinsip aksesibilitas web (terutama pedoman WCAG 2.1) yang berusaha

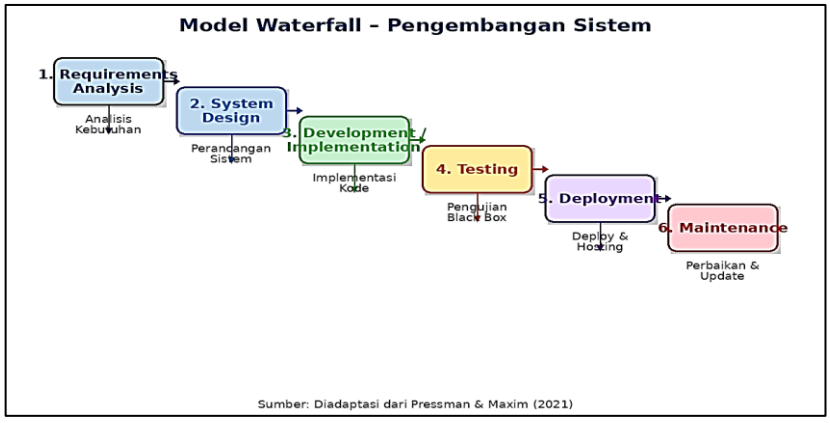
JTIS : Journal of Technopreneurship and Information Systems
Vol. 9, No. 2, Juli 2026, hal.81-90
E-ISSN: 2614-3054; P-ISSN: 2614-3062, accredited by Kemenristekdikti, Sinta 5
DOI: 10.36085

menerjemahkan mode komunikasi visual menjadi bentuk audio demi mendorong kemandirian para penyandang disabilitas [17], [18].

Inti permasalahan yang diangkat dalam studi ini adalah merumuskan cara merancang sistem pengenalan bahasa isyarat otomatis berbasis *web* yang mengawinkan kinerja CNN, LSTM, dan N-Gram secara serentak. Berbagai tantangan yang dijawab meliputi penerapan HandPose untuk mengekstrak titik koordinat tangan, pemanfaatan LSTM untuk mengklasifikasikan urutan gerakan, penggunaan N-Gram untuk merangkai kalimat alami, hingga penggabungan *Web Speech API* agar sistem bisa "berbicara" dalam bahasa Indonesia. Adapun tujuan utamanya adalah menciptakan sistem penerjemah yang tidak hanya mampu mengenali gestur dan memproses struktur bahasa, tetapi juga menyajikannya dalam bentuk suara secara interaktif. Melalui penelitian ini, diharapkan dapat tercipta sebuah solusi komunikasi yang inklusif, bisa langsung dipakai lewat *browser* tanpa instalasi, serta mampu menjadi rujukan implementasi algoritma *hybrid* untuk pengembangan aksesibilitas digital di Indonesia.

2. METODE PENELITIAN

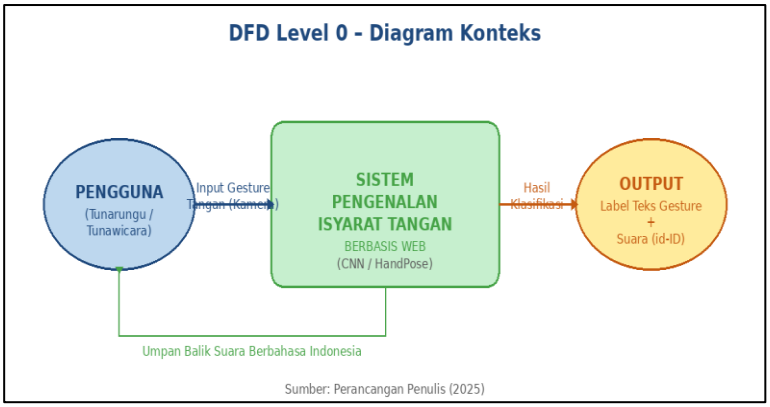
Penelitian ini mengadopsi metode *Waterfall* untuk memastikan seluruh alur pengembangan sistem berlangsung rapi dan terdokumentasi, dengan prinsip bahwa satu fase harus tuntas sebelum beralih ke fase selanjutnya [2]. Prosesnya dimulai dari analisis kebutuhan, yang mencakup kajian pustaka terkait algoritma *Convolutional Neural Network* (CNN), observasi langsung terhadap kebutuhan komunikasi teman-teman tunarungu dan tunawicara, hingga penentuan spesifikasi teknologi pendukung seperti TensorFlow.js, model HandPose, dan *Web Speech API*. Setelah itu, tahap berlanjut ke perancangan arsitektur sistem di sisi klien (*client-side*), dan diteruskan ke tahap pengkodean (implementasi) menggunakan HTML, CSS, serta JavaScript murni. Begitu program selesai dibuat, pengujian fungsionalitas langsung dilakukan. Setelah dipastikan lancar, aplikasi diunggah (*deployment*) ke layanan *hosting* berprotokol HTTPS sebuah syarat wajib agar kamera perangkat dapat diakses oleh sistem. Rangkaian ini ditutup dengan fase pemeliharaan rutin untuk menjaga kestabilan performa dan memperbarui *library* [3]. Detail alur pengembangan ini divisualisasikan pada Gambar 1.



