

LAPORAN AKHIR HASIL ANALISIS PROJECT KECERDASAN BUATAN

“Komputer Vision : Penerjemah Bahasa Isyarat”



DISUSUN OLEH :

- 1. LANANG FAJRI RAMADHAN (2300018264)**
- 2. REYNARD RAKHA FATIN NUGRAHA (2300018234)**
- 3. SEPTIA NENENSIH (2300018239)**

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS AHMAD DAHLAN YOGYAKARTA**

2025

KATA PENGANTAR

Puji syukur kami panjatkan kehadiran Allah SWT, karena atas rahmat, karunia, serta hidayah-Nya, kami dapat menyelesaikan laporan akhir proyek mata kuliah Kecerdasan Buatan yang berjudul “Komputer Vision: Penerjemah Bahasa Isyarat” dengan baik dan tepat waktu. Laporan ini disusun sebagai pemenuhan tugas akhir, dengan harapan dapat memberikan manfaat serta wawasan baru, baik bagi tim penyusun maupun bagi para pembaca.

Tak lupa pula, selawat serta salam senantiasa kami haturkan kepada junjungan kita, Nabi Muhammad SAW, yang telah menjadi suri teladan terbaik bagi seluruh umat manusia. Semoga kita semua kelak mendapatkan syafaatnya di Yaumul Kiyamah nanti. Aamiin ya Rabbal 'alamin.

Dalam penyusunan laporan ini, kami menyadari bahwa keberhasilannya tidak terlepas dari bimbingan, arahan, dan dukungan dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati, kami ingin mengucapkan terima kasih yang sebesar-besarnya kepada dosen pengampu mata kuliah Kecerdasan Buatan, yang telah dengan sabar memberikan bimbingan, ilmu, serta motivasi yang sangat berharga selama proses pengerjaan proyek ini. Ucapan terima kasih juga kami sampaikan kepada seluruh rekan tim yang telah bekerja sama dengan solid dan memberikan kontribusi terbaiknya.

Kami menyadari sepenuhnya bahwa laporan ini masih memiliki banyak kekurangan dan jauh dari kata sempurna. Oleh karena itu, kami sangat terbuka dan mengharapkan kritik serta saran yang bersifat membangun dari para pembaca demi perbaikan di masa yang akan datang.

Akhir kata, Semoga laporan ini dapat diterima dengan baik dan memberikan kontribusi positif bagi pengembangan ilmu pengetahuan.

Yogyakarta, 12 Juli 2025

Tim Penyusun

A. Latar Belakang

Komunikasi merupakan hak fundamental bagi setiap individu dan menjadi pilar utama dalam interaksi sosial. Di tengah keberagaman cara berkomunikasi, bahasa isyarat memegang peranan krusial sebagai media komunikasi utama bagi komunitas Teman Tuli. Namun, kesenjangan komunikasi yang signifikan masih menjadi tantangan besar, karena mayoritas masyarakat awam belum memahami bahasa isyarat. Keterbatasan ini menciptakan hambatan dalam interaksi sehari-hari dan berpotensi membatasi inklusivitas sosial bagi Teman Tuli.

Hambatan komunikasi ini berdampak luas pada berbagai aspek kehidupan, mulai dari akses terhadap layanan publik seperti kesehatan dan pendidikan, hingga kesempatan kerja dan partisipasi penuh dalam kegiatan kemasyarakatan. Proses pembelajaran bahasa isyarat secara konvensional seringkali membutuhkan waktu, dedikasi, dan sumber daya yang tidak dimiliki oleh semua orang, sehingga jurang komunikasi ini sulit untuk dijembatani secara efektif dan cepat.

Untuk mengatasi permasalahan tersebut, inovasi teknologi berbasis kecerdasan buatan (AI) menawarkan solusi yang menjanjikan. Oleh karena itu, kami mengusulkan pengembangan aplikasi “Penerjemah Bahasa Isyarat” yang mampu mendeteksi dan menerjemahkan abjad jari dalam Bahasa Isyarat Indonesia (BISINDO) atau Sistem Bahasa Isyarat Indonesia (SIBI) secara real-time. Dengan memanfaatkan teknologi computer vision, aplikasi ini dapat mengenali gestur jari melalui kamera perangkat dan langsung menampilkannya dalam bentuk teks.

