Machine Learning System untuk Mendeteksi Gerakan Tubuh Menggunakan Library Mediapipe

**Irfan Nurdiansyah 1; Reni Utami 2; Muchamad Sandy 3**

*1,2,3 Fakultas Ilmu Komputer, Universitas Dian Nusantara, Jl. Tj. Duren Barat. 2 No.1, RT.1/RW.5*

*1irfan.nurdiansyah@dosen.undira.ac.id, 2 reni.utami@dosen.undira.ac.id, 3 muchamad.sandy@dosen.undira.ac.id*

Kata kunci:

Machine Learning, Mediapipe, Body Movement Detection, Motion Classification, Computer Vision

**Abstract**

Human body movement detection is a crucial challenge in various fields such as sports, healthcare, and human-computer interaction. Conventional methods often require expensive specialized hardware or complex processing techniques. This study aims to develop a Machine Learning-based system capable of detecting and classifying human body movements in real-time using the Mediapipe library.

The methodology includes collecting movement data, extracting features using Mediapipe to detect keypoints on the body, and preprocessing the data before applying Machine Learning models. Classification models such as Random Forest, Support Vector Machine (SVM), and Neural Networks are implemented to recognize movement patterns. The system's performance is evaluated using metrics such as accuracy, precision, recall, and F1-score.

The results indicate that the Mediapipe-based approach can detect body movements with high accuracy. This system is more lightweight compared to sensor-based or specialized hardware methods, making it suitable for broader applications.

**Pendahuluan**

Bahasa isyarat, sebagai modalitas komunikasi utama bagi komunitas tunarungu, memiliki peran krusial dalam interaksi sosial dan aksesibilitas informasi. Namun, perbedaan antara bahasa isyarat dan bahasa lisan/tulis seringkali menjadi penghalang komunikasi [1]. Oleh karena itu, pengembangan sistem machine learning untuk deteksi bahasa isyarat menjadi penting guna menjembatani kesenjangan komunikasi ini. Sistem ini diharapkan mampu menerjemahkan bahasa isyarat ke dalam bahasa yang dapat dipahami oleh masyarakat umum, sehingga meningkatkan inklusi dan partisipasi komunitas tunarungu dalam berbagai aspek kehidupan.

Penelitian ini difokuskan pada perancangan dan implementasi sistem machine learning [2] yang mampu mengenali bahasa isyarat. Sistem ini memanfaatkan pemrosesan citra [3] dan deep learning [4] untuk mengidentifikasi gerakan tangan [5], ekspresi wajah, dan elemen visual lainnya yang merupakan ciri khas bahasa isyarat. Tujuan utama adalah menciptakan sistem yang akurat dan efisien dalam menerjemahkan bahasa isyarat ke dalam teks atau suara [6], sehingga memudahkan komunikasi antara individu tunarungu dan masyarakat umum.

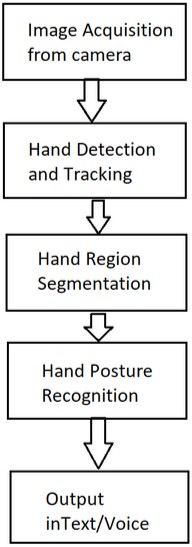
Dalam penelitian ini, kami memanfaatkan MediaPipe [7], sebuah framework sumber terbuka yang dikembangkan oleh Google, untuk memfasilitasi deteksi dan pelacakan gerakan tangan serta ekstraksi fitur-fitur penting dari bahasa isyarat. MediaPipe menyediakan solusi yang efisien dan real-time untuk pemrosesan data multimodal, termasuk video. Modul Hand Pose Estimation [8] dari MediaPipe memungkinkan identifikasi titik-titik kunci (landmark) pada tangan dengan akurasi yang cukup tinggi. Informasi ini kemudian digunakan sebagai input untuk model machine learning yang bertugas menerjemahkan bahasa isyarat.

Penggunaan MediaPipe [9] dalam penelitian ini memberikan beberapa keuntungan. Pertama, framework ini mudah diintegrasikan dengan berbagai bahasa pemrograman dan platform, sehingga mempercepat proses pengembangan sistem. Kedua, MediaPipe telah dioptimalkan untuk kinerja tinggi, memungkinkan pemrosesan video secara real-time tanpa mengorbankan akurasi. Ketiga, ketersediaan model pre-trained untuk deteksi tangan [10] dan estimasi pose mengurangi kebutuhan sumber daya komputasi dan waktu pelatihan model. Dengan memanfaatkan MediaPipe, penelitian ini dapat lebih fokus pada pengembangan algoritma machine learning yang efektif untuk penerjemahan bahasa isyarat, alih-alih membangun sistem deteksi tangan dari awal.