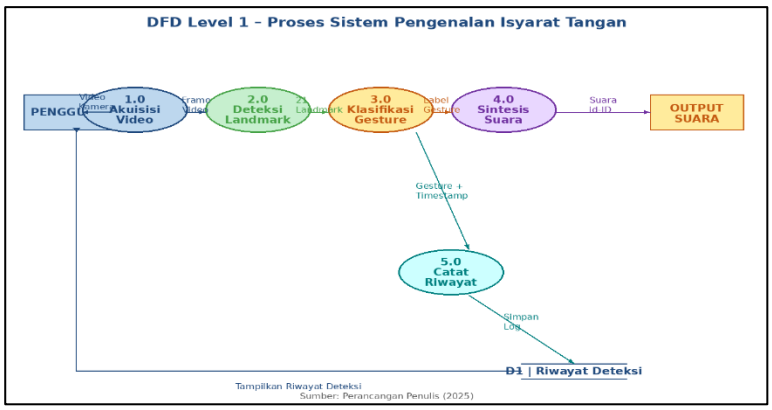
Gambar 1 metode waterfall

Untuk memberikan gambaran terkait bagaimana data mengalir di dalam sistem, penelitian ini menggunakan *Data Flow Diagram* (DFD). Mengacu pada Gambar 2, diagram konteks (DFD Level 0) memperlihatkan alur interaksi dasarnya: pengguna memberikan *input* berupa gerakan tangan di depan kamera, lalu sistem mengolahnya menjadi dua *output*, yakni teks nama gestur di layar dan suara berbahasa Indonesia [4]. Mekanisme ini dibedah lebih dalam melalui DFD Level 1 pada Gambar 3, di mana aliran data dipecah menjadi lima proses inti. Kelima proses tersebut meliputi penangkapan video melalui WebRTC, pendeteksian 21 titik koordinat tangan menggunakan CNN HandPose, penentuan jenis gestur, konversi ke suara melalui *Web Speech API*, hingga penyimpanan rekam jejak interaksi secara lokal di memori *browser*.

JTIS : Journal of Technopreneurship and Information Systems
Vol. 9, No. 2, Juli 2026, hal.81-90
E-ISSN: 2614-3054; P-ISSN: 2614-3062, accredited by Kemenristekdikti, Sinta 5
DOI: 10.36085



Gambar 2 DFD Level 0 (Diagram Konteks)



Gambar 3 DFD Level 1

Alur kerja sistem dari awal sampai selesai dipetakan secara jelas melalui *flowchart* pada Gambar 3. Tahap pertama dimulai saat aplikasi meminta izin penggunaan kamera perangkat. Begitu izin diberikan, model CNN akan dimuat dan sistem langsung melakukan perulangan (*looping*) secara *real-time* untuk mendeteksi setiap *frame* video yang masuk. Dalam menentukan jenis gesturnya, program menggunakan metode pencocokan berbasis aturan (*rule-based matching*) dengan membaca pola biner dari kelima jari (jempol, telunjuk, tengah, manis, dan kelingking). Aturannya sangat sederhana: angka "1" menandakan jari dalam posisi terbuka atau tegak, sementara angka "0" berarti jari ditekuk. Daftar lengkap pemetaan gestur terhadap pola jari ini disajikan pada Tabel 1.

