

# Deteksi Bahasa Isyarat Indonesia (SIBI) dengan MediaPipe dan *Feedforward Neural Network*

Azka Brilliant Firdaus<sup>1\*</sup>, Fattah Satrio Atmojo<sup>2</sup>, Naufal Choirul Ananda<sup>3</sup>, Vincensius Gilang Pramudito<sup>4</sup>,  
Ridwan Dwi Irawan<sup>5</sup>

<sup>1</sup>Teknik Informatika/Ilmu Komputer  
Universitas Duta Bangsa  
<sup>1\*</sup>210103126@mhs.udb.ac.id

<sup>2</sup>Teknik Informatika/Ilmu Komputer  
Universitas Duta Bangsa  
<sup>2</sup>210103131@mhs.udb.ac.id

<sup>3</sup>Teknik Informatika/Ilmu Komputer  
Universitas Duta Bangsa  
<sup>3</sup>210103139@mhs.udb.ac.id

<sup>4</sup>Teknik Informatika/Ilmu Komputer  
Universitas Duta Bangsa  
<sup>4</sup>210103144@mhs.udb.ac.id

<sup>5</sup>Teknik Informatika/Ilmu Komputer  
Universitas Duta Bangsa  
<sup>5</sup>ridwan\_dwiirawan@udb.ac.id

**Abstrak**— Penelitian ini bertujuan untuk mengembangkan sistem klasifikasi huruf alfabet dalam Sistem Isyarat Bahasa Indonesia (SIBI) secara real-time menggunakan teknologi pengolahan citra dan kecerdasan buatan. Sistem dibangun melalui pendekatan *machine learning pipeline* dengan lima tahap utama: pengumpulan data, ekstraksi fitur, pelatihan model, evaluasi model, dan implementasi real-time. Data diperoleh dari Kaggle dan mencakup 24 kelas huruf (kecuali J dan Z), masing-masing terdiri dari 220 gambar. Ekstraksi fitur dilakukan dengan MediaPipe yang menghasilkan 63 fitur numerik dari 21 titik landmark tangan. Model klasifikasi dilatih menggunakan algoritma Feedforward Neural Network (FFNN) dengan dua lapis tersembunyi, fungsi aktivasi ReLU, serta teknik *dropout* untuk menghindari overfitting. Setelah 100 epoch pelatihan, model mencapai akurasi validasi sebesar 51,23%. Evaluasi menunjukkan beberapa huruf memiliki akurasi tinggi, sementara huruf lain cenderung membingungkan sistem karena bentuk gestur yang mirip. Sistem berhasil diuji secara real-time menggunakan kamera laptop dengan hasil yang menjanjikan. Deteksi huruf A, B, dan D masing-masing memperoleh nilai akurasi 0.98, 0.89, dan 0.94. Hasil ini menunjukkan bahwa kombinasi MediaPipe dan FFNN memberikan hasil akurasi mendekati 1 untuk mengenali bahasa isyarat Indonesia dalam bentuk huruf secara langsung. Penelitian ini menjadi langkah awal menuju sistem penerjemahan SIBI yang lebih menyeluruh dan aplikatif, khususnya dalam mendukung komunikasi bagi penyandang disabilitas.

**Kata kunci**— Bahasa Isyarat, SIBI, MediaPipe, Neural Network, Deteksi Real-Time.

**Abstract**— This study aims to develop a real-time classification system for alphabet letters in Indonesian Sign System (SIBI) using image processing and artificial intelligence technologies. The system follows a machine learning pipeline consisting of five stages: data collection, feature extraction, model training, model evaluation, and real-time implementation. The dataset was obtained from Kaggle and includes 24 letter classes (excluding J and Z), each with 220 images. Feature extraction was performed using MediaPipe, generating 63 numerical features from 21 hand landmarks. The classification model was trained using the Feedforward Neural Network (FFNN) algorithm with two hidden layers, ReLU activation functions, and dropout techniques to prevent overfitting. After 100 training epochs, the model achieved a validation accuracy of 51.23%. Evaluation showed that certain letters had high accuracy, while others were misclassified due to similar hand gestures. The system was tested in real-time using a laptop camera, producing promising results. Detection of the letters A, B, and D achieved accuracy scores of 0.98, 0.89, and 0.94, respectively. These results demonstrate that the combination of MediaPipe and FFNN provides accuracy results close to 1 for recognizing Indonesian sign language letters in real-time. This research lays the groundwork for the development of a more comprehensive and applicable SIBI translation system, especially in supporting communication for individuals with hearing and speech impairments.