Oleh karena itu, batasan penelitian ini meliputi kemampuan sistem dalam mengenali satu huruf untuk setiap input, keterbatasan deteksi tubuh, dan belum mampunya sistem untuk merangkai beberapa huruf menjadi kata utuh. Meskipun demikian, penelitian ini merupakan langkah awal yang penting dalam pengembangan sistem deteksi bahasa isyarat yang lebih komprehensif. Penelitian lebih lanjut akan difokuskan pada peningkatan akurasi dan kemampuan sistem dalam mengenali kata dan kalimat, serta memperluas cakupan deteksi bahasa isyarat.

**Metode penelitian**

Untuk metode penelitian pada sistem ini terdiri dari dua bagian utama, yaitu perangkat perekam dan komputer. Perangkat perekam bertugas menangkap gerakan isyarat yang dilakukan oleh penyandang tunarungu, kemudian mengirimkan informasi tersebut ke komputer. Komputer kemudian menganalisis data tersebut untuk mengungkap makna dari isyarat yang dilakukan. Penjelasan ini biasanya dilengkapi dengan ilustrasi visual.

****

**Gambar 1.** Proses dalam proses penelitian.

## **Tahapan Penelitian**

Berdasarkan diagram alur atau flowchart yang menggambarkan proses pengenalan atau penerjemahan bahasa isyarat menjadi teks atau suara. Diagram ini terdiri dari beberapa tahapan yang berurutan, dimulai dari akuisisi gambar hingga menghasilkan keluaran berupa teks atau suara, antara lain :

1. **Image Acquisition from Camera (Pengambilan Gambar dari Kamera)** : Tahap ini merupakan langkah awal dalam proses. Kamera digunakan untuk mengambil gambar atau video dari bahasa isyarat yang dilakukan oleh pengguna. Gambar atau video ini akan menjadi input untuk tahapan selanjutnya.
2. **Hand Detection and Tracking (Deteksi dan Pelacakan Tangan)** : Setelah gambar diambil, sistem akan mendeteksi dan melacak keberadaan tangan dalam gambar atau video. Proses ini melibatkan identifikasi posisi tangan, ukuran, dan gerakan tangan dari waktu ke waktu.
3. **Hand Region Segmentation (Segmentasi Wilayah Tangan)** : Pada tahap ini, sistem akan memisahkan atau mengekstrak wilayah tangan dari bagian gambar atau video lainnya. Tujuannya adalah untuk memfokuskan analisis pada tangan yang melakukan bahasa isyarat.
4. **Hand Posture Recognition (Pengenalan Postur Tangan)** : Setelah wilayah tangan tersegmentasi, sistem akan mengenali postur atau bentuk tangan yang digunakan dalam bahasa isyarat. Setiap postur tangan memiliki arti atau simbol tertentu dalam bahasa isyarat.
5. **Output in Text/Voice (Keluaran dalam Teks/Suara)** : Tahap terakhir adalah menghasilkan keluaran berdasarkan postur tangan yang dikenali. Keluaran ini dapat berupa teks yang ditampilkan pada layar atau suara yang dihasilkan oleh speaker. Teks atau suara ini merupakan terjemahan dari bahasa isyarat yang dilakukan oleh pengguna.

## **Research Framework**

Framework penelitian ini dirancang untuk menguraikan tahapan utama dalam pengembangan sistem deteksi gerakan tubuh berbasis Machine Learning menggunakan Mediapipe. Proses dimulai dengan akuisisi data, di mana kamera digunakan untuk menangkap pergerakan tubuh manusia dalam bentuk video atau gambar. Selanjutnya, sistem melakukan ekstraksi fitur, yaitu mendeteksi titik-titik kunci (keypoints) pada tubuh menggunakan Mediapipe, yang merepresentasikan gerakan secara numerik. Setelah fitur diperoleh, tahap pra-pemrosesan data dilakukan untuk menormalkan koordinat dan menghilangkan informasi yang tidak relevan, sehingga data siap digunakan dalam model Machine Learning.