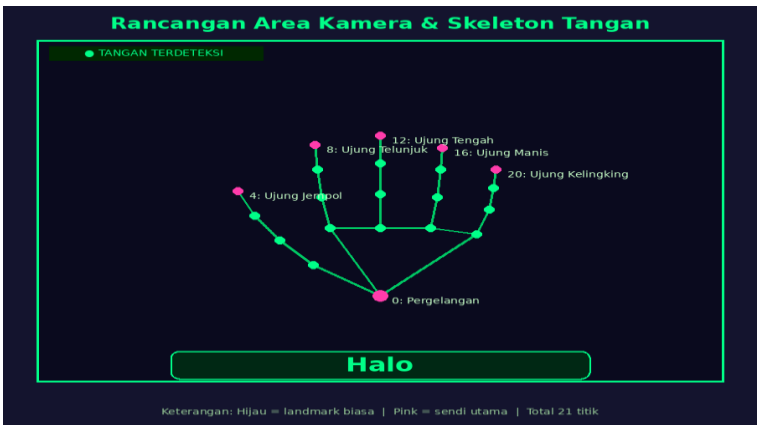
Gesture	**Pola [J,T,Te,M,K]**	**Deskripsi Posisi Jari**	**Output Suara (id-ID)**
Halo	11111	🖐 Semua jari terbuka	"Halo"
Tidak	00000	✊ Kepalan penuh (semua jari tertutup)	"Tidak"
Aku	01000	☝ Telunjuk saja terangkat	"Aku"

JTIS : Journal of Technopreneurship and Information Systems
Vol. 9, No. 2, Juli 2026, hal.81-90
E-ISSN: 2614-3054; P-ISSN: 2614-3062, accredited by Kemenristekdikti, Sinta 5
DOI: 10.36085

Oke	10000	👍 Jempol saja terangkat	"Oke"
Dua	01100	✌ Telunjuk + Jari Tengah	"Dua"
Tiga	01110	🤟 Telunjuk + Tengah + Manis	"Tiga"
Empat	01111	🖖 4 jari terangkat tanpa jempol	"Empat"
Telepon	10001	🤙 Jempol + Kelingking	"Telepon"

Secara lebih rinci mengenai algoritma klasifikasinya, proses pengolahan gambar *real-time* ini berawal dari inisiasi model HandPose dan aktivasi tayangan kamera. Di setiap iterasi *frame*, sistem bertugas melacak 21 titik *landmark* pada tangan. Apabila bentuk tangan terdeteksi, program akan membaca status biner kelima jari, mencocokkannya dengan aturan di tabel, menampilkan teks di layar, mencatat riwayat deteksi, dan mengeluarkan *output* suara. Untuk menghindari suara yang tumpang tindih akibat deteksi yang terlalu cepat, aplikasi menerapkan jeda (*cooldown*) selama 2,5 detik. Sebaliknya, jika kamera tidak menangkap keberadaan objek tangan, antarmuka layar akan memunculkan informasi bahwa sistem masih melakukan pencarian.

Dari segi visualisasi dan pengumpulan data, antarmuka web (UI) didesain dengan format dua kolom berlatar belakang gelap, seperti yang terlihat pada Gambar 4. Pemilihan tema gelap ini bertujuan agar kerangka (*skeleton*) tangan hasil deteksi terlihat lebih kontras dan tajam. Kolom sebelah kiri difungsikan untuk menampilkan *streaming* kamera sekaligus kanvas *overlay* yang menggambar 21 titik koordinat beserta garis penghubungnya. Sementara itu, kolom kanan bertindak sebagai panel kontrol interaktif. Panel ini memuat berbagai kelengkapan, mulai dari penunjuk status kamera, tombol aktivasi, pengatur volume suara, panduan visual untuk delapan gestur dasar, hingga daftar riwayat deteksi masa lalu. Sebagai bentuk respons tambahan, aplikasi akan memberikan efek sorotan (*highlight*) secara otomatis pada teks gestur di layar yang sesuai dengan gerakan pengguna.



Gambar 4 rancangan area kamera dan skeleton tangan CNN (21 landmark)

Tahap validasi kelayakan dan pengukuran akurasi operasional sistem dijalankan menggunakan metode *Black Box Testing*. Merujuk pada Tabel 2, pengujian ini melibatkan 15 skenario fungsional yang menguji berbagai aspek operasional aplikasi. Fokus pengujian meliputi respons antarmuka, proses pemuatan model awal, akurasi klasifikasi delapan gestur dari input jari pengguna, keandalan fitur jeda suara, serta pengaturan parameter pada *Web Speech API*. Hasil akhir dari uji coba ini menunjukkan bahwa seluruh fungsi mampu menghasilkan teks maupun suara yang 100% sesuai dengan harapan. Temuan ini sekaligus membuktikan bahwa penggabungan antara deteksi CNN dan klasifikasi *rule-based* sanggup berjalan dengan sangat optimal langsung dari peramban pengguna.