**Keywords**— Sign Language, SIBI, MediPipe, Neural Network, Real-Time Detection.

## I. PENDAHULUAN

Komunikasi merupakan aspek fundamental dalam kehidupan manusia yang mendukung aktivitas sosial sehari-hari [1]. Bagi individu dengan keterbatasan pendengaran dan kemampuan berbicara, komunikasi verbal menjadi hal yang sulit, sehingga mereka

mengandalkan bahasa isyarat sebagai sarana utama untuk menyampaikan pesan [2]. Di Indonesia, bentuk bahasa isyarat yang diakui secara resmi adalah Sistem Isyarat Bahasa Indonesia (SIBI), yaitu sistem yang dirancang berdasarkan tata bahasa Indonesia dengan mengombinasikan gestur tangan,

posisi jari, dan ekspresi tubuh untuk menyampaikan informasi [3].

Walaupun SIBI telah memberikan kontribusi besar dalam mendukung komunikasi bagi penyandang disabilitas, pemahaman dan penguasaan terhadap sistem ini belum tersebar secara merata, termasuk di kalangan pengembang teknologi.[4]. Hal ini menimbulkan kebutuhan akan sistem yang mampu mengenali bahasa isyarat secara otomatis, dimulai dari gestur dasar seperti alfabet. Tujuan dari penelitian ini bukan untuk menciptakan penerjemah penuh bahasa isyarat, melainkan membangun sistem awal yang mampu mengenali dan mengklasifikasikan huruf dalam SIBI secara akurat melalui pemrosesan komputasi [5].

Perkembangan teknologi pengolahan citra dan kecerdasan buatan memberikan peluang besar untuk mengembangkan sistem pengenalan bahasa isyarat [6]. Salah satu teknologi yang sering dimanfaatkan dalam pengolahan fitur visual adalah MediaPipe, yang mampu mendeteksi titik-titik penting pada tangan secara real-time dengan presisi tinggi [7]. Informasi koordinat dari titik-titik ini dapat dianalisis untuk mengenali pola gerakan tangan yang menjadi dasar bahasa isyarat.

Agar sistem mampu mengenali pola bahasa isyarat dengan baik, diperlukan algoritma klasifikasi yang andal. Pada penelitian ini digunakan metode *Feedforward Neural Network* (FFNN) yang mampu memproses data numerik hasil ekstraksi dari *MediaPipe* [8]. FFNN memiliki kemampuan untuk mengidentifikasi pola dari data koordinat landmark tangan secara efisien dan terstruktur. Sementara itu, *MediaPipe* digunakan sebagai alat utama untuk mengekstrak fitur visual berupa titik-titik kunci tangan dari citra input secara *real-time* [9]. Kombinasi antara *MediaPipe* dan FFNN ini memungkinkan sistem yang dikembangkan untuk mendeteksi, mengklasifikasikan, dan mengenali isyarat huruf alfabet berdasarkan standar Sistem Isyarat Bahasa Indonesia (SIBI) dengan tingkat keakuratan yang tinggi melalui input kamera secara langsung.

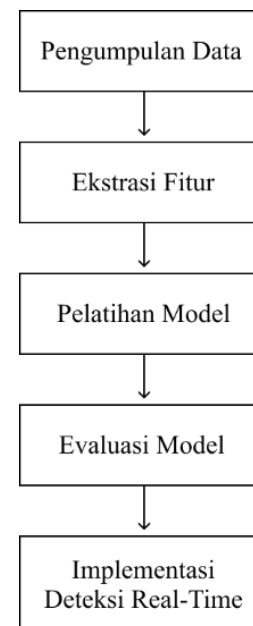
Penelitian ini difokuskan pada pengembangan sistem klasifikasi alfabet dalam SIBI berbasis citra tangan secara real-time. Dengan memanfaatkan *MediaPipe* untuk proses ekstraksi fitur dan FFNN sebagai model klasifikasi, penelitian ini berupaya

menguji performa sistem dalam mengenali gestur huruf secara akurat. Hasil dari pengujian diharapkan dapat menjadi pijakan awal untuk pengembangan lebih lanjut sistem penerjemah bahasa isyarat yang lebih luas dan aplikatif di masa mendatang.

## II. METODOLOGI PENELITIAN

Penelitian ini menggunakan pendekatan eksperimen yang berorientasi pada pengujian sistem berbasis kecerdasan buatan untuk mendeteksi huruf dalam bahasa isyarat Indonesia (SIBI) secara real-time. Metode ini dipilih karena sesuai untuk mengevaluasi kinerja sistem dalam mengenali gestur bahasa isyarat berdasarkan data yang tersedia. Proses pengembangan mengikuti alur *machine learning pipeline* yang terstruktur dalam beberapa tahapan utama, yakni pengumpulan data, ekstraksi fitur, pelatihan model, evaluasi performa, hingga penerapan sistem secara langsung [10].