Pada tahap pengembangan model, beberapa algoritma pembelajaran mesin seperti Support Vector Machine (SVM), Random Forest, dan Neural Network diterapkan untuk mengenali pola gerakan tubuh. Model yang telah dilatih kemudian diuji pada tahap evaluasi model untuk mengukur performanya berdasarkan metrik akurasi, presisi, recall, dan F1-score. Hasil akhir dari framework ini adalah sistem yang mampu mendeteksi dan mengklasifikasikan gerakan tubuh dengan akurasi tinggi, memungkinkan penggunaannya dalam berbagai aplikasi seperti kesehatan, olahraga, dan interaksi manusia-komputer.



**Gambar 2** **Proses Research Framework**

Diagram di atas menunjukkan Research Framework, yang menggambarkan alur kerja penelitian mulai dari akuisisi data hingga evaluasi model.

1. **Metode Penelitian**

Metode penelitian ini terdiri dari beberapa tahap utama yang memastikan sistem deteksi gerakan tubuh dapat berfungsi secara optimal. Tahap pertama adalah pengumpulan data, di mana kamera digunakan untuk menangkap gambar atau video dari berbagai gerakan tubuh manusia. Data ini kemudian diolah dalam tahap ekstraksi fitur, di mana titik-titik kunci pada tubuh manusia dideteksi menggunakan Mediapipe. Setiap titik kunci ini mewakili koordinat tubuh dalam bentuk numerik, yang nantinya digunakan sebagai fitur untuk model Machine Learning.

Setelah fitur diperoleh, data melewati tahap pra-pemrosesan, yang mencakup normalisasi koordinat untuk memastikan skala data seragam dan menghilangkan bias dalam proses pelatihan model. Data yang telah diproses kemudian digunakan dalam tahap pengembangan model, di mana algoritma Machine Learning diterapkan untuk mengenali dan mengklasifikasikan gerakan tubuh. Model yang telah dilatih diuji dalam tahap evaluasi model, menggunakan metrik seperti akurasi, presisi, recall, dan F1-score untuk menentukan efektivitas sistem. Setelah evaluasi selesai, model yang memiliki performa terbaik digunakan untuk mendeteksi gerakan tubuh secara real-time, dengan hasil akhir ditampilkan dalam bentuk teks atau visualisasi gerakan.



**Gambar 3 Metode Penelitian**

Diagram di atas menunjukkan Metode Penelitian, yang menjelaskan tahapan utama dari pengumpulan data hingga evaluasi model.

**Hasil dan diskusi**

Penelitian ini dikembangkan dengan menggunakan bahasa pemrograman Python serta memanfaatkan alat serta teknik pembelajaran mendalam untuk berbagai fungsi, termasuk Pelacakan Tangan. Proses pada penelitian ini yaitu sistem menangkap gambar dari kamera, kemudian mendeteksi titik-titik kunci gerakan tangan, serta memproses data gambar, dan membuat keputusan berdasarkan model yang telah dilatih dengan metode Machine Learning.

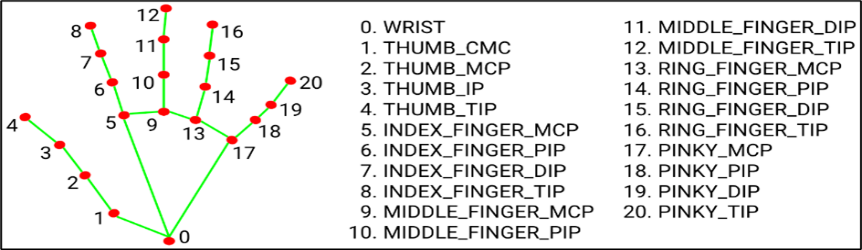
Pada penelitian ini, proses implementasi sistem dirancang untuk mengenali 24 huruf yang digunakan dalam bahasa isyarat, dengan langkah awal dengan pengambilan gambar gestur tangan, kemudian melakukan beberapa proses implementasi, antara lain :



**Gambar 4.** Gestur tangan yang digunakan dalam proses penelitian

* 1. Deteksi Titik Utama

Langkah awal pada proses implementasi adalah menangkap gerakan tangan menggunakan kamera, dengan bantuan pustaka Python OpenCV untuk menghasilkan gambar dalam format BGR. Setelah itu, dilakukan penyesuaian seperti orientasi gambar dan konversi warna dari BGR ke RGB, yang melibatkan identifikasi titik-titik penting pada setiap tangan. Dengan menggunakan *MediaPipe*, sistem mampu mendeteksi 21 landmark atau titik penting pada masing-masing tangan. Setiap titik ini diwakili oleh koordinat dua dimensi (x, y), dan kombinasi dari semua koordinat tersebut membentuk pola atau gerakan tangan yang spesifik.



