Journal of Applied Informatics and Computing (JAIC)
Vol.10, No.2, April 2026, pp. 1631~1638
e-ISSN: 2548-6861

Indonesian Sign Language Application Using Mediapipe and Gated Recurrent Unit in Real Time

Adi Triswantoro [1]*, Febrian Murti Dewanto [2]*, Aris Tri Joko Harjanto [3]*
* Informatika, Universitas Persatuan Guru Republik Indonesia Semarang
adi@upgris.ac.id [1], febrianmd@upgris.ac.id [2], aristrijaka@upgris.ac.id [3]

Article Info

Article history:

Received 2026-01-02
Revised 2026-02-15
Accepted 2026-02-27

Keyword:

BISINDO,
Deep Learning,
Gated Recurrent Unit,
MediaPipe,
Real-time Recognition.

ABSTRACT

Indonesian Sign Language (BISINDO) is a natural communication tool for the deaf community. However, the communication gap between signers and the general public remains a challenge due to the dynamic nature of sign language. This study proposes a real-time recognition system using the Gated Recurrent Unit (GRU) method. The system utilizes MediaPipe Holistic to extract 1,662 spatial keypoints, which are then processed as temporal sequences of 60 frames. The dataset comprises 300 video samples of ten dynamic BISINDO gestures ('apa', 'bapak', 'dimana', 'halo', 'hari', 'ibu', 'kabar', 'siapa', ‘kenapa’, ‘rumah’) recorded from multi dependent user under consistent indoor lighting conditions. The proposed model architecture consists of two GRU layers with Batch Normalization and Dropout to optimize performance with a total of 415,946 parameters. Results show that the model successfully achieves 86,67% accuracy and efficient inference speeds, making it suitable for real-time application on standard computing devices. This research serves as a proof-of-concept for assistive communication technology.

I. PENDAHULUAN

Komunikasi didefinisikan sebagai proses pertukaran informasi esensial yang memfasilitasi hubungan antara manusia dengan lingkungannya. Melalui penyampaian pesan yang sistematis, baik secara individu maupun organisasional, proses ini memastikan terjadinya konektivitas sosial yang dibutuhkan dalam kehidupan bermasyarakat[1].

Merujuk pada data Sensus Penduduk Long Form 2020 dari Badan Pusat Statistik (BPS), tercatat sekitar 1,43% populasi Indonesia merupakan penyandang disabilitas, di mana 0,36% di antaranya memiliki kendala pendengaran dan 0,35% memiliki kendala bicara. Statistik tersebut mengonfirmasi tantangan komunikasi nyata yang dihadapi oleh komunitas tunarungu dan tunawicara dalam aktivitas keseharian mereka. Secara global, *World Health Organization (WHO)* melaporkan bahwa lebih dari 1,5 miliar orang menderita gangguan pendengaran, dengan tren yang terus meningkat seiring bertambahnya usia harapan hidup populasi. Hambatan dalam berkomunikasi ini tidak hanya berdampak pada terbatasnya akses pendidikan dan peluang profesional, tetapi juga menjadi penghalang bagi partisipasi sosial dan integrasi penuh di masyarakat[2].

Sebagai negara dengan kerentanan terhadap berbagai jenis disabilitas, masalah tunarungu di Indonesia memerlukan perhatian serius karena konsekuensi sosial dan kesehatan yang ditimbulkannya. Gangguan pada indra pendengaran, baik dalam skala parsial maupun menyeluruh, mengakibatkan penyandangnya kesulitan dalam melakukan komunikasi verbal secara konvensional. Hal ini tidak jarang menyebabkan munculnya kesenjangan status sosial bagi penyandang tunarungu di tengah masyarakat. Dalam menjalankan aktivitas sehari-hari, komunitas ini menggunakan bahasa isyarat sebagai media interaksi yang krusial. Bahasa isyarat sendiri didefinisikan sebagai metode komunikasi yang mengutamakan penggunaan gerakan tangan, bahasa tubuh, dan artikulasi bibir sebagai pengganti komunikasi verbal lisan[3].

Ada dua penggunaan bahasa isyarat di Indonesia, yaitu Sistem Isyarat Bahasa Indonesia (SIBI) dan Bahasa Isyarat Indonesia (BISINDO). SIBI, yang strukturnya banyak mengadopsi *American Sign Language* (ASL), telah

ditetapkan sebagai standar resmi oleh pemerintah untuk diimplementasikan di lingkungan pendidikan formal.

Komunitas tuli di Indonesia lebih dominan menggunakan BISINDO dalam interaksi sosial sehari-hari karena dianggap sebagai bahasa isyarat yang tumbuh secara alami sesuai budaya lokal [4].

Dalam penelitian terdahulu oleh Borman dan Priyopradono menunjukkan bahwa penggunaan *pixel based* pada citra statis memiliki keterbatasan dalam mengenali isyarat dinamis (seperti huruf J dan R) serta sering mengalami kesalahan klasifikasi pada bentuk tangan yang memiliki kemiripan visual[5]. Beberapa huruf (seperti B, D, P, M, dan N) sulit dikenali karena bentuk tangan yang hampir mirip satu sama lain. Hal ini mengindikasikan adanya kebutuhan akan model yang mampu menangkap dependensi temporal dan fitur koordinat yang lebih spesifik untuk membedakan kata-kata dengan posisi awal serupa namun arah gerakan berbeda. Jika hanya mengandalkan fitur statis seperti dalam artikel tersebut, kata-kata dengan posisi tangan awal yang serupa akan sulit dibedakan.