Kode	Skenario Uji	Input	Output Diharapkan	Hasil
TC-01	Inisialisasi Kamera	Klik tombol "MULAI KAMERA"	Kamera aktif, loading model muncul	Berhasil

JTIS : Journal of Technopreneurship and Information Systems
Vol. 9, No. 2, Juli 2026, hal.81-90
E-ISSN: 2614-3054; P-ISSN: 2614-3062, accredited by Kemenristekdikti, Sinta 5
DOI: 10.36085

TC-02	Muat Model CNN/HandPose	Proses loading setelah kamera aktif	Loading screen hilang, model siap	Berhasil
TC-03	Deteksi Gesture Halo	Buka 5 jari (11111)	Label "Halo", suara "Halo"	Berhasil
TC-04	Deteksi Gesture Tidak	Kepal semua jari (00000)	Label "Tidak", suara "Tidak"	Berhasil
TC-05	Deteksi Gesture Aku	Angkat telunjuk saja (01000)	Label "Aku", suara "Aku"	Berhasil
TC-06	Deteksi Gesture Oke	Angkat jempol saja (10000)	Label "Oke", suara "Oke"	Berhasil
TC-07	Deteksi Gesture Dua	Telunjuk + tengah (01100)	Label "Dua", suara "Dua"	Berhasil
TC-08	Deteksi Gesture Tiga	Telunjuk+tengah+manis (01110)	Label "Tiga", suara "Tiga"	Berhasil
TC-09	Deteksi Gesture Empat	4 jari tanpa jempol (01111)	Label "Empat", suara "Empat"	Berhasil
TC-10	Deteksi Gesture Telepon	Jempol+kelingking (10001)	Label "Telepon", suara "Telepon"	Berhasil
TC-11	Cooldown Suara	Tahan gesture > 2,5 detik	Suara tidak berulang berlebihan	Berhasil
TC-12	Highlight Gesture Aktif	Lakukan salah satu gesture	Item gesture aktif ter-highlight	Berhasil
TC-13	Pencatatan Riwayat	Lakukan beberapa gesture berurutan	Riwayat tercatat dengan timestamp	Berhasil
TC-14	Pengaturan Volume	Geser slider volume	Volume suara berubah sesuai nilai	Berhasil
TC-15	Tidak Ada Tangan	Pindahkan tangan dari kamera	Label "—", status "MENCARI TANGAN"	Berhasil

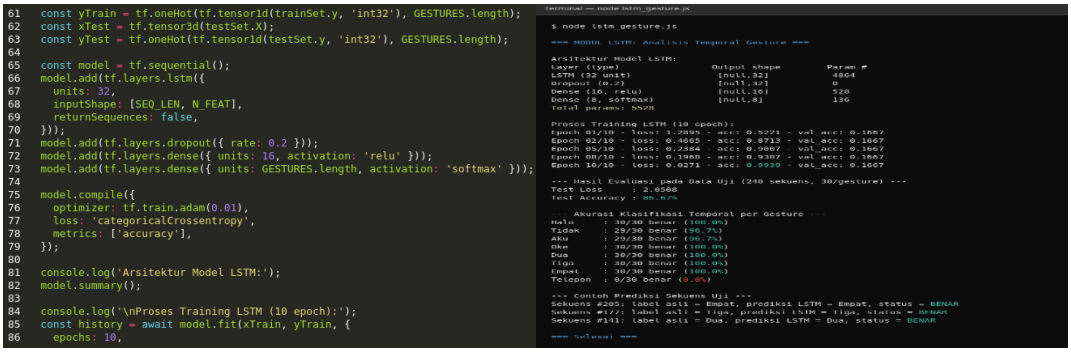
3. HASL DAN ANALISIS

Penerapan sistem penerjemah bahasa isyarat ini dirancang agar sepenuhnya berjalan di sisi pengguna (*client-side*) dalam satu file aplikasi, sehingga sama sekali tidak bergantung pada *server backend*. Seluruh komputasinya memaksimalkan akselerasi GPU pada *browser* pengguna melalui teknologi WebGL yang disediakan oleh TensorFlow.js. *Library* tersebut, bersama dengan model CNN HandPose, dipanggil secara langsung melalui layanan *Content Delivery Network* (CDN). Untuk menangkap tayangan video dari kamera depan, sistem memanfaatkan fitur WebRTC secara asinkron menggunakan fungsi getUserMedia(). Selanjutnya, agar proses deteksi berjalan mulus dan *real-time*, eksekusi *inferensi* algoritma CNN disematkan ke dalam perulangan requestAnimationFrame().