**Gambar 5.** Koordinat tangan untuk deteksi gerakan

* 1. Menghitung *landmark*

Sebagai komponen kunci untuk melatih model dan memprediksi gerakan baru di masa depan, 21 landmark yang terdeteksi harus disimpan dengan format yang benar. *MediaPipe* memberikan landmark dalam bentuk objek JSON atau kamus Python dalam format tiga dimensi. Namun, untuk keperluan pelatihan model atau prediksi, diperlukan data dalam bentuk Array Landmark.

Pada proses ini dilakukan dengan iterasi pada semua landmark untuk menghitung koordinat yang akurat pada bingkai gambar utama. Koordinat tersebut kemudian diubah menjadi daftar Python sederhana dengan format [x, y], dan setiap koordinat ditambahkan ke dalam daftar lain yang berisi keseluruhan 21 koordinat tangan. Pada dibawah ini merupakan pseudocode untuk menghitung landmark.

BEGIN

# 1. Buka Kamera untuk menangkap gambar/video

Open Camera

WHILE Camera is ON:

Capture Video Frame

# 2. Ubah Frame ke format RGB

Convert Frame to RGB

# 3. Deteksi titik-titik kunci (landmark) dengan Mediapipe

landmarks = Detect Keypoints(Frame)

# 4. Periksa apakah landmark terdeteksi

IF landmarks is NOT EMPTY THEN:

Initialize empty list: landmark\_list

# 5. Iterasi melalui setiap landmark yang terdeteksi

FOR each landmark in landmarks:

x = landmark.x # Koordinat X

y = landmark.y # Koordinat Y

z = landmark.z # Koordinat Z (jika tersedia)

# 6. Simpan koordinat landmark dalam daftar

Append (x, y, z) to landmark\_list

# 7. Output hasil koordinat landmark

Print "Landmark Coordinates:", landmark\_list

ELSE:

Print "No landmarks detected."

# 8. Tutup Kamera setelah proses selesai

Close Camera

END

**Gambar 6.** Salah Satu Pseudocode untuk Menghitung Landmark

Untuk penjelasan pada pseudocode diatas sebagai berikut :

**a.** Tangkap Frame Video:

* **frame video**: adalah frame video yang ditangkap pada waktu t.

**b. Ubah Frame ke Format RGB**:

* Konversi frame ​ ke format RGB.
* Formula: =f() di mana f adalah fungsi konversi dari format frame ke RGB.

c. **Deteksi Titik Kunci:**

* Setiap frame ​ akan diproses oleh model deteksi titik kunci (seperti Mediapipe).
* Landmark yang terdeteksi adalah kumpulan titik L={(x1,y1,z1),(x2,y2,z2),…,(xn,yn,zn)} di mana (xi,yi,zi) adalah koordinat 3D (jika ada) untuk titik ke-i.

d. Periksa Apakah Landmark Terdeteksi

* Jika set L tidak kosong, lanjutkan ke langkah berikutnya.
* Formula pengecekan: L≠∅(Landmark terdeteksi)

e. Iterasi Melalui Setiap Landmark yang Terdeteksi

Untuk setiap landmark li=(xi,yi,zi) dalam set L:

* Koordinat xi, yi ​, dan zi adalah posisi titik kunci pada frame.
* Simpan koordinat ke dalam daftar:

Set landmark\_list akan berisi semua koordinat yang terdeteksi: landmark\_list={(x1,y1,z1),(x2,y2,z2),…,(xn,yn,zn)}

f. Hasil Koordinat Landmark:

* Print daftar koordinat yang terdeteksi: Print(landmark\_list)

g. Hasil akhir untuk formula yakni :

* **Frame RGB** : =
* **Landmark** : L={(x1,y1,z1),(x2,y2,z2),…,(xn,yn,zn)}
* **Penyimpanan Koordinat** : landmark\_list = L
  1. Pra-Proses Landmark

Tahap berikutnya dalam penelitian ini yaitu pra-pemrosesan, sistem mengambil salinan dalam data asli array 2D untuk digunakan, kemudian mengubah semua koordinat menjadi koordinat relative sehingga menjadi frame independen. Sehingga akan mengambil koordinat relatif dari koordinat pergelangan tangan. Setelah proses tersebut mendapatkan semua koordinat relatif, kemudian meratakan semua data menggunakan NumPy. Sehingga mengubah array 2 dimensi menjadi array satu dimensi, serta membuat semua titik data antara nilai 0-1, sebagai format yang tepat untuk melatih model. Pada dibawah ini merupakan pseudocode untuk pra-proses landmark.