Penelitian oleh Saputra dalam pengenalan pola visual bahasa isyarat melalui metode *Convolutional Neural Network* (CNN) berbasis citra [6]. Terdapat batasan fundamental yang menjadi fokus utama dalam penelitian ini yaitu sifat data (Statis vs. Dinamis). Metode CNN dalam penelitian terdahulu berfokus pada pengenalan pola huruf atau isyarat statis dari gambar tunggal. Hal ini tidak mencukupi untuk menangkap esensi bahasa isyarat BISINDO yang bersifat dinamis, di mana makna sebuah kata sering kali ditentukan oleh transisi gerakan tangan dari waktu ke waktu dan Skalabilitas Fitur, dengan hanya menggunakan 3 kelas isyarat ("Rumah", "Tenda", "Halo"), model CNN konvensional belum diuji untuk menangani kompleksitas fitur tubuh yang lebih luas seperti ekspresi wajah dan pose tubuh secara simultan. Penelitian yang dilakukan Feliciano metode *transfer learning* menggunakan model VGG16 dan Xception juga masih menggunakan dataset citra [7].

Masalah utama yang diidentifikasi dalam penelitian ini meliputi beberapa aspek penting. Pertama, kompleksitas data menjadi tantangan karena pengolahan citra video mentah berbasis piksel memerlukan daya komputasi yang sangat besar serta waktu pemrosesan yang relatif lama. Kedua, terdapat dependensi temporal, di mana gerakan isyarat tidak hanya berupa gambar statis, melainkan rangkaian gerakan yang maknanya ditentukan oleh perubahan posisi dari waktu ke waktu. Ketiga, akurasi klasifikasi juga menjadi kendala, terutama dalam membedakan kata-kata yang memiliki posisi awal yang serupa tetapi arah gerakannya berbeda.

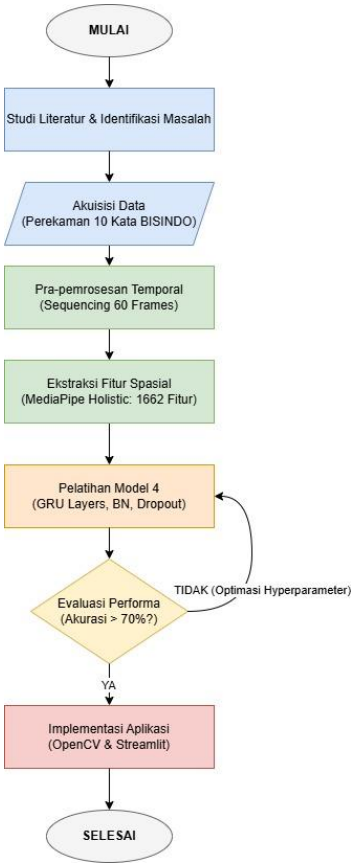
Tujuan utama dari penelitian ini adalah merancang arsitektur model berbasis GRU yang dioptimasi untuk mengenali 10 kosakata dasar BISINDO, yaitu “apa”, “bapak”, “dimana”, “halo”, “hari”, “ibu”, “kabar”, “siapa”, “kenapa”, dan “rumah”. Selain itu, penelitian ini juga bertujuan untuk mencapai tingkat akurasi di atas 70% agar sistem yang dikembangkan layak untuk diimplementasikan secara praktis. Di samping itu, penelitian ini mengembangkan aplikasi antarmuka menggunakan OpenCV dan Streamlit yang mampu melakukan penerjemahan bahasa isyarat secara real-time dengan latensi yang rendah..

II. METODE PENELITIAN

Penelitian ini menggunakan metode Eksperimental dan *Research and Development* berbasis *Deep Learning*, khususnya menggunakan arsitektur *Gated Recurrent Unit* (GRU) untuk pengolahan data sekuensial (urutan waktu). Secara teknis, penelitian ini menerapkan pendekatan *Computer Vision* untuk menerjemahkan gerakan dinamis Bahasa Isyarat Indonesia (BISINDO) menjadi teks secara *real-time*.

A. Tahapan Penelitian

Tahapan prosedur penelitian secara teknis diuraikan pada gambar 2. Tahap awal melibatkan pencarian referensi terkait pengenalan bahasa isyarat menggunakan *deep learning* serta pendefinisian hambatan dalam komunikasi Bahasa Isyarat Indonesia (BISINDO). Data dikumpulkan melalui proses perekaman video untuk 10 kosakata BISINDO yang berbeda. Video direkam dalam kondisi yang terkontrol untuk memastikan kualitas dataset.



Gambar 1 Alur Penelitian

Data koordinat kemudian diproses ke dalam format sekuensial dengan durasi 60 *frames* pada kecepatan 30 *fps* per sekuensi gerakan. Hal ini bertujuan agar model dapat mempelajari informasi temporal dari setiap isyarat. Menggunakan pustaka *MediaPipe Holistic*, sistem mengekstraksi titik koordinat tubuh untuk menghasilkan total 1662 fitur spasial per *frame*. Fitur ini mencakup koordinat tangan, pose tubuh, dan wajah. Pengembangan model menggunakan arsitektur *Gated Recurrent Unit* (GRU) yang dilengkapi dengan lapisan *Batch Normalization* (BN) untuk stabilitas pelatihan dan *Dropout* untuk mencegah terjadinya *overfitting*. Model dievaluasi berdasarkan metrik akurasi. Jika akurasi model lebih rendah dari ambang batas 70%, maka dilakukan optimasi *hyperparameter* pada tahap pelatihan ulang. Jika akurasi terpenuhi, penelitian berlanjut ke tahap implementasi. Model akhir diintegrasikan ke dalam antarmuka aplikasi menggunakan *library* OpenCV untuk pemrosesan video *real-time* dan *framework* Streamlit sebagai antarmuka pengguna.

III. HASIL DAN PEMBAHASAN

Hasil yang didapatkan dari penelitian ini akan dijelaskan berdasarkan alur penelitian seperti berikut :

A. Akuisisi Data

Proses akuisisi data dilakukan oleh *multi signer dependent* yang merekam sejumlah 10 kata yaitu 'apa', 'bapak', 'dimana', 'halo', 'hari', 'ibu', 'kabar', 'siapa', ‘kenapa’, ‘rumah’.