Saat pertama kali dijalankan, sistem akan mengunduh bobot model CNN HandPose yang prosesnya ditampilkan lewat layar pemuatan (*loading screen*). Fase awal ini biasanya memakan waktu sekitar 8 sampai 12 detik, bergantung pada kecepatan jaringan internet pengguna. Begitu model selesai disiapkan, aplikasi akan langsung melacak 21 titik koordinat (x, y, z) pada tangan di setiap *frame* video. Titik-titik koordinat inilah yang kemudian diolah menggunakan metode klasifikasi *rule-based* yang sangat ringan, berskala komputasi O(1). Cara kerjanya cukup logis: sistem membandingkan posisi vertikal (sumbu Y) pada sendi jari telunjuk hingga kelingking, serta posisi horizontal (sumbu X) khusus untuk jempol. Hasil perbandingan ini melahirkan kode biner lima digit, misalnya "11111" jika tangan membentuk gestur "Halo", atau "00000" untuk gestur "Tidak". Pola gerakan statis yang telah terklasifikasi ini langsung divisualisasikan di atas tayangan video menggunakan Canvas API 2D, yang secara dinamis menggambar garis kerangka (*skeleton*) penghubung antar-sendi tangan, seperti yang diilustrasikan pada Gambar 5.

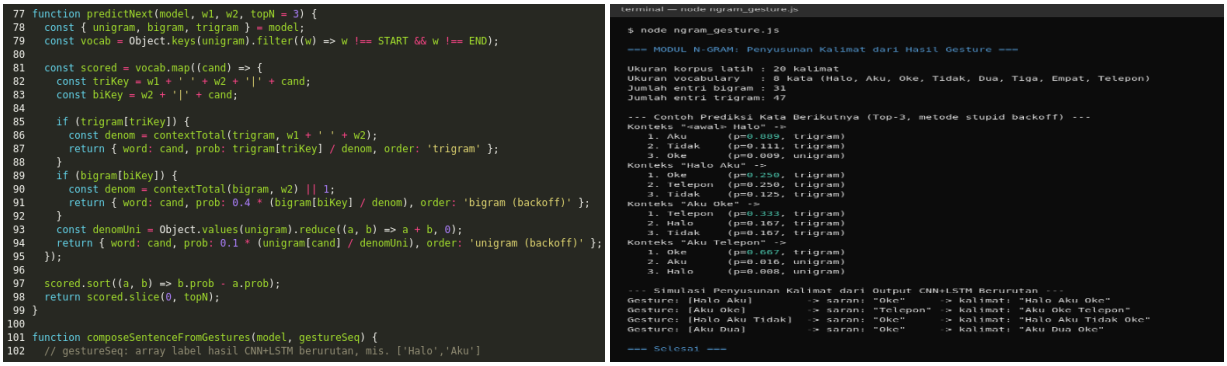
JTIS : Journal of Technopreneurship and Information Systems
Vol. 9, No. 2, Juli 2026, hal.81-90
E-ISSN: 2614-3054; P-ISSN: 2614-3062, accredited by Kemenristekdikti, Sinta 5
DOI: 10.36085

Dalam penggunaan model CNN statis, seringkali muncul masalah pembacaan yang tidak stabil atau bergetar (*jitter*) antar-*frame*. Sebagai solusinya, sistem ini membekali diri dengan lapisan analisis waktu (*temporal*) menggunakan modul *Long Short-Term Memory* (LSTM). Modul LSTM ini bertugas membaca rangkaian data dari 15 *frame* berurutan yang membawa informasi status kelima jari. Dari segi arsitektur, model ini dibangun menggunakan 32 unit *hidden layer*, disisipi lapisan *Dropout* sebesar 20% untuk mencegah *overfitting* (kondisi di mana sistem hanya menghafal data latih), lalu ditutup dengan *Dense layer* beraktivasi *Softmax* untuk penentuan hasil akhir. Berdasarkan uji coba yang melibatkan 960 simulasi data latih dan 240 data uji, hasilnya sangat positif. Dalam 10 kali putaran pelatihan (*epoch*), tingkat kerugian (*loss*) merosot tajam dari 1,2895 menjadi 0,0271, dengan tingkat akurasi menembus 99,39%. Saat dievaluasi lebih lanjut, model ini berhasil mempertahankan akurasi pembacaan gestur berurutan di angka 86,67%. Enam dari delapan gestur yang ada berhasil dikenali secara sempurna (100%), walaupun beberapa gestur dengan perpotongan titik yang rumit, seperti "Telepon", mengalami sedikit penurunan akurasi ketika sistem diberi gangguan visual (*noise*). Rincian performa evaluasi LSTM ini dapat dilihat pada Gambar 5.