BEGIN

# 1. Buka Kamera untuk menangkap gambar/video

Open Camera

WHILE Camera is ON:

Capture Video Frame

# 2. Konversi Frame ke format RGB

Convert Frame to RGB

# 3. Deteksi titik-titik kunci (landmark) menggunakan Mediapipe

landmarks = Detect Keypoints(Frame)

# 4. Periksa apakah landmark terdeteksi

IF landmarks is NOT EMPTY THEN:

Initialize empty list: processed\_landmarks

# 5. Ambil koordinat pergelangan tangan sebagai titik referensi

wrist\_x = landmarks[0].x

wrist\_y = landmarks[0].y

# 6. Iterasi melalui setiap landmark untuk menghitung koordinat relatif

FOR each landmark in landmarks:

x\_relative = landmark.x - wrist\_x

y\_relative = landmark.y - wrist\_y

# 7. Simpan koordinat relatif dalam daftar

Append (x\_relative, y\_relative) to processed\_landmarks

# 8. Normalisasi koordinat relatif (0 - 1) menggunakan Min-Max Scaling

Normalize processed\_landmarks using:

X\_norm = (X - X\_min) / (X\_max - X\_min)

Y\_norm = (Y - Y\_min) / (Y\_max - Y\_min)

# 9. Ubah data menjadi array 1D untuk input model

Flatten processed\_landmarks into a single array

# 10. Simpan hasil pra-pemrosesan untuk model Machine Learning

Save processed\_landmarks to CSV or Database

# 11. Output hasil pra-pemrosesan

Print "Processed Landmarks:", processed\_landmarks

ELSE:

Print "No landmarks detected."

# 12. Tutup Kamera setelah proses selesai

Close Camera

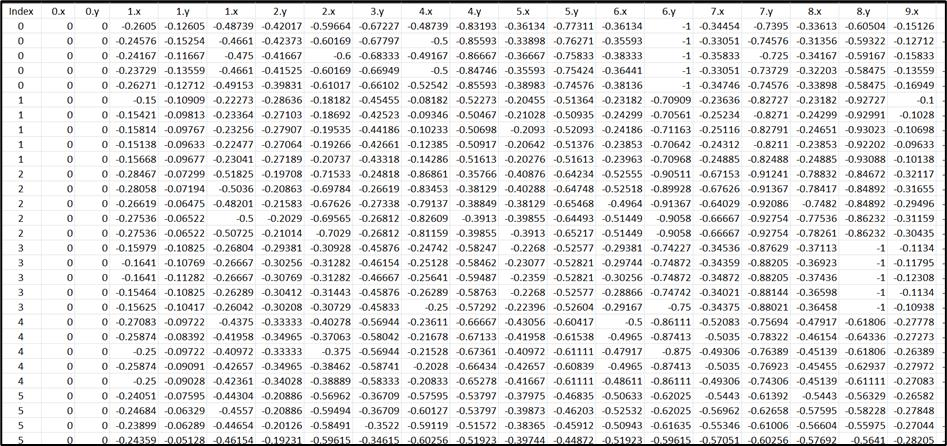
END

**Gambar 7.** Salah Satu Pseudocode untuk Pra-Proses Landmark

* 1. Pengumpulan Data

Pada langkah berikutnya, sistem yang telah menghasilkan array satu dimensi yang sudah dinormalisasi, kemudian bisa digunakan untuk mengumpulkan data dan memprediksi gerakan tangan yang baru. Selain itu juga, sistem dilengkapi dengan fitur untuk beralih antara dua mode, yaitu prediksi dan pencatatan.

Dalam proses pengumpulan data, pada penelitian ini, pada sistem yang digunakan mengaktifkan mode Logging sehingga dapat menginformasikan gerakan tangan serta proses manual pada sistem yang hasil landmark akan didapatkan melalui semua langkah pra-pemrosesan dan menyimpan dalam bentuk file CSV.