Aplikasi ini dirancang dengan dua tujuan utama. Pertama, sebagai jembatan komunikasi instan yang memungkinkan masyarakat awam untuk memahami pesan yang disampaikan melalui isyarat jari tanpa harus menghafalkannya terlebih dahulu. Kedua, sebagai alat bantu belajar yang interaktif dan mudah diakses, sehingga mendorong lebih banyak orang untuk mempelajari bahasa isyarat secara mandiri. Dengan demikian, pengembangan aplikasi ini tidak hanya bertujuan untuk menyediakan solusi teknologi, tetapi juga untuk membangun masyarakat yang lebih inklusif, empatik, dan setara bagi semua.

B. Tujuan/Manfaat

1. Manfaat Umum :

a) Menjembatani Kesenjangan Komunikasi

Aplikasi ini berfungsi sebagai jembatan *real-time* yang memecah hambatan bahasa antara komunitas Tuli dan masyarakat awam, sehingga interaksi di berbagai situasi seperti di tempat umum atau layanan publik menjadi lebih lancar.

b) Melestarikan dan Mempromosikan Bahasa Isyarat

Dengan menjadikan bahasa isyarat lebih mudah diakses dan dipahami, aplikasi ini secara tidak langsung ikut melestarikan serta mendorong lebih banyak orang untuk tertarik mempelajari bahasa isyarat secara lebih mendalam.

2. Bagi Pengguna Awam (Masyarakat Umum) :

- a) Kemudahan Berinteraksi Secara Langsung: Memungkinkan pengguna awam untuk dapat memahami dan berkomunikasi dasar dengan Teman Tuli secara spontan, tanpa harus mengikuti kursus atau menghafal isyarat terlebih dahulu.
- b) Sarana Pembelajaran yang Praktis: Menjadi alat bantu belajar yang interaktif untuk mengenal abjad jari dan dasar-dasar bahasa isyarat, di mana pengguna bisa langsung mempraktikkannya.

3. Bagi Komunitas Tuli (Pengguna Bahasa Isyarat)

- a) Meningkatkan Kemandirian dan Aksesibilitas: Memberikan kemandirian lebih bagi Teman Tuli untuk berkomunikasi di berbagai tempat, seperti di toko, kantor pemerintahan, atau fasilitas kesehatan, tanpa harus selalu bergantung pada kehadiran juru bahasa isyarat.
- b) Memperluas Lingkup Interaksi Sosial: Memudahkan Teman Tuli untuk berinteraksi dengan lebih banyak orang, sehingga dapat mengurangi rasa terisolasi dan membangun hubungan sosial yang lebih luas.

4. Bagi Pembuat Aplikasi

- a) Mengaplikasikan dan Mengembangkan Kompetensi: Menjadi proyek nyata untuk menerapkan dan memperdalam keahlian di bidang teknologi canggih seperti Kecerdasan Buatan (*Artificial Intelligence*), Pengolahan Citra (*Computer Vision*), dan Pembelajaran Mesin (*Machine Learning*).
- b) Memberikan Kontribusi Sosial yang Bermakna: Memberikan kesempatan untuk menciptakan sebuah karya teknologi yang memiliki dampak sosial positif, yaitu dengan membantu meningkatkan inklusivitas dan kesetaraan dalam masyarakat.

C. Tools dan library yang digunakan

Dalam penyusunan project bahasa isyarat ini, ada beberapa tools dan library yang digunakan di antara nya :

- a) Tools

- Bahasa Pemrograman Python: Ini adalah alat utama berupa bahasa yang digunakan untuk menulis instruksi dan logika dari keseluruhan program.
- Visual Studio Code: Ini adalah *Integrated Development Environment* (IDE) atau editor teks, yaitu perangkat lunak yang menjadi tempat untuk menulis, mengelola, dan menjalankan kode Python.

b) Library (Pustaka)

- OpenCV: Pustaka khusus untuk kebutuhan computer vision, seperti mengakses kamera, memproses gambar, dan menampilkan video secara *real-time*.
- MediaPipe: Pustaka yang menyediakan model siap pakai untuk mendeteksi dan melacak titik-titik penting (landmarks) pada tangan, wajah, dan tubuh.
- Scikit-learn (sklearn): Pustaka yang berisi kumpulan alat dan algoritma untuk tugas-tugas machine learning, seperti melatih model klasifikasi.
- Matplotlib: Pustaka yang digunakan untuk membuat visualisasi data, seperti grafik atau plot, untuk menganalisis hasil model.