Pendekatan *machine learning pipeline* digunakan karena proses pengembangan sistem ini melibatkan beberapa tahapan berurutan, mulai dari pengumpulan data, ekstraksi fitur, pelatihan model, evaluasi kinerja, hingga implementasi deteksi secara langsung. Berikut alur tahapan penelitian menggunakan metode eksperimen.



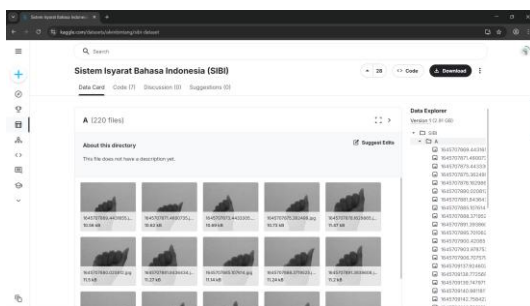
Gambar 1. Tahapan penelitian

Berdasarkan Gambar 1, tahapan penelitian ini menggunakan metode eksperimen dengan

pendekatan *machine learning pipeline*, yang terdiri dari lima tahap utama, yaitu pengumpulan data, ekstraksi fitur, pelatihan model, evaluasi model, dan implementasi deteksi *real-time*.

#### A. Pengumpulan Data

Data yang digunakan dalam penelitian ini diperoleh dari platform Kaggle dan terdiri dari gambar tangan yang merepresentasikan huruf-huruf alfabet dalam bahasa isyarat Indonesia (SIBI). Dataset mencakup 24 kelas huruf, dengan jumlah gambar yang seimbang untuk setiap kelas, dan tidak menyertakan huruf J dan Z, karena kedua huruf tersebut memerlukan gerakan dinamis yang belum didukung oleh pembacaan sistem.



Gambar 2. Dataset bahasa isyarat SIBI dari Kaggle

Dataset yang digunakan hanya memuat gambar statis, sehingga huruf J dan Z tidak disertakan. Setiap huruf pada dataset terdiri dari 220 gambar, yang mencakup berbagai variasi posisi tangan dan kondisi pencahayaan untuk meningkatkan ketahanan model terhadap input yang beragam.

#### B. Ekstraksi Fitur

Setelah data terkumpul, dilakukan ekstraksi fitur menggunakan MediaPipe. MediaPipe akan mendeteksi 21 titik landmark tangan dari gambar, masing-masing dalam bentuk koordinat tiga dimensi (x, y, z). Data numerik hasil ekstraksi ini menjadi representasi posisi tangan yang digunakan dalam proses pelatihan model.

#### C. Pelatihan Model

Pelatihan model dilakukan menggunakan algoritma *Feedforward Neural Network* (FFNN) yang dilatih dengan data hasil ekstraksi landmark tangan. Dataset dibagi menjadi data pelatihan dan pengujian dengan rasio 80:20 untuk memastikan

keseimbangan antara proses pembelajaran dan pengujian. Evaluasi Model

Setelah proses pelatihan selesai, model dievaluasi menggunakan data pengujian untuk mengukur akurasi klasifikasi. Evaluasi dilakukan guna mengetahui seberapa baik model dalam mengenali masing-masing kelas huruf yang telah dilatih. Hasil dari evaluasi ini menjadi dasar untuk menilai performa model sebelum digunakan dalam implementasi deteksi *real-time*

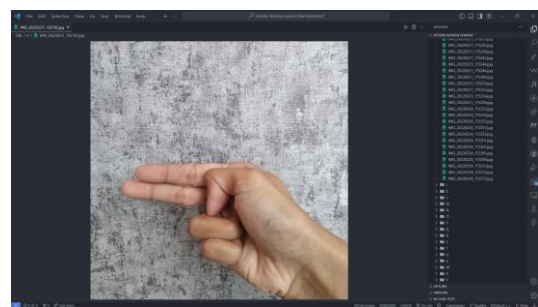
#### D. Implementasi Deteksi Real-Time

Model yang tervalidasi selanjutnya diimplementasikan dalam sistem deteksi secara real-time menggunakan kamera laptop. Sistem akan membaca citra tangan melalui webcam dan mengklasifikasikan huruf SIBI berdasarkan output dari model FFNN, yang ditampilkan secara langsung melalui terminal. Deteksi dilakukan secara berkelanjutan selama sistem masih berjalan, tanpa batasan waktu tertentu, selama *webcam* aktif dan program dijalankan.