****

**Gambar 8.** Hasil dari pemrosesan data

* 1. Fase Pelatihan

Pada penelitian ini, hasil dari proses pengumpulan data digunakan untuk melatih model dengan menggunakan metode Sequential Keras Tensorflow [11] sebagai sistem yang digunakan untuk melatih Jaringan Neural. Pada dibawah ini merupakan pseudocode untuk fase pelatihan.

BEGIN

# 1. Persiapkan Data Pelatihan

Load dataset from CSV or Database

Split dataset into Features (X) and Labels (Y)

Preprocess Data:

- Normalize Features (X) if necessary

- Encode Labels (Y) if necessary

Split data into Training Set (X\_train, Y\_train) and Validation Set (X\_val, Y\_val)

# 2. Tentukan Arsitektur Model Jaringan Neural

Initialize model = Sequential()

# 3. Tambahkan Input Layer

Add Input Layer with Dense(units=units, input\_dim=input\_size, activation='activation\_function') to model

# 4. Tambahkan Hidden Layers (Layer Tersembunyi)

FOR each hidden layer in model architecture:

Add Hidden Layer with Dense(units=units, activation='activation\_function') to model

# 5. Tambahkan Output Layer

Add Output Layer with Dense(output\_size, activation='output\_activation\_function') to model

# 6. Kompilasi Model

Compile model with:

- Loss function (e.g., 'binary\_crossentropy' or 'categorical\_crossentropy')

- Optimizer (e.g., 'adam', 'sgd')

- Metrics (e.g., 'accuracy')

# 7. Pelatihan Model

Set number of epochs and batch size

Train model using:

model.fit(X\_train, Y\_train, epochs=num\_epochs, batch\_size=batch\_size, validation\_data=(X\_val, Y\_val))

# 8. Evaluasi Model

Evaluate model on test data (X\_test, Y\_test)

Get test loss and test accuracy from model evaluation

# 9. Simpan Model

Save trained model to file (e.g., 'trained\_model.h5')

# 10. Output Hasil Pelatihan

Print "Training complete. Model saved to 'trained\_model.h5'."

Print "Test Loss:", test\_loss

Print "Test Accuracy:", test\_accuracy

END

**Gambar 9.** Kode untuk pseudocode fase pelatihan

* 1. Fase Pengujian

Pada tahapan terakhir pada penelitian ini, yaitu proses menganalisa gambar dengan menggunakan sistem deteksi tangan dengan melakukan proses ulang untuk pengumpulan data yang sama dengan menggabungkan array landmark dalam kumpulan data, kemudian di proses array dari setiap frame ke Model Jaringan Syaraf Tiruan yang telah dilatih untuk memprediksi kelas yang sesuai dari tanda atau isyarat tangan yang dilakukan.

Kemudian, hasil dari array tersebut dikirim ke Model Jaringan Syaraf Tiruan atau Otak Proyek, dengan membandingkan data yang diberikan sebelumnya. Jika hasil dari proses tersebut ditemukan kecocokan dengan kelas alfabet, maka indeks kelas alfabet yang terdeteksi, sehingga akan memudahkan untuk menemukan alfabet. Setelah kita mendapatkan indeks kelas, kita dapat dengan mudah menemukan alfabetnya.

* 1. Hasil

Hasil dari sistem deteksi gerakan tubuh berbasis *Machine* *Learning* menggunakan *Mediapipe* menunjukkan bahwa sistem mampu mengenali dan mengklasifikasikan berbagai jenis gerakan dengan tingkat akurasi yang tinggi. Proses deteksi dimulai dengan pengambilan gambar atau video melalui kamera, di mana sistem kemudian mengidentifikasi titik-titik kunci (landmark) pada tubuh pengguna. Titik-titik ini dianalisis untuk menentukan pola gerakan yang dilakukan. Setelah data landmark diperoleh, sistem melakukan pra-pemrosesan dengan normalisasi koordinat agar skala data lebih seragam. Model *Machine Learning* yang diterapkan, seperti *Support Vector Machine* (SVM) atau *Neural Network*, kemudian memproses data ini untuk mengklasifikasikan gerakan ke dalam kategori yang telah ditentukan, seperti berjalan, melompat, atau mengangkat tangan. Hasil deteksi ditampilkan dalam bentuk teks pada layar atau dapat dikonversi menjadi perintah suara untuk meningkatkan interaksi pengguna. Berdasarkan evaluasi performa menggunakan metrik akurasi, presisi, *recall*, dan *F1*-*score*, sistem ini terbukti efisien dan dapat digunakan dalam berbagai aplikasi, termasuk bidang kesehatan, olahraga, dan interaksi manusia-komputer. Dibawah ini ialah hasil untuk pseudocode untuk deteksi gerakan tubuh dengan mediapipe.