D. Cara Kerja aplikasi

Pengembangan aplikasi penerjemah bahasa isyarat ini melalui empat tahap utama yang saling berkesinambungan, mulai dari pengumpulan data hingga implementasi.

a) Pengumpulan Data (collect_img.py)

Tahap pertama adalah pengumpulan dataset mentah. Dengan menjalankan skrip ini, kami mengambil sampel gambar untuk setiap kelas abjad isyarat (misalnya, 100 gambar untuk gestur 'A', 100 untuk 'B', dan seterusnya). Kumpulan gambar ini menjadi fondasi utama yang akan digunakan untuk melatih model AI.

b) Pra-pemrosesan dan Pembuatan Dataset (create_dataset.py)

Setelah data gambar terkumpul, skrip ini dijalankan untuk mengolahnya. Setiap gambar diproses untuk mengekstraksi fitur-fitur penting (seperti titik-titik landmark tangan) dan diubah ke dalam format numerik yang seragam. Hasilnya kemudian disimpan dalam satu file, data.pickle, agar mudah dan cepat diakses pada tahap selanjutnya.

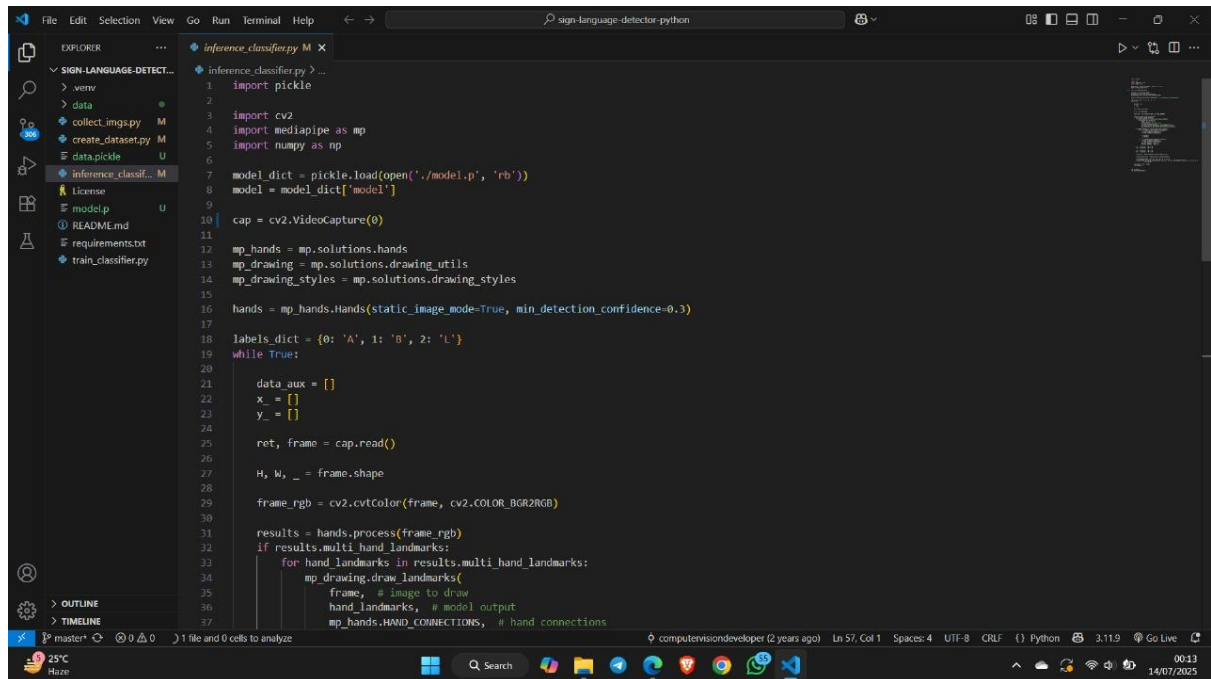
c) Pelatihan Model Klasifikasi (train_classifier.py)