### III. HASIL DAN PEMBAHASAN

#### A. Hasil Ekstraksi Fitur

Proses ekstraksi fitur dilakukan terhadap dataset gambar tangan yang merepresentasikan huruf-huruf alfabet dalam bahasa isyarat Indonesia (SIBI). Gambar-gambar tersebut dianalisis menggunakan MediaPipe, sebuah framework berbasis machine learning yang mampu mendeteksi 21 titik landmark pada tangan secara akurat. Setiap landmark memiliki tiga koordinat posisi (x, y, z), sehingga setiap gambar dikonversi menjadi 63 fitur numerik yang menjadi representasi spasial dari gestur tangan.



Gambar 3. Model gambar untuk tahap ekstraksi fitur dengan Mediapipe

Seluruh gambar diproses secara otomatis melalui skrip yang membaca setiap file, melakukan deteksi tangan, dan mengekstrak koordinat *landmark*. Proses ini direpresentasikan dalam potongan kode berikut:

```
for hand_landmarks in results.multi_hand_landmarks:
    for landmark in hand_landmarks.landmark:
        keypoints.extend([landmark.x, landmark.y,
                           landmark.z])
```

Jika sistem tidak berhasil mendeteksi tangan pada suatu gambar, maka fitur akan digantikan dengan nilai nol sebanyak 63 elemen untuk menjaga konsistensi dimensi data. Nilai 63 ini berasal dari jumlah total fitur yang diharapkan, yaitu 21 titik *landmark* pada tangan yang masing-masing memiliki tiga komponen koordinat: x, y, dan z. Dengan demikian, meskipun tidak ada tangan terdeteksi, sistem tetap menghasilkan vektor berdimensi sama ( $21 \times 3 = 63$ ), sehingga tidak mengganggu struktur input yang dibutuhkan oleh model klasifikasi. dengan nilai nol sebanyak 63 elemen untuk menjaga konsistensi dimensi data:

```
keypoints.extend([0.0] * (21 * 3))
```

Selanjutnya, hasil dari proses ini disimpan sebagai file numerik *.npy* dan dikelompokkan sesuai dengan kelas huruf masing-masing. Format penyimpanan *.npy* dipilih karena merupakan format standar NumPy yang efisien untuk menyimpan array numerik dalam bentuk biner. Format ini memudahkan proses pembacaan ulang data dalam tahap pelatihan dan evaluasi model karena mendukung pembacaan cepat serta tidak memerlukan parsing tambahan seperti pada format teks. Proses penyimpanan data dilakukan melalui perintah berikut:

```
output_filename = os.path.splitext(image_file)[0] +
'.npy'

output_path = os.path.join(PROCESSED_DATA_DIR,
                             class_name, output_filename)

np.save(output_path, np.array(keypoints))
```

Setelah proses dijalankan, sistem menampilkan keluaran berupa proses pemrosesan setiap kelas huruf secara berurutan. Berikut adalah ringkasan hasil keluaran dari terminal saat menjalankan perintah:

```
# python extract_landmark.py
Memulai ekstraksi landmark dari gambar mentah...
Memproses kelas: A (220 gambar)
Selesai 100/220 gambar di A
Selesai 200/220 gambar di A
Memproses kelas: B (220 gambar)
Selesai 100/220 gambar di B
Selesai 200/220 gambar di B
...
Memproses kelas: Y (220 gambar)
Selesai 100/220 gambar di Y
Selesai 200/220 gambar di Y
Ekstraksi landmark selesai.
```

Berdasarkan hasil tersebut, mayoritas gambar berhasil diproses dengan baik oleh MediaPipe, dengan deteksi landmark yang stabil. Sebagian kecil gambar (kurang dari 5%) tidak terdeteksi dengan sempurna karena posisi tangan yang buram, terlalu kecil, atau tidak sesuai posisi standar. Namun, secara keseluruhan, proses ekstraksi fitur dapat dikatakan berhasil membentuk dataset numerik yang representatif dan dapat diandalkan. Dengan total 24 kelas dan 220 gambar per kelas, sistem menghasilkan sekitar 5.280 file data fitur (*.npy*) yang siap digunakan dalam proses pelatihan. Jumlah file ini sebanding dengan jumlah total gambar yang berhasil diproses, karena setiap gambar menghasilkan satu file *.npy*. Ketersediaan data fitur yang konsisten menjadi dasar penting bagi model klasifikasi untuk mengenali pola gerakan tangan dari masing-masing huruf secara akurat pada tahap selanjutnya.