Tiap kata masing-masing terdiri dari 30 video repetisi dengan variasi *signer* lima (5) kali menghadap kamera *frontal*, lima (5) kali miring 45^0 ke kiri dan lima (5) kali miring 45^0 ke kanan. Pencahayaan indoor dengan penerangan lampu ruangan. Disimpan dalam bentuk video dengan format mp4. Direkam menggunakan webcam merk Nemesis dengan resolusi 1920x1080, *frametrate* 15,24 fps. Durasi masih bervariatif antara 2-3 detik.

B. Pra-pemrosesan

Data koordinat kemudian diproses ke dalam format sekuensial dengan durasi 60 *frames* pada kecepatan 30 *fps* per sekuensi gerakan untuk normalisasi data sebelum masuk ekstraksi fitur spasial dan pelatihan.

Tahap pra-pemrosesan merupakan langkah krusial untuk memastikan bahwa data video yang digunakan dalam pelatihan model memiliki standar yang seragam. Mengingat input untuk model GRU sangat bergantung pada dimensi waktu (sekuensial), maka dilakukan proses normalisasi temporal.

Tahap pemrosesan dalam penelitian ini diawali dengan penetapan parameter standar untuk menghasilkan output video yang seragam. Seluruh video mentah dikonfigurasi memiliki kecepatan 30 frame per detik (FPS) dengan durasi tetap selama 2 detik, sehingga setiap sekuensi video terdiri dari 60 frame sebagai dimensi input yang konsisten bagi model. Selanjutnya, dilakukan proses ekstraksi dan interpolasi frame karena setiap video asli memiliki durasi dan jumlah frame yang berbeda. Teknik sampling menggunakan fungsi linier (np.linspace) diterapkan untuk memilih secara proporsional 60 frame dari awal hingga akhir video, sehingga esensi gerakan isyarat tetap terjaga baik pada video yang terlalu cepat maupun terlalu lambat.

Setelah frame terpilih diperoleh, dilakukan proses rekonstruksi video dengan menuliskan kembali frame-frame tersebut ke dalam format video baru (.mp4) menggunakan codec mp4v. Proses ini mempertahankan resolusi asli (lebar dan tinggi) video, namun dengan frame rate yang telah distandarkan pada 30 FPS. Seluruh tahapan ini dijalankan secara otomatis melalui pemrosesan batch berbasis kelas, di mana sistem melakukan iterasi pada setiap folder kosakata BISINDO seperti “apa”, “halo”, dan “bapak”, serta memproses seluruh file video di dalamnya secara berurutan.

Hasil akhir dari proses ini adalah dataset baru yang terorganisasi dalam direktori Dataset_NormalisasiMP2, yang berisi video-video dengan durasi dan frame rate yang seragam. Dataset ini dirancang untuk meminimalkan variansi temporal sehingga dapat meningkatkan konsistensi dan performa model dalam mengenali pola gerakan bahasa isyarat.

C. Ekstraksi Fitur Spasial

Setelah video dinormalisasi, tahap selanjutnya adalah mentransformasi data visual mentah menjadi data numerik berupa koordinat titik kunci (*keypoints*).

Proses ekstraksi fitur dalam penelitian ini diawali dengan konfigurasi model Holistic menggunakan MediaPipe, di mana sistem diinisialisasi dengan ambang batas kepercayaan deteksi dan pelacakan sebesar 0,5. Setiap frame video diproses dengan mengonversi ruang warna dari BGR ke RGB agar sesuai dengan kebutuhan pemrosesan internal model.

Selanjutnya, dilakukan segmentasi dan ekstraksi fitur secara multi-modal untuk menangkap gerakan isyarat secara komprehensif. Sistem mengekstraksi empat komponen utama, yaitu landmark wajah, pose tubuh, serta tangan kiri dan kanan. Landmark wajah menghasilkan 468 titik kunci tiga dimensi (x, y, z) dengan total 1.404 fitur yang berperan penting dalam menangkap ekspresi non-manual. Landmark pose mengekstraksi 33 titik kunci dari bahu hingga pinggang, masing-masing dengan atribut x, y, z, dan visibilitas sehingga menghasilkan 132 fitur. Sementara itu, landmark tangan kiri dan kanan masing-masing terdiri dari 21 titik kunci dengan koordinat x, y, z, yang secara keseluruhan memberikan tambahan 126 fitur. Dengan demikian, total fitur yang dihasilkan pada setiap frame mencapai 1.662 fitur.

Seluruh koordinat yang diperoleh kemudian diratakan menjadi satu vektor melalui fungsi extract_keypoints. Untuk menjaga konsistensi dimensi data, sistem menerapkan teknik zero padding menggunakan np.zeros apabila terdapat bagian tubuh yang tidak terdeteksi, sehingga panjang vektor tetap 1.662 nilai pada setiap frame. Selain itu, untuk keperluan evaluasi, sistem juga menyediakan visualisasi secara real-

time menggunakan fungsi draw_styled_landmarks, yang menampilkan titik koordinat dan garis koneksi pada video sehingga peneliti dapat memantau kualitas deteksi selama proses berlangsung.

Vektor fitur dari setiap frame kemudian disusun secara berurutan menjadi sebuah matriks NumPy, sehingga satu video direpresentasikan dalam bentuk data berdimensi (60, 1.662). Struktur ini mengandung informasi spasial dan temporal yang padat, dan selanjutnya digunakan sebagai input utama bagi model Gated Recurrent Unit (GRU). Sebagai hasil akhir, sistem tidak hanya menyimpan data numerik dalam format .npy untuk keperluan pelatihan, tetapi juga menghasilkan video visualisasi dalam format .mp4 guna mendukung proses verifikasi dan audit kualitas data sebelum tahap pemodelan dilakukan.