Pada tahap ini, dataset yang berada di file data.pickle digunakan untuk melatih sebuah model machine learning. Proses ini mengajarkan AI untuk mengenali dan membedakan pola unik dari setiap kelas abjad, sehingga model dapat mengklasifikasikan gestur 'A', 'B', 'C', dan seterusnya. Model cerdas yang telah berhasil dilatih ini kemudian disimpan dalam file model.p.

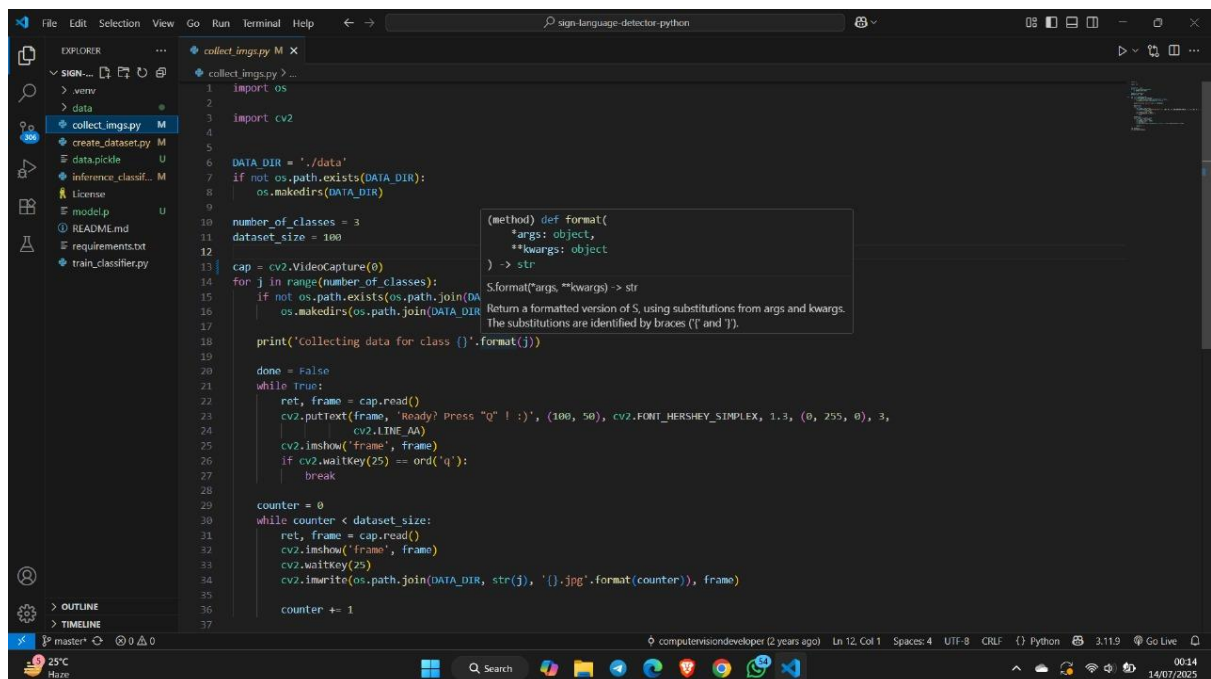
d) Implementasi dan Pengujian Antarmuka (interface_classifier.py)

Ini adalah tahap akhir di mana fungsionalitas aplikasi diuji. Skrip ini akan memuat model yang telah dilatih (model.p) dan menggunakannya untuk membuat prediksi pada input video dari kamera secara real-time. Hasil prediksi (misalnya, huruf 'L') kemudian ditampilkan di layar, membuktikan bahwa AI dapat menerjemahkan gestur tangan secara langsung.

E. Cuplikan Kode Program pembuatan Aplikasi