### B. Hasil Pelatihan Model

Pelatihan dilakukan menggunakan algoritma *Feedforward Neural Network* (FFNN) dengan input berupa data numerik hasil ekstraksi *landmark* tangan. Dataset yang telah diproses terdiri dari 24 kelas huruf dengan masing-masing 220 sampel, sehingga total terdapat 5.280 data. Data ini kemudian dibagi menjadi dua bagian menggunakan rasio 80:20, yaitu 4.224 data untuk pelatihan dan 1.056 data untuk pengujian. Pembagian dilakukan dengan fungsi *train\_test\_split* dari Scikit-learn, menggunakan parameter *stratify* agar distribusi label tetap seimbang. Potongan kode berikut menunjukkan proses pembagian data:

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y,
    test_size=0.2,
    random_state=42,
    stratify=y)
```

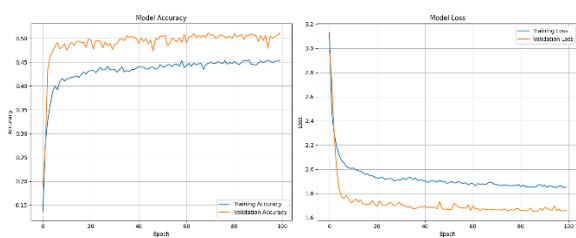
Arsitektur model FFNN yang digunakan terdiri dari dua *hidden layer*. *Hidden layer* pertama memiliki 128 *neuron* dan *hidden layer* kedua memiliki 64 *neuron*, masing-masing menggunakan fungsi aktivasi ReLU. Untuk mencegah *overfitting*, digunakan *Batch Normalization* dan *Dropout* sebesar 30% pada setiap *hidden layer*. *Output layer* terdiri dari 24 *neuron* sesuai jumlah kelas, dengan fungsi aktivasi *softmax* untuk menghasilkan probabilitas dari setiap kelas. Model dibangun dengan *library* Keras menggunakan struktur sebagai berikut:

```
model = Sequential()
model.add(Dense(128, activation='relu', input_shape=(63,)))
model.add(BatchNormalization())
model.add(Dropout(0.3))
model.add(Dense(64, activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.3))
model.add(Dense(num_classes, activation='softmax'))
```

Selanjutnya, model dilatih selama 100 *epoch* menggunakan *Adaptive Moment Estimation* dan fungsi *loss categorical\_crossentropy*. Pada *epoch* pertama, model mencapai akurasi pelatihan sebesar 13.68% dan akurasi validasi sebesar 18.84%. Setelah 100 *epoch*, akurasi pelatihan meningkat menjadi 45.24% dan akurasi validasi mencapai 51.23%.

```
Epoch 1/100
132/132 [=====] - 1s
3ms/step - loss: 3.1306 - accuracy: 0.1368 - val_loss:
2.9860 - val_accuracy: 0.1884
...
Epoch 100/100
132/132 [=====] - 0s
2ms/step - loss: 1.8504 - accuracy: 0.4524 - val_loss:
1.6605 - val_accuracy: 0.5123
```

Grafik menunjukkan perkembangan akurasi dan loss selama 100 *epoch* pelatihan, baik untuk data latih maupun data validasi. Terlihat bahwa terjadi peningkatan performa yang stabil tanpa indikasi *overfitting* yang signifikan.



Gambar 4. Grafik akurasi dan loss pelatihan data

Berikut adalah grafik yang menunjukkan akurasi pelatihan untuk masing-masing kelas huruf SIBI. Beberapa huruf seperti 'P' memiliki akurasi mendekati sempurna, sementara beberapa huruf lain seperti 'Q' dan 'O' menunjukkan akurasi yang lebih rendah.



Gambar 5. Akurasi pelatihan tiap kelas huruf

Model yang telah dilatih kemudian disimpan dalam format *.h5* untuk digunakan pada tahap implementasi deteksi *real-time*. Penggunaan rasio 80:20 dalam pembagian data pelatihan dan data pengujian memberikan keseimbangan yang baik antara proses pembelajaran dan evaluasi. Dengan 80% data digunakan untuk melatih model, sistem memperoleh cukup banyak sampel untuk mempelajari pola dari masing-masing kelas huruf. Sementara itu, 20% sisanya digunakan sebagai data uji untuk menilai sejauh mana model mampu menggeneralisasi pada data yang belum pernah dilihat sebelumnya. Hasil pelatihan yang menunjukkan akurasi validasi yang relatif sejalan dengan akurasi pelatihan menjadi indikator bahwa pembagian ini efektif dalam mengurangi risiko *overfitting* dan menghasilkan model yang stabil.

### C. Hasil Evaluasi Model

Setelah proses pelatihan selesai, model dievaluasi menggunakan data pengujian untuk mengukur sejauh mana performa klasifikasi terhadap data yang belum pernah dilihat sebelumnya. Evaluasi ini penting untuk mengetahui kemampuan generalisasi model serta mendeteksi potensi *overfitting*. Data pengujian yang digunakan adalah 20% dari total dataset, yang sebelumnya telah dipisahkan menggunakan teknik stratifikasi agar distribusi label tetap merata. Proses evaluasi dilakukan dengan menghitung akurasi keseluruhan serta analisis akurasi per kelas.

Prediksi dilakukan dengan menghitung  $\text{argmax}$  dari hasil softmax model untuk menentukan kelas dengan probabilitas tertinggi. Potongan kode berikut digunakan untuk menghasilkan prediksi dan label sebenarnya dari data pelatihan atau pengujian:

```
y_pred_train = model.predict(X_train)
y_pred_train_classes = np.argmax(y_pred_train, axis=1)
y_true_train_classes = np.argmax(y_train, axis=1)
```

Berdasarkan hasil akhir pada epoch ke-100, model mencapai akurasi validasi sebesar 51.23% pada data pengujian. Nilai ini diperoleh dari output terminal pelatihan sebagai berikut:

```
Epoch 100/100 132/132 [=====]
- 0s 2ms/step - loss: 1.8504 - accuracy: 0.4524 -
val_loss: 1.6605 - val_accuracy: 0.5123
```

Angka  $\text{val\_accuracy}$ : 0.5123 menunjukkan bahwa dari seluruh prediksi yang dilakukan terhadap data pengujian, sekitar 51.23% berhasil diklasifikasikan dengan benar. Nilai ini sejalan dengan tren pelatihan sebelumnya yang menunjukkan peningkatan stabil tanpa gejala *overfitting* signifikan. Untuk analisis lebih lanjut, digunakan *confusion matrix* dan grafik akurasi per kelas, yang menunjukkan bahwa model mampu mengenali beberapa huruf seperti P, M, dan L dengan akurasi tinggi, sementara beberapa huruf seperti Q, O, dan C sering mengalami kesalahan klasifikasi karena kemiripan gestur. Evaluasi ini menjadi acuan penting dalam mengukur kekuatan dan kelemahan model serta memberikan arah perbaikan pada tahap pengembangan berikutnya.

#### D. Hasil Implementasi Data Real-Time

Model klasifikasi yang telah dilatih kemudian diimplementasikan dalam sistem deteksi bahasa isyarat secara *real-time*. Implementasi ini dilakukan menggunakan *webcam* laptop sebagai media input, di mana sistem akan secara langsung menangkap citra tangan pengguna dan mendeteksinya menggunakan *MediaPipe*. Setelah proses deteksi *landmark* tangan selesai, data koordinat akan diteruskan ke model FFNN untuk diklasifikasikan menjadi salah satu huruf dari alfabet bahasa isyarat Indonesia (SIBI).

Proses deteksi *real-time* dijalankan melalui program yang secara terus-menerus membaca input video dari kamera laptop, memproses setiap *frame* untuk mendeteksi keberadaan tangan, dan

melakukan prediksi terhadap huruf bahasa isyarat yang dikenali. Sistem ini bekerja secara berkelanjutan menggunakan logika perulangan yang memastikan bahwa setiap *frame* video dianalisis selama program berjalan. Hasil prediksi ditampilkan secara langsung pada tampilan video (*frame*) dalam bentuk teks yang menunjukkan huruf yang terdeteksi beserta nilai akurasi dari model. Dengan demikian, pengguna dapat melihat secara langsung huruf hasil klasifikasi serta tingkat keyakinan sistem terhadap hasil tersebut. Tidak terdapat batasan durasi tertentu dalam proses ini. Selama aplikasi aktif, pengguna dapat melakukan *gesture* kapan saja dan sistem akan tetap melakukan klasifikasi secara langsung.

Berikut adalah potongan kode yang digunakan untuk melakukan deteksi secara *real-time*:

```
cap = cv2.VideoCapture(0)
while True:
    ret, frame = cap.read()
    if not ret:
        break

    results = hands.process(cv2.cvtColor(frame,
cv2.COLOR_BGR2RGB))

    if results.multi_hand_landmarks:
        for hand_landmarks in results.multi_hand_landmarks:
            keypoints = []
            for lm in hand_landmarks.landmark:
                keypoints.extend([lm.x, lm.y, lm.z])
            keypoints = np.array(keypoints).reshape(1, -1)
            prediction = model.predict(keypoints)
            predicted_class = np.argmax(prediction)

            print(f"Prediksihuruf:{class_names[predicted_class]}")
```

Gambar berikut memperlihatkan cuplikan hasil deteksi sistem saat melakukan pengenalan gesture huruf A, B, dan D secara *real-time*. Ketiga gambar ini merupakan contoh pengujian yang diambil secara langsung selama proses implementasi berjalan menggunakan *webcam* laptop. Masing-masing gambar menunjukkan sistem berhasil mendeteksi titik-titik landmark tangan menggunakan *MediaPipe* dan melakukan klasifikasi huruf melalui model FFNN.

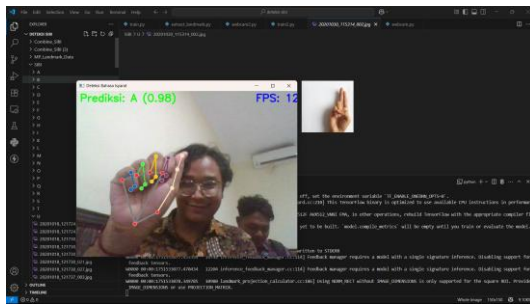
Pengujian sistem dilakukan terhadap tiga *gesture* huruf, yaitu A, B, dan D. Pengujian pertama menunjukkan hasil deteksi gesture huruf A, dengan prediksi yang sangat akurat ditandai oleh *confidence score* sebesar 0.98. Sistem menampilkan huruf A pada tampilan video dengan penempatan teks yang mengikuti posisi tangan.

## IV. KESIMPULAN

Penelitian ini berhasil mengembangkan sistem deteksi huruf alfabet dalam bahasa isyarat Indonesia (SIBI) secara *real-time* dengan menggunakan kombinasi metode MediaPipe untuk ekstraksi fitur dan *Feedforward Neural Network* (FFNN) sebagai model klasifikasi. Dari hasil pelatihan model, sistem menunjukkan peningkatan akurasi yang stabil hingga mencapai akurasi validasi sebesar 51,23%. Hasil evaluasi per kelas menunjukkan bahwa beberapa huruf seperti P dan M dikenali dengan sangat baik, sementara huruf seperti Q dan O masih mengalami kekeliruan karena kemiripan gestur. Implementasi deteksi secara *real-time* berhasil menunjukkan bahwa sistem mampu mengklasifikasikan *gesture* huruf A, B, dan D dengan *confidence score* berturut-turut sebesar 0.98, 0.89, dan 0.94. Capaian ini membuktikan bahwa sistem memiliki kemampuan dalam mengenali bahasa isyarat secara langsung dengan tingkat akurasi yang mendekati 1. Dengan demikian, tujuan penelitian untuk mengembangkan sistem klasifikasi SIBI berbasis citra *real-time* telah tercapai. Penelitian ini juga memberikan fondasi awal yang kuat untuk pengembangan sistem penerjemah bahasa isyarat yang lebih kompleks dan luas di masa depan

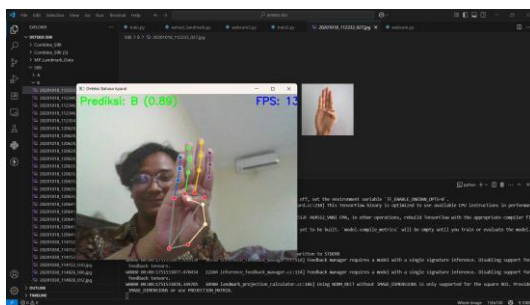
## REFERENSI

- [1] O. Mailani, I. Nuraeni, S. A. Syakila, J. Lazuardi, and P. I. Komunikasi, "Bahasa Sebagai Alat Komunikasi Dalam Kehidupan Manusia," *KAMPRET Journal*, vol. 1, no. 2, pp. 1–10, Jan. 2022, doi: <https://doi.org/10.35335/kampret.v1i1.8>.
- [2] Zakia Husna, Silvia Rizki, Fitriani Fitriani, and Ulfa Amalia, "Model Penanganan Anak Berkebutuhan Khusus Tuna Wicara pada Sekolah Jalan Ceurih Kecamatan Ule Kareng, Kota Banda Aceh," *Khirani: Jurnal Pendidikan Anak Usia Dini*, vol. 3, no. 1, pp. 232–242, May 2025, doi: 10.47861/khirani.v3i1.1743.