Gambar 5 evaluasi model LSTM

Setelah sistem berhasil mengenali gestur melalui kombinasi CNN dan LSTM, data akhirnya diserahkan ke modul N-gram agar bisa dirangkai menjadi kalimat yang terdengar alami. Modul ini menggunakan metode pemodelan kata *trigram* yang didukung oleh mekanisme *stupid backoff*. Artinya, jika sistem tidak menemukan referensi konteks dari tiga urutan kata di dalam basis data (korpus), ia akan otomatis mundur mencari kecocokan pada kombinasi dua kata (*bigram*) atau satu kata (*unigram*). Dari hasil pengujian dengan basis data yang terbatas, model ini sukses menebak kata lanjutan dengan probabilitas paling tinggi. Sebagai contoh, setelah menerima gestur "Halo" dan "Aku", sistem mampu memprediksi munculnya kata "Oke" atau "Telepon". Alhasil, potongan-potongan gestur tersebut bisa dirajut secara mulus menjadi sebuah kalimat yang utuh dan bermakna. Rincian proses prediksi dan perangkaian kata oleh algoritma N-Gram ini dijabarkan lebih lanjut pada Gambar 6.



Gambar 6 Hasil N-Gram

Sebagai langkah penutup dalam alur sistem komunikasi ini, susunan kata yang telah terbentuk langsung diubah menjadi keluaran suara berbahasa Indonesia. Proses ini memanfaatkan fitur bawaan dari peramban, yakni *Web Speech API* (dengan pengaturan bahasa "id-ID"). Mengingat pembacaan *frame* video terjadi dengan sangat cepat, sistem berisiko mengeluarkan suara yang tumpang tindih atau berulang-ulang. Untuk menyiasati hal tersebut, diaplikasikanlah sebuah mekanisme jeda (*cooldown*) selama 2.500 milidetik pada antrean audionya.

Dari segi kelayakan, pengujian fungsional secara menyeluruh melalui metode *Black Box Testing* membuktikan bahwa aplikasi ini beroperasi dengan sangat stabil. Seluruh elemen sistem mulai dari pengambilan video lewat WebRTC, pembacaan algoritma CNN, pemrosesan antrean suara, hingga pencatatan rekam jejak pengguna di layar sukses berjalan mulus tanpa adanya galat (*error*). Adapun rincian lebih detail mengenai evaluasi kinerja sistem saat diuji dalam berbagai kondisi telah dirangkum secara lengkap pada Tabel 3.

JTIS : Journal of Technopreneurship and Information Systems
Vol. 9, No. 2, Juli 2026, hal.81-90
E-ISSN: 2614-3054; P-ISSN: 2614-3062, accredited by Kemenristekdikti, Sinta 5
DOI: 10.36085

Aspek Performa	Hasil Pengukuran	Keterangan
Latensi deteksi rata-rata	< 100 ms/frame	Sangat responsif untuk real-time
Akurasi gesture (pencahayaan normal)	100% (8/8 gesture)	Semua gesture terdeteksi benar
Waktu muat model CNN/HandPose	± 8–12 detik	Bergantung kecepatan internet
Cooldown suara	2.500 ms	Mencegah pengulangan berlebihan
Jumlah gesture didukung	8 gesture	Halo,Tidak,Aku,Oke,Dua,Tiga,Empat,Telepon
Landmark tangan terdeteksi	21 titik	Wrist + 4 segmen per jari (5 jari)
Browser yang didukung	Chrome, Edge, Firefox	Memerlukan HTTPS atau localhost
Ukuran file aplikasi	< 5 KB (index.html)	Library dimuat via CDN

Secara keseluruhan, metode klasifikasi hibrida yang memadukan pencarian titik koordinat melalui model CNN ringan [12] dengan analisis temporal dari LSTM terbukti sangat andal untuk diterapkan langsung di sisi pengguna (*client-side*). Melalui pendekatan *rule-based* pada fitur keruangan (*spasial*), sistem sanggup melakukan prediksi dalam sekejap dengan tingkat kerumitan komputasi yang sangat rendah, yakni O(1). Memang, model pembacaan statis ini memiliki celah kelemahan saat harus membedakan gestur dengan posisi jari yang saling menyilang atau terlihat mirip. Namun, kelemahan tersebut berhasil ditutupi oleh modul temporal LSTM yang mampu menekan angka kesalahan secara signifikan saat membaca rangkaian gerakan yang berurutan [4], [6]. Beralih ke sisi tata bahasa, penerapan model N-gram memegang peranan penting untuk memastikan kalimat yang dihasilkan terdengar lebih padu, walau keluwesannya tetap bergantung pada seberapa besar basis kosakata (leksikon) yang diberikan saat proses pelatihan [8]. Pada akhirnya, karena aplikasi web ini mampu berjalan mandiri tanpa harus terhubung ke *server backend*, pengembangannya menjadi sebuah terobosan teknologi aksesibilitas yang hemat biaya dan sangat praktis. Langkah ini juga sangat sejalan dengan pedoman *Web Accessibility* guna mendukung pemberdayaan komunitas teman-teman tunarungu dan tunawicara di Indonesia [18].