BEGIN

# 1. Buka Kamera untuk menangkap gambar/video

Open Camera

WHILE Camera is ON:

Capture Video Frame

# 2. Ubah Frame ke format RGB

Convert Frame to RGB

# 3. Deteksi titik-titik kunci tubuh menggunakan Mediapipe

landmarks = Detect Body Keypoints(Frame)

# 4. Periksa apakah landmark terdeteksi

IF landmarks is NOT EMPTY THEN:

Initialize empty list: body\_landmarks\_list

# 5. Iterasi melalui setiap landmark yang terdeteksi

FOR each landmark in landmarks:

x = landmark.x # Koordinat X

y = landmark.y # Koordinat Y

z = landmark.z # Koordinat Z (jika tersedia)

# 6. Simpan koordinat landmark dalam daftar

Append (x, y, z) to body\_landmarks\_list

# 7. Normalisasi koordinat untuk memastikan skala seragam

Normalize body\_landmarks\_list

# 8. Klasifikasi gerakan tubuh berdasarkan data landmark

predicted\_action = ClassifyMovement(body\_landmarks\_list)

# 9. Tampilkan hasil deteksi

Print "Detected Movement:", predicted\_action

ELSE:

Print "No body movement detected."

# 10. Tutup Kamera setelah proses selesai

Close Camera

END

**Gambar 10.** Pseudocode mendeteksi gerak tubuh dengan Mediapipe

Pada penjelasan pseudocode di atas pada langkah pertama dan kedua merupakan Kamera menangkap video dan mengonversi formatnya ke RGB, dan langkah ketiga Sistem mendeteksi titik-titik kunci tubuh menggunakan Mediapipe. Dan langkah keempat sampai tahap ke enam untuk titik kunci terdeteksi, sistem akan menyimpan koordinatnya. Dan langkah ketujuh data landmark dinormalisasi agar dapat digunakan dalam model klasifikasi. Langkah kedelapan *Model Machine Learning* memprediksi gerakan berdasarkan pola titik-titik kunci. Dan langkah kesembilan pada hasil deteksi ditampilkan, misalnya "*Jumping*", "*Walking*", atau "*Running*". Dan langkah terakhir kamera ditutup setelah proses selesai.

**Kesimpulan**

Sistem deteksi bahasa isyarat memiliki potensi besar untuk merevolusi komunikasi dan mendorong inklusivitas bagi pengguna bahasa isyarat. Teknologi ini dapat menjembatani kesenjangan antara bahasa isyarat dan bahasa lisan/tulisan, memungkinkan komunikasi yang lebih lancar dalam berbagai situasi. Dengan mengatasi hambatan komunikasi, sistem ini memberdayakan pengguna bahasa isyarat di berbagai sektor, seperti pendidikan, pekerjaan, dan layanan kesehatan, sehingga menciptakan masyarakat yang lebih inklusif dan setara.

Selain itu, MediaPipe dengan modul Hand Pose Estimation-nya, menyediakan kemampuan untuk mengidentifikasi titik-titik kunci (landmark) pada tangan dengan akurasi tinggi. Data landmark ini kemudian menjadi input penting bagi model machine learning yang bertugas menerjemahkan bahasa isyarat. Penggunaan MediaPipe memberikan keuntungan signifikan dalam penelitian ini, termasuk kemudahan integrasi, kinerja real-time yang optimal, dan ketersediaan model pre-trained yang mengurangi kebutuhan sumber daya komputasi.

Serta, penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam pengembangan sistem deteksi bahasa isyarat yang inklusif dan efektif. Meskipun masih terdapat batasan, seperti kemampuan sistem dalam mengenali satu huruf untuk setiap input, penelitian ini menjadi landasan penting untuk pengembangan sistem yang lebih canggih di masa depan. Pemanfaatan MediaPipe dalam penelitian ini juga menunjukkan potensi besar teknologi ini dalam mendukung pengembangan aplikasi-aplikasi berbasis visi komputer, khususnya dalam konteks penerjemahan bahasa isyarat.