- [3] R. Nanda, "Perancangan Aplikasi Tuna Wicara Dan Tuna Rungu Dengan Metode Waterfall Berbasis Android," *Jurnal Teknik Informatika (JEKIN)*, vol. 2, no. 3, pp. 132–141, Jan. 2022, doi: <https://doi.org/10.58794/jekin.v3i1.189>.
- [4] P. Rika Perdana, F. Meiliawati, Y. Rukmayadi, and T. Ardianto, "Efektifitas Pelatihan Bahasa Isyarat Dasar bagi Calon Guru Sekolah Inklusi di Wilayah Provinsi Banten," *JOURNAL OF DISABILITY STUDIES AND RESEARCH (JDSR)*, vol. 1, pp. 06–22, 2022, doi: <https://doi.org/10.30631/jdsr.v1i1.1328>.
- [5] M. Sholawati, K. Auliasari, and F. X. Ariwibisono, "Pengembangan Aplikasi Pengenalan Bahasa Isyarat Abjad SIBI Menggunakan Metode Convolutional Neural Network (CNN)," *Jurnal Mahasiswa Teknik Informatika*, vol. 6, no. 1, pp. 134–44, Feb. 2022, doi: <https://doi.org/10.36040/jati.v6i1.4507>.
- [6] A. Ayu, S. Mustofa, F. Ma'rouf, and M. Ferdiansyah, "Implementasi Citra Digital Untuk Permainan Suit Bagi Anak-Anak," *PROSIDING SEMINAR NASIONAL TEKNOLOGI DAN SAINS*, vol. 4, pp. 214–222, 2025.



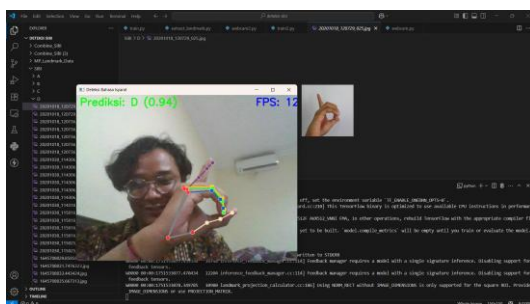
Gambar 6. Pengujian deteksi real-time untuk huruf A

Selanjutnya, hasil pengenalan *gesture* huruf B, yang diklasifikasikan dengan *confidence score* sebesar 0.89. Walaupun lebih rendah dari huruf A, hasil ini tetap menunjukkan prediksi yang kuat dan *respons* yang stabil terhadap bentuk *gesture*.



Gambar 7. Pengujian deteksi real-time untuk huruf B

Pengujian *gesture* huruf D, yang dikenali dengan *confidence score* sebesar 0.94. Deteksi berjalan lancar, dan sistem mampu membedakan *gesture* ini dari huruf lain yang memiliki bentuk jari serupa.



Gambar 8. Pengujian deteksi real-time untuk huruf D

Perlu dicatat bahwa gambar ini hanya merupakan sampel dari deteksi *real-time*, yang dipilih untuk menggambarkan bagaimana sistem bekerja dalam kondisi nyata. Selama aplikasi dijalankan, pengguna dapat melakukan berbagai *gesture* dan sistem akan memprosesnya secara kontinu tanpa batasan durasi.

- [7] K. Feny Sugiantari, P. Hendra Suputra, L. Joni Erawati Dewi, G. Bakti Pratama Putra, and K. Kunci Klasifikasi, "Pendekatan MLP dalam Klasifikasi Bahasa Isyarat: Analisis Jarak Euclidean Landmark Tangan," *Jurnal Informatika (JIKA)*, vol. 9, no. 2, pp. 209–218, 2025, doi: <http://dx.doi.org/10.31000/jika.v9i2.13368>.
- [8] G. F. A. YUDHISTIRA, S. SUTEDJO, and R. RAKHMAWATI, "Distribution Transformer Secondary Bushing Temperature Detection Device using Feed Forward Neural Network," *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 11, no. 4, p. 983, Oct. 2023, doi: [10.26760/elkomika.v11i4.983](https://doi.org/10.26760/elkomika.v11i4.983).
- [9] K. Feny Sugiantari, P. Hendra Suputra, L. Joni Erawati Dewi, G. Bakti Pratama Putra, and K. Kunci Klasifikasi, "Pendekatan MLP dalam Klasifikasi Bahasa Isyarat: Analisis Jarak Euclidean Landmark Tangan," *Jurnal Informatika (JIKA)*, vol. 9, no. 2, pp. 209–218, Apr. 2025.
- [10] A. Pramudito, P. R. Wanarti, M. Rohman, and L. Anifah Afiliati, "Analisis dan Simulasi Sistem Kontrol Suhu Otomatis Berbasis Fuzzy Logic," *Jurnal Teknik Elektro*, vol. 14, pp. 43–47, Jul. 2024, doi: <https://doi.org/10.26740/jte.v14n1.p43-47>.