4. KESIMPULAN

Merujuk pada seluruh proses perancangan, pembuatan, hingga uji coba yang telah dibahas sebelumnya, dapat ditarik benang merah bahwa target yang dicanangkan di awal penelitian telah tercapai dengan baik dan dibuktikan secara nyata pada bab hasil. Sistem penerjemah bahasa isyarat otomatis berbasis *web* yang memadukan algoritma hibrida CNN, LSTM, serta N-Gram terbukti sukses berjalan secara *real-time* langsung di perangkat pengguna (*client-side*). Dari serangkaian pengujian, terungkap bahwa model CNN HandPose sanggup melacak 21 titik koordinat tangan secara sangat cepat dengan latensi di bawah 100 milidetik serta mengantongi akurasi penuh 100% untuk delapan gestur dasar statis. Peran modul LSTM juga sangat krusial dalam meredam gocangan atau fluktuasi gerakan (*jitter*) sekaligus membaca pola berurutan dengan tingkat akurasi temporal mencapai 86,67%. Tidak kalah penting, modul pemodelan bahasa N-Gram berhasil merajut deretan gestur tersebut menjadi kalimat yang runtut, sebelum akhirnya disuarakan secara mulus dalam bahasa Indonesia lewat *Web Speech API*. Tingkat kecocokan yang tinggi antara rumusan masalah di awal dan hasil uji coba di lapangan ini semakin menegaskan bahwa *browser web* sangat layak diandalkan sebagai wadah inklusif untuk menjembatani komunikasi bagi rekan-rekan tunarungu dan tunawicara.

Kendati aplikasi ini sudah berfungsi secara optimal, temuan dari evaluasi kinerja membuka lebar pintu untuk berbagai pengembangan di masa mendatang. Sebagai langkah penyempurnaan, model *machine learning* ke depannya bisa ditingkatkan dengan memperbanyak variasi data latih LSTM seperti memasड़की skenario pencahayaan yang berbeda, latar belakang yang lebih dinamis, hingga kemampuan mendeteksi dua tangan sekaligus (*bimanual*) agar mampu menampung kosa kata standar bahasa isyarat di Indonesia (seperti BISINDO atau SIBI) secara utuh. Selain itu, riset lanjutan juga bisa diarahkan untuk memperkaya basis korpus N-Gram dengan cakupan kalimat sehari-hari yang lebih luas agar susunan prediksinya semakin luwes. Ke depannya, teknologi ini juga sangat potensial untuk dikembangkan menjadi sistem komunikasi dua arah (dari suara menjadi isyarat visual) dalam bentuk *Progressive Web App* (PWA), sehingga bisa segera diaplikasikan secara luas di berbagai fasilitas umum seperti rumah sakit, instansi pelayanan publik, dan sekolah-sekolah di Indonesia.

UCAPAN TERIMA KASIH

Penulis mengucapkan puji dan syukur ke hadirat Tuhan Yang Maha Esa atas limpahan rahmat serta karunia-Nya, sehingga rangkaian penelitian ini bisa rampung dengan baik. Rasa terima kasih yang tulus juga disampaikan kepada Program Studi Teknik Informatika khususnya Bapak Ardi Wijaya, S.Kom., M.Kom. atas dukungan fasilitas serta terciptanya atmosfer akademik yang sangat mendukung selama proses penyusunan karya ini. Apresiasi yang setinggi-tingginya turut penulis haturkan kepada Ibu Muntahanah, S.Kom., M.Kom. selaku dosen pembimbing, yang dengan penuh kesabaran telah meluangkan waktu untuk membimbing, mengarahkan, dan memberikan masukan kritis dari awal perancangan hingga naskah ini tuntas. Tidak lupa, ucapan terima kasih ditujukan kepada keluarga tercinta, rekan-rekan seperjuangan, serta semua pihak yang tidak bisa disebutkan satu per satu, yang senantiasa mengalirkan dukungan moral maupun material hingga sistem penerjemah bahasa isyarat dan penulisan ilmiah ini akhirnya dapat terwujud.