Gambar 2 Visualisasi Multi Signer Datase Apa 01 Mediapipe Holistic

Gambar 3 ini menunjukkan implementasi *landmark* di tahap Ekstraksi Fitur Spasial menggunakan *MediaPipe Holistic* pada salah satu video dataset kata “apa”.

D. Split Train dan *Test*

Proses pengembangan model diawali dengan pembagian dataset menggunakan teknik *stratified random sampling* dengan rasio 80:20, yang bertujuan untuk menjaga keseimbangan proporsi label pada data latih dan data uji demi validitas evaluasi. Data Latih: (240, 60, 1662), Data Uji: (60, 60, 1662)

Kode : *X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y).*

E. Pelatihan Model

Penyusunan Model dilakukan dengan mengintegrasikan lapisan *Gated Recurrent Unit* (GRU) secara bertingkat untuk menangkap pola temporal yang kompleks dari 1.662 fitur koordinat MediaPipe. Berikut adalah rincian tahapan perancangan dan pelatihan model:

1) Konstruksi Arsitektur Sequential, model dibangun menggunakan struktur Sequential dengan lapisan input yang menerima matriks berdimensi (60, 1662). Dimensi ini merepresentasikan sekuens 60 frame video yang masing-masing memiliki 1.662 fitur titik kunci hasil ekstraksi MediaPipe.

2) Ekstraksi Fitur Temporal dengan Stacked GRU. Inti dari model ini terletak pada dua lapisan Gated Recurrent Unit (GRU). Lapisan Pertama (gru_1): Berfungsi sebagai penyaring fitur awal yang mempertahankan dimensi temporal. Dengan output shape (None, 60, 64), lapisan ini memproses 60 time-steps dan menghasilkan 64 fitur tersembunyi untuk setiap langkahnya. Lapisan ini memiliki parameter terbesar (331.776), menunjukkan kapasitasnya yang besar dalam mempelajari pola rumit dari data mentah. Lapisan Kedua (gru_2), lapisan ini mengonsolidasikan informasi temporal menjadi satu vektor representasi tunggal berukuran 128. Ini adalah titik di mana model "merangkum" seluruh urutan waktu menjadi pemahaman konteks global.

3) Stabilisasi dan Regularisasi. Untuk mencegah masalah klasik seperti overfitting dan gradient vanishing, model ini menyisipkan beberapa lapisan pendukung Batch Normalization: Digunakan setelah setiap lapisan GRU untuk menormalisasi aktivasi, yang mempercepat proses pelatihan dan membuat model lebih kebal terhadap variasi skala input. Dropout, diterapkan secara strategis (dengan 0 parameter karena fungsinya hanya memutus koneksi secara acak saat training) untuk memastikan model tidak terlalu bergantung pada neuron tertentu, sehingga meningkatkan generalisasi pada data baru.

4) Klasifikasi Akhir melalui Fully Connected Layers. Setelah fitur temporal berhasil diekstraksi, data dialirkan ke lapisan Dense menjadi dense_1 sebagai lapisan perantara dengan 64 neuron untuk memetakan fitur GRU ke ruang keputusan yang lebih kecil. dense_2 (Output) merupakan lapisan terakhir dengan 10 unit. Tergantung pada fungsinya, ini menunjukkan bahwa model dirancang untuk tugas klasifikasi 10 kelas (misalnya, klasifikasi angka atau kategori aktivitas).

5) Optimasi dan Pelatihan. Model dikompilasi menggunakan optimizer Adam dan fungsi kerugian categorical crossentropy. Pelatihan dilakukan selama 300 epoch dengan ukuran batch sebanyak 10. Untuk menjamin efektivitas, sistem menerapkan mekanisme ModelCheckpoint yang secara otomatis menyimpan bobot model terbaik berdasarkan pencapaian validation categorical accuracy tertinggi selama proses iterasi berlangsung.

Berikut kode python tahap pelatihan model yang dilakukan di *jupyter notebook*:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import GRU, Dense,
Dropout, BatchNormalization

def build_model(input_shape, num_classes):
model_gru1 = Sequential([
        # Layer GRU 1
        GRU(64,              return_sequences=True,
activation='relu', input_shape=input_shape),
        BatchNormalization(),
        Dropout(0.3),
        # Layer GRU 2
        GRU(128,            return_sequences=False,
activation='relu'),
        BatchNormalization(),
        Dropout(0.3),
        # Hidden Layer
        Dense(64, activation='relu'),
        Dropout(0.3),
```

```
        # Output Layer
        Dense(num_classes, activation='softmax')
    ])

 model_gru1.compile(optimizer='Adam',
loss='categorical_crossentropy',
metrics=['categorical_accuracy'])
    return model_gru1

# Inisialisasi model
model_gru1 = build_model((60, 1662),
actions.shape[0])
model_gru1.summary()
# Callback agar tidak overfitting
early_stop = EarlyStopping(monitor='val_loss',
patience=40, restore_best_weights=True)
checkpoint =
ModelCheckpoint('model_gru1_terbaik.h5',
monitor='val_categorical_accuracy',
save_best_only=True)