Sehingga, pengembangan sistem deteksi bahasa isyarat yang berkelanjutan, melalui riset dan kolaborasi, sangat penting untuk mengoptimalkan potensinya. Solusi ini menjadi fondasi bagi pengembangan antarmuka pengguna berbasis pengenalan bahasa isyarat, yang dapat diadaptasi untuk melatih isyarat-isyarat baru di masa mendatang. Inovasi dan upaya yang konsisten akan membuka jalan bagi komunikasi yang inklusif dan efektif di era modern.

**Referensi**

[1] I. B. A. Peling, I. M. P. A. Ariawan, and G. B. Subiksa, “Deteksi Bahasa Isyarat Menggunakan Tensorflow Lite dan American Sign Language (ASL),” *J. Krisnadana*, vol. 3, no. 2, pp. 90–100, 2024, doi: 10.58982/krisnadana.v3i2.534.

[2] D. A. Primadiansyah *et al.*, “Literature Review : Implementasi Machine Learning Dalam Pengenalan Bahasa Isyarat Indonesia ( BISINDO ),” vol. 2, no. 5, pp. 821–824, 2024.

[3] N. Alexander, R. B. Widodo, and W. Swastika, “Penggunaan Machine Learning Dalam Klasifikasi Bahasa Isyarat BISINDO Menggunakan Kamera,” *Pros. Semin. Nas. Inform. Sist. Inf.*, vol. 3, no. 1, pp. 11–26, 2023.

[4] Agus Nugroho, “Deteksi Bahasa Isyarat Bisindo Menggunakan Metode Machine Learning,” *J. Process.*, vol. 18, no. 2, pp. 152–158, 2023, doi: 10.33998/processor.2023.18.2.1380.

[5] F. Rachardi, “Deteksi Gambar Gestur Kosakata Bahasa Isyarat Indonesia dengan Convolutional Neural Network,” *Institutional Repos. UIN Syarif Hidayatullah*, p. 192, 2020.

[6] S. N. Budiman, S. Lestanti, H. Yuana, and B. N. Awwalin, “SIBI (Sistem Bahasa Isyarat Indonesia) berbasis Machine Learning dan Computer Vision untuk Membantu Komunikasi Tuna Rungu dan Tuna Wicara,” *J. Teknol. dan Manaj. Inform.*, vol. 9, no. 2, pp. 119–128, 2023, doi: 10.26905/jtmi.v9i2.10993.

[7] I. Suyudi, S. Sudadio, and S. Suherman, “Pengenalan Bahasa Isyarat Indonesia menggunakan Mediapipe dengan Model Random Forest dan Multinomial Logistic Regression,” *J. Ilmu Siber dan Teknol. Digit.*, vol. 1, no. 1, pp. 65–80, 2023, doi: 10.35912/jisted.v1i1.1899.

[8] A. R. Ardiansyah, A. H. Nur’azizan, and R. Fernandis, “Implementasi Deteksi Bahasa Isyarat Tangan Menggunakan OpenCV dan MediaPipe,” *Stain. (Seminar Nas. Teknol. Sains)*, vol. 3, no. 1, pp. 331–337, 2024.

[9] N. Anam, “Sistem Deteksi Simbol Pada Sibi (Sistem Isyarat Bahasa Indonesia) Menggunakan Mediapipe Dan Resnet-50,” 2022.

[10] R. Kumar, A. Bajpai, and A. Sinha, “Mediapipe and CNNs for Real-Time ASL Gesture Recognition.” 2023, doi: 10.48550/arXiv.2305.05296.

[11] A. H. Gustsa and G. S. Permadi, “Sistem Deteksi Bahasa Isyarat Secara Realtime Dengan Tensorflow Object Detection dan Python Menggunakan Metode Convolutional Neural Network,” *Inov. J. Ilm. Inov. …*, vol. 7, no. 2, pp. 1–10, 2023.

|  |  |
| --- | --- |
|  | **Irfan Nurdiansyah S.Kom., M.Kom.**  Universitas Dian Nusantara  Fakultas Ilmu Komputer  Jl.Tanjung Duren Barat 2 No.1 |
|  | **Reni Utami S.SI., M.Kom.**  Universitas Dian Nusantara  Fakultas Ilmu Komputer  Jl.Tanjung Duren Barat 2 No.1 |

|  |  |
| --- | --- |
|  | **Muchamad Sandy S.Kom., M.M.SI**  Universitas Dian Nusantara  Fakultas Ilmu Komputer  Jl.Tanjung Duren Barat 2 No.1 |