REFERENSI

[1] Badan Pusat Statistik. (2023). Statistik penduduk penyandang disabilitas Indonesia 2023. BPS Indonesia.
[2] Cholifah, W. N., Yulianingsih, & Sagita, S. M. (2018). Pengujian black box testing pada aplikasi Action & Strategy berbasis Android dengan teknologi PhoneGap. Jurnal String, 3(2), 206–210.
[3] Google. (2023a). HandPose model for TensorFlow.js. Diperoleh dari https://github.com/tensorflow/tfjs-models/tree/master/handpose.
[4] Google. (2023b). TensorFlow.js: Machine learning for the web and beyond. Diperoleh dari https://www.tensorflow.org/js
[5] Handayani, A. N., Hariyono, H., Nasih, A. M., Rochmawati, R., Hitipeuw, I., Ar Rosyid, H., Ardiansah, J. T., Praja, R. I., Nurdiansyah, A., & Azizah, D. F. (2025). Comparative analysis of speech-to-text APIs for supporting communication of the deaf community. Indonesian Journal of Data and Science, 6(3), 492–500.
[6] Jurafsky, D., & Martin, J. H. (2021). Speech and Language Processing (3rd ed.). Draft version. Stanford University.
[7] Khan, R., Ahmad, S., & Ali, M. (2022). Performance analysis of N-gram language models for text prediction. International Journal of Advanced Computer Science and Applications, 13(5), 45–52.
[8] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444.
[9] Norré, L. (2020). Word prediction using N-gram models in augmentative and alternative communication systems. Assistive Technology Journal, 32(4), 210–218.
[10] Mahardika, M. A. R., Yudistira, N., & Ridok, A. (2024). Sistem rekognisi citra digital bahasa isyarat menggunakan convolutional neural network dan spatial transformer. Jurnal Teknologi Informasi dan Ilmu Komputer, 11(6), 1159–1168. https://doi.org/10.25126/jtiik.2023118098
[11] Moryossef, A., Tsochantaridis, I., Aharoni, R., Ebling, S., & Narayanan, S. (2020). Real-time sign language detection using human pose estimation. arXiv preprint arXiv:2008.04637.
[12] Mozilla. (2026a). Canvas API. Diperoleh dari https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API.
[13]Mozilla. (2026b). WebRTC API. Diperoleh dari https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API.
[14]Myers, G. J., Sandler, C., & Badgett, T. (2022). The art of software testing (4th ed.). John Wiley & Sons.
[15]Norré, M. (2020). Evaluation of a word prediction system in an augmentative and alternative communication for disabled people. Mathematical Modelling of Engineering Problems, 8(1–4), 49–54.
[16]Oyedotun, O. K., & Khashman, A. (2017). Deep learning in vision-based static hand gesture recognition. Neural Computing and Applications, 28(12), 3941–3951.
[17]Paul, S. K., Walid, M. A. A., Paul, R. R., Uddin, M. J., Rana, M. S., Devnath, M. K., Dipu, I. R., & Haque, M. M. (2024). An Adam based CNN and LSTM approach for sign language recognition in real time for deaf people. Bulletin of Electrical Engineering and Informatics, 13(1), 499–509.
[18]Pressman, R. S., & Maxim, B. R. (2021). Software engineering: A practitioner's approach (9th ed.). McGraw-Hill Education.
[19]Reddy, S., Kumar, P., & Sharma, V. (2022). Time series prediction using LSTM networks. Journal of Artificial Intelligence Research, 68, 345–360.
[20]Rodríguez-Correa, P. A., Valencia-Arias, A., Patiño-Toro, O. N., Oblitas Díaz, Y., & Teodori De la Puente, R. (2023). Benefits and development of assistive technologies for Deaf people's communication: A systematic review. Frontiers in Education, 8, 1121597.
[21]Smilkov, D., Thorat, N., Assogba, Y., Yuan, A., Kreeger, N., Yu, P., Zhang, K., Cai, S., Nielsen, E., Soergel, D., Bileschi, S., Terry, M., Nicholson, C., Gupta, S. N., Sirajuddin, S., Sculley, D., Monga, R., Corrado, G., Viégas, F. B., & Wattenberg, M. (2019). TensorFlow.js: Machine learning for the web and beyond. arXiv preprint arXiv:1901.05350.
[22]Sommerville, I. (2022). Software engineering (10th ed.). Pearson Education.
[23]W3C. (2018). Web content accessibility guidelines (WCAG) 2.1. Diperoleh dari https://www.w3.org/TR/WCAG21/.

[24]W3C. (2022). WebRTC 1.0: Real-time communication between browsers. Diperoleh dari https://www.w3.org/TR/webrtc/.
[25]W3C. (2023). Web Speech API specification. Diperoleh dari https://wicg.github.io/speech-api/.
[26]Yadav, A., Patel, K., & Mehta, R. (2021). Applications of LSTM in natural language processing: A review. International Journal of Machine Learning and Computing, 11(3), 250–256.
[27]Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C. L., & Grundmann, M. (2020). MediaPipe Hands: On-device real-time hand tracking. arXiv preprint arXiv:2006.10214..