# Proses Training
history = model_gru1.fit(
    X_train, y_train,
    epochs=300,
    batch_size=8,
    validation_data=(X_test, y_test),
    callbacks=[early_stop, checkpoint]
)
```

TABEL 1
RINCIAN PARAMETER MODEL

No	Nama Layer	Tipe Layer	Output Shape	Jumlah Parameter
1	**gru_1**	GRU	(None, 60, 64)	331,776
2	**batch_norm_1**	BatchNormalization	(None, 60, 64)	256
3	**dropout_1**	Dropout	(None, 60, 64)	0
4	**gru_2**	GRU	(None, 128)	74,496
5	**batch_norm_2**	BatchNormalization	(None, 128)	512
6	**dropout_2**	Dropout	(None, 128)	0
7	**dense_1**	Dense (FC)	(None, 64)	8,256
8	**dropout_3**	Dropout	(None, 64)	0
9	**dense_2**	Dense (Output)	(None, 10)	650

Tabel 1 tersebut merangkum rincian parameter untuk konfigurasi arsitektur model yang dirancang secara spesifik untuk menangani kompleksitas fitur spasial dari MediaPipe dan dependensi temporal dari gerakan BISINDO, dengan struktur sebagai berikut :

1) *gru*_1 (GRU) Pintu Masuk Data. Layer ini menerima urutan data (misalnya 60 langkah waktu). Dengan 331.776 parameter, ini adalah "otak" utama yang mempelajari pola hubungan antar waktu. *Output* (*None*, 60, 64) artinya ia masih mempertahankan urutan lengkap untuk diproses layer berikutnya.

2) *batch_normalization*_1 Penstabil Data. Berfungsi menormalkan hasil dari GRU pertama agar nilai datanya tidak terlalu besar atau kecil (tetap stabil). Ini membantu model belajar lebih cepat dan mencegah gangguan distribusi data selama pelatihan.

3) *dropout*_1 Pencegah Hafalan. Layer ini secara acak "mematikan" beberapa neuron selama latihan. Tujuannya agar model tidak hanya menghafal data (*overfitting*), tetapi benar-benar belajar polanya. Parameternya 0 karena tidak ada bobot yang disimpan di sini.

4) *gru*_2 (GRU) Penyimpul Informasi. Berbeda dengan GRU pertama, layer ini mengubah urutan waktu yang panjang tadi menjadi satu vektor tunggal berukuran 128. Output (None, 128) menandakan bahwa data tidak lagi berbentuk urutan, melainkan sudah menjadi ringkasan fitur yang padat.

5) *batch_normalization_2* Penstabil Akhir. Sama seperti sebelumnya, layer ini memastikan ringkasan fitur dari GRU kedua tetap berada dalam skala yang optimal sebelum masuk ke bagian pengambilan keputusan (klasifikasi).

6) *dropout_2* Regulasi Tambahan. Memberikan lapisan keamanan kedua agar model tetap fleksibel dan memiliki generalisasi yang baik pada data baru yang belum pernah dilihat sebelumnya.

7) *dense_1(Dense)* Pemroses Fitur. Ini adalah lapisan saraf standar (*Fully Connected*). Ia mengambil 128 fitur dari GRU dan memetakannya menjadi 64 fitur yang lebih spesifik untuk membantu proses klasifikasi akhir.

8) *dropout_3 Final Check*. *Dropout* terakhir sebelum hasil akhir dikeluarkan, memastikan keputusan yang diambil oleh model tidak hanya bergantung pada satu jalur neuron saja.

9) *dense_2* (Dense) Lapisan Keputusan (*Output*). Layer terakhir dengan 10 unit. *Output* (*None*, 10) menunjukkan hasil akhir model, yaitu probabilitas untuk 10 kategori berbeda.

10) *Total Parameter*: Model memiliki 415,946 parameter yang dapat dilatih. Jumlah ini relatif kecil dibandingkan model lainnya, sehingga sangat efisien untuk proses *real-time*.

F. Evaluasi Performa

Hasil pengujian pada Epoch 100, Model menunjukkan peningkatan signifikan dalam mengenali gerakan dinamis, mencapai akurasi validasi sebesar 86,67%. Hasil ini membuktikan bahwa kombinasi fitur MediaPipe yang padat dengan arsitektur GRU bertingkat efektif dalam mengatasi hambatan pengenalan isyarat yang memiliki kemiripan posisi awal namun berbeda arah gerakan.

30/30 1s *30ms/step - categorical_accuracy: 0.9542 - loss: 0.1480 - val_categorical_accuracy: 0.8667 - val_loss: 0.4263*
Epoch 100/300

TABEL 2
HASIL KLASIFIKASI MODEL

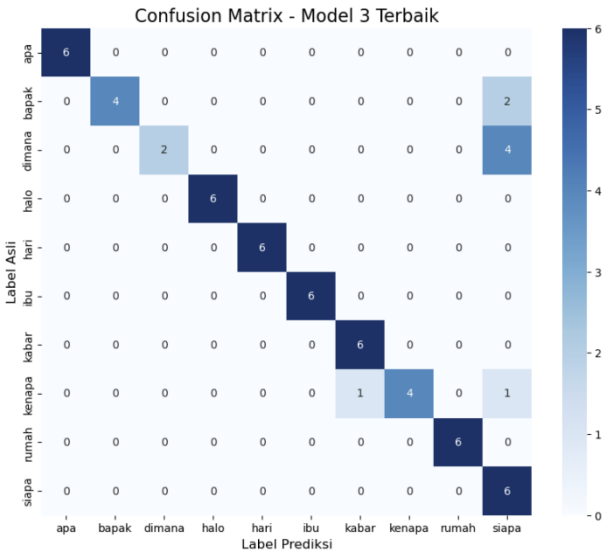
Kata (Label)	Precision	Recall	F1-Score	Support
apa	1.00	1.00	1.00	6
bapak	1.00	0.67	0.80	6
dimana	1.00	0.33	0.50	6
halo	1.00	1.00	1.00	6
hari	1.00	1.00	1.00	6
ibu	1.00	1.00	1.00	6
kabar	0.86	1.00	0.92	6
kenapa	1.00	0.67	0.80	6
rumah	1.00	1.00	1.00	6
siapa	0.46	1.00	0.63	6
Rata-rata (Macro)	0.93	0.87	0.87	60

Tabel 2 hasil klasifikasi model tersebut merupakan Laporan Klasifikasi (*Classification Report*) yang mengevaluasi kinerja Model (GRU) dalam mengenali 10 kelas bahasa isyarat BISINDO. Tabel ini memberikan gambaran detail mengenai kelas mana yang berhasil dikenali dengan baik dan kelas mana yang menjadi titik lemah model. Berikut adalah penjelasan mendalam mengenai parameter dan hasil pada tabel tersebut:

- Kelompok Performa Sempurna (apa, halo, hari, ibu, rumah): Model menunjukkan pengenalan luar biasa pada kata-kata ini dengan skor 1.00 (100%) di semua metrik. Ini berarti model tidak pernah salah menebak (*Precision*) dan tidak pernah melewatkan satu pun sampel (*Recall*) dari kata-kata tersebut.
- Kabar (Skor F1: 0.92): Model mampu menangkap semua contoh kata "kabar" (*Recall* 1.00), namun ada sedikit kesalahan di mana kata lain salah dikira sebagai "kabar" (*Precision* 0.86).
- Bapak dan kenapa (Skor F1: 0.80): Kedua kata ini memiliki masalah yang sama. Meskipun model sangat yakin saat menebak (*Precision* 1.00), model gagal mendeteksi sekitar 33% dari total kemunculan sebenarnya (*Recall* 0.67). Artinya, ada beberapa sampel "bapak" dan "kenapa" yang tidak terdeteksi.
- Siapa (Skor F1: 0.63): Kata ini adalah penyebab utama penurunan presisi global. Model berhasil menangkap semua kata "siapa" (*Recall* 1.00), tetapi model juga sangat sering salah mengklasifikasikan kata lain sebagai "siapa" (*Precision* 0.46). Ini menunjukkan adanya ambiguitas fitur pada kata ini.
- Dimana (Skor F1: 0.50): Ini adalah titik terlemah model. Meski prediksi yang dihasilkan akurat (*Precision* 1.00), model hanya mampu mengenali 33% dari total sampel yang ada. Sebagian besar kata "dimana" luput dari deteksi model (terklasifikasi ke label lain).

Gambar 4 menyajikan *Confusion Matrix* dari performa Model. Berdasarkan data matriks, kita dapat mengidentifikasi di mana model bekerja sempurna dan di mana model mengalami kebingungan (*confusion*).

Klasifikasi Sempurna (Skor 6/6): Kata "apa", "halo", "hari", "ibu", "kabar", "rumah", dan "siapa" berhasil diklasifikasikan dengan akurasi 100%. Tidak ada satu pun sampel dari kata-kata ini yang salah masuk ke kategori lain.

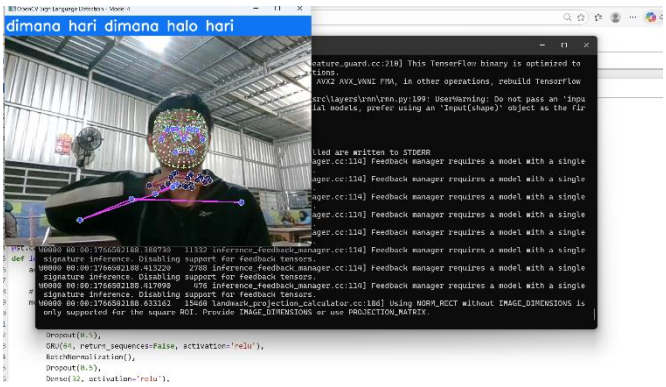


Gambar 3 Confusion Matrix

Klasifikasi dengan Kesalahan, yaitu kata bapak dari 6 data asli, model berhasil menebak 4 dengan benar, namun 2 data salah diprediksi sebagai "siapa". Kata dimana adalah titik terlemah model. Hanya 2 data yang berhasil ditebak dengan benar, sementara 4 data lainnya salah diprediksi sebagai "siapa". Kata kenapa, model menebak 4 data dengan benar, tetapi 1 data salah dikira sebagai "kabar" dan 1 data salah dikira sebagai "siapa".

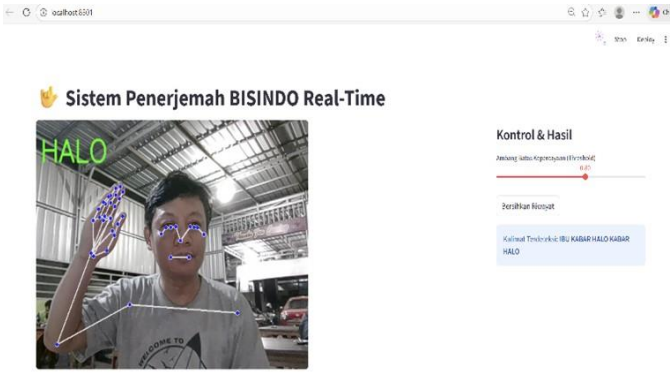
G. Implementasi Aplikasi

Tahap ini menjelaskan penggabungan antara perangkat lunak pengolah video (OpenCV), ekstraktor fitur (MediaPipe), dan otak pengenal (Model GRU). OpenCV bertugas sebagai bagian yang menangkap setiap *frame* dari kamera secara kontinu. Hasil pengujian menunjukkan bahwa sistem mampu melakukan klasifikasi secara responsif dengan menampilkan label teks hasil prediksi di layar secara seketika. Hal ini membuktikan bahwa integrasi MediaPipe dan GRU memiliki efisiensi yang cukup tinggi untuk digunakan sebagai aplikasi penerjemah isyarat praktis dengan latensi minimal.



Gambar 4 Opencv Realtime

Gambar 5 menunjukkan implementasi sistem dalam lingkungan *real-time* menggunakan antarmuka OpenCV. Model yang telah dilatih diintegrasikan untuk melakukan inferensi langsung terhadap masukan video dari kamera.



Gambar 5 Implementasi di Streamlit

Gambar 6 merupakan bukti keberhasilan implementasi antarmuka pengguna (UI) berbasis web menggunakan framework Streamlit. Tampilan ini menunjukkan bahwa model GRU tidak hanya berjalan di terminal atau OpenCV lokal, tetapi sudah siap dideploy sebagai aplikasi interaktif.

Berikut adalah penjelasan detail mengenai komponen-komponen yang tampak pada gambar tersebut:

- Akurasi Prediksi ("HALO"): Teks hijau bertuliskan "HALO" di pojok kiri atas video menunjukkan bahwa model berhasil melakukan klasifikasi dengan benar. Hal ini membuktikan bahwa meskipun dijalankan di lingkungan browser (localhost:8501), model GRU tetap mampu menangkap pola gerakan secara akurat.
- Stabilitas *Landmark* MediaPipe: Titik-titik biru pada tangan dan wajah menunjukkan bahwa proses ekstraksi 1.662 fitur tetap berjalan stabil. Terlihat deteksi tangan yang sangat detail pada gerakan melambaikan tangan, yang merupakan kunci utama pembeda kata "HALO" dengan kata lainnya.
- Ambang Batas Kepercayaan (*Threshold* 0.80): Penggunaan *slider* ini sangat krusial. Angka 0.80 menunjukkan bahwa sistem hanya akan menampilkan hasil jika model yakin lebih dari 80%. Ini berfungsi sebagai filter untuk mengurangi "noise" atau kesalahan deteksi saat tangan sedang bergerak transisi.
- Interaktivitas ("Bersihkan Riwayat"): Adanya tombol ini memudahkan pengguna untuk menghapus log kalimat dan memulai percakapan baru tanpa harus memuat ulang (*restart*) aplikasi.
- Fitur *Sentence History*: Di bawah tulisan "Kalimat Terdeteksi", terlihat rangkaian kata "IBU KABAR HALO KABAR HALO". Ini adalah fitur unggulan dari skrip Streamlit. Sistem tidak hanya menebak satu kata lalu hilang, tetapi merekam urutan kata yang berhasil dideteksi untuk membentuk sebuah kalimat utuh.
- Kekuatan Temporal GRU: Munculnya rangkaian kata ini membuktikan bahwa mekanisme *buffer* 60 frame yang bekerja dengan baik dalam menangkap perpindahan antar gerakan isyarat.

IV. KESIMPULAN

Penelitian ini berhasil mengintegrasikan teknologi MediaPipe Holistic dan arsitektur Gated Recurrent Unit (GRU) untuk mengenali 10 kosakata BISINDO secara dinamis. Dengan mengekstraksi 1.662 titik koordinat sebagai fitur spasial dan memprosesnya melalui model sekuensial dengan total 415,946 parameter, sistem mampu mencapai akurasi keseluruhan sebesar 86,67%. Hasil evaluasi menunjukkan performa sempurna pada kata ""apa", "halo", "hari", "ibu", "kabar", "rumah", dan "siapa", serta efektivitas mekanisme *gate* pada GRU dalam membedakan isyarat dengan posisi awal serupa seperti "bapak" dan “kenapa” (*F1-Score* 0.80). Dari uji data performa sistem secara real time, yaitu rata-rata kecepatan proses sebesar 3.34 FPS, *Total Latency* 285.1 ms, *Inference Model* 160 ms.

Akurasi sebesar 86,67% pada pengujian *real-time* ini dianggap memadai sebagai tahap awal *(proof-of-concept)* mengingat model memproses data video dinamis yang memiliki kompleksitas jauh lebih tinggi dibandingkan klasifikasi gambar statis. Rendahnya nilai pada kelas tertentu dipicu oleh kemiripan lintasan koordinat antar gestur yang mengakibatkan terjadinya misklasifikasi

Pada tahap implementasi, sistem terbukti dapat berjalan secara responsif melalui antarmuka OpenCV dan platform berbasis web Streamlit dengan memanfaatkan mekanisme *buffer* 60 *frames*. Penggunaan Streamlit memberikan nilai tambah berupa fitur interaktif seperti pengaturan *threshold* (0.80) dan akumulasi kalimat hasil prediksi yang meningkatkan pengalaman pengguna dalam berkomunikasi secara *real-time*. Keberhasilan sistem dalam melakukan *inference* langsung di lingkungan peramban dengan visualisasi *landmark* yang stabil mengukuhkan bahwa kombinasi ekstraksi koordinat padat dan jaringan saraf rekuren merupakan solusi yang efisien secara komputasi dan layak dikembangkan lebih lanjut untuk aplikasi penerjemah bahasa isyarat yang inklusif.

Berdasarkan hasil penelitian dan evaluasi yang telah dilakukan, terdapat saran untuk pengembangan sistem di masa mendatang guna meningkatkan akurasi dan fungsionalitas yaitu Peningkatan Volume dan Variasi Dataset, disarankan untuk menambah jumlah sampel data latih secara signifikan. Penambahan variasi dari sisi sudut pengambilan gambar (*angle*), jarak subjek ke kamera, serta keberagaman subjek (orang yang berbeda) sangat krusial untuk memperkuat generalisasi model.

DAFTAR PUSTAKA

[1] L. Arisandi and B. Satya, "Sistem Klarifikasi Bahasa Isyarat Indonesia (Bisindo) Dengan Menggunakan Algoritma Convolutional Neural Network," *J. Sist. Cerdas*, vol. 5, no. 3, pp. 135–146, Dec. 2022, doi: 10.37396/jsc.v5i3.262.

[2] C. Artamma and M. Rahardi, "L2IC and MobileViT-XXS for BISINDO Alphabet Recognition," vol. 9, no. 6.

[3] M. Kharis, M. Andika, and H. A. D. Rani, "Aplikasi Augmented Reality Interaktif Tuna Rungu Berbasis Android pada Sekolah Luar Biasa ABCD Muhammadiyah Palu," 2024.

[4] F. M. Dewanto, A. T. J. Harjanta, N. Q. Nada, and B. Agus, "Pengenalan Gestur Bahasa Isyarat Indonesia dengan Mediapipe Keypoints," vol. 6, no. 2, 2024.

[5] R. I. Borman and B. Priyopradono, "Implementasi Penerjemah Bahasa Isyarat Pada Bahasa Isyarat Indonesia (BISINDO) Dengan Metode Principal Component Analysis (PCA)," *J. Inform. J. Pengemb. IT*, vol. 3, no. 1, pp. 103–108, Jan. 2018, doi: 10.30591/jpit.v3i1.631.

[6] R. A. Saputra, F. M. Setiawan, E. Ryansyah, and C. Rozikin, "Pendeteksi Bahasa Isyarat Menggunakan TensorFlow dengan Metode Convolutional Neural Network," *J. Inform. Dan Rekayasa KomputerJAKAKOM*, vol. 5, no. 2.

[7] M. Feliciano, W. Gunawan, and N. E. Jahja, "American Sign Language (Asl) Recognition Using The Cnn Method," . *Vol.*, vol. 7, no. 2.

[8] Y. Brianorman and R. Munir, "Perbandingan Pre-Trained CNN: Klasifikasi Pengenalan Bahasa Isyarat Huruf Hijaiyah," *J Sist. Info Bisnis*, vol. 13, no. 1, pp. 52–59, Jul. 2023, doi: 10.21456/vol13iss1pp52-59.

[9] E. Tikasni, E. Utami, and D. Ariatmanto, "Analisis Akurasi Object Detection Menggunakan Tensorflow Untuk Pengenalan Bahasa Isyarat Tangan Menggunakan Metode SSD," *J. FASILKOM*, vol. 14, no. 2, pp. 385–393, Aug. 2024, doi: 10.37859/jf.v14i2.7512.

[10] A. Widya Agata, W. S J Saputra, and C. Aji Putra, "Pengenalan Bahasa Isyarat Indonesia (BISINDO) Menggunakan Algoritma Sscale Invariant Feature Transform (SIFT) Dan Convolutional Neural Network (CNN)," *JATI J. Mhs. Tek. Inform.*, vol. 8, no. 1, pp. 1054–1061, Mar. 2024, doi: 10.36040/jati.v8i1.8917.

[11] F. A. Febrianti, M. A. Rahman, N. Alani, R. Nuriyanti, and I. P. Sari, "Pemanfaatan Website Interaktif Berbasis Bahasa Isyarat Indonesia (BISINDO) sebagai Media Pembelajaran Inklusif bagi Anak Sekolah Dasar," 2025.

[12] C. Lugaresi *et al.*, "MediaPipe: A Framework for Perceiving and Processing Reality".

[13] D. P. Kartaputra, H. Gunawan, and A. E. Lestari, "Deteksi Alfabet BISINDO Menggunakan Mediapipe Holistic Secara Real-Time," *J. Teknol. Inf. Dan Komun.*, vol. 12, no. 1, pp. 46–52, Jun. 2023, doi: 10.58761/jurtikstmikbandung.v12i1.4652.

[14] S. Nur Budiman, S. Lestanti, S. Marselius Evvandri, and R. Kartika Putri, "Pengenalan Gestur Gerakan Jari Untuk Mengontrol Volume Di Komputer Menggunakan Library OpenCV Dan Mediapipe," *Antivirus J. Ilm. Tek. Inform.*, vol. 16, no. 2, pp. 223–232, Nov. 2022, doi: 10.35457/antivirus.v16i2.2508.

[15] J. Bora, S. Dehingia, A. Boruah, A. A. Chetia, and D. Gogoi, "Real-time Assamese Sign Language Recognition using MediaPipe and Deep Learning," *Procedia Comput. Sci.*, vol. 218, pp. 1384–1393, 2023, doi: 10.1016/j.procs.2023.01.117.

[16] P. Kurnia Sari, G. Qorik Oktagalu Pratamasunu, and F. Nur Fajri, "Deteksi Tangan Otomatis Pada Video Percakapan Bahasa Isyarat Indonesia Menggunakan Metode Deep Gated Recurrent Unit (GRU)," *J. Komput. Terap.*, vol. 8, no. 1, pp. 186–193, Jun. 2022, doi: 10.35143/jkt.v8i1.4901.

[17] M. A. Syifa and D. R. S. Saputro, "Stance Detection Dengan Algoritme Gated Recurrent Unit (GRU)," 2023.

[18] I. Sulistiyowati, H. Maulana Ichsan, and I. Anshory, "Konveyor Penyortir Objek Dengan Deteksi Warna Menggunakan Kamera ESP-32 Berbasis OpenCV Python," *Pros. TAU SNARS-TEK Semin. Nas. Rekayasa Dan Teknol.*, vol. 4, no. 1, pp. 35–41, Aug. 2024, doi: 10.47970/snarstek.v2i1.711.

[19] R. D. Martinez Marenco *et al.*, *Automated segmentation of breast cancer in ultrasound images using self-learning deep learning models*. Ediciones Universidad Simón Bolívar, 2024. doi: 10.17081/r.book.2025.01.16118.

[20] I. P. Y. Agus Ariwanta, K. Y. Ernanda Aryanto, and I. G. A. Gunadi, "Suricata Accuracy Optimization Based On Live Analysis Using One-Class Support Vector Machine Method And Streamlit Framework," *J. Tek. Inform. Jutif*, vol. 5, no. 2, pp. 415–427, Apr. 2024, doi: 10.52436/1.jutif.2024.5.2.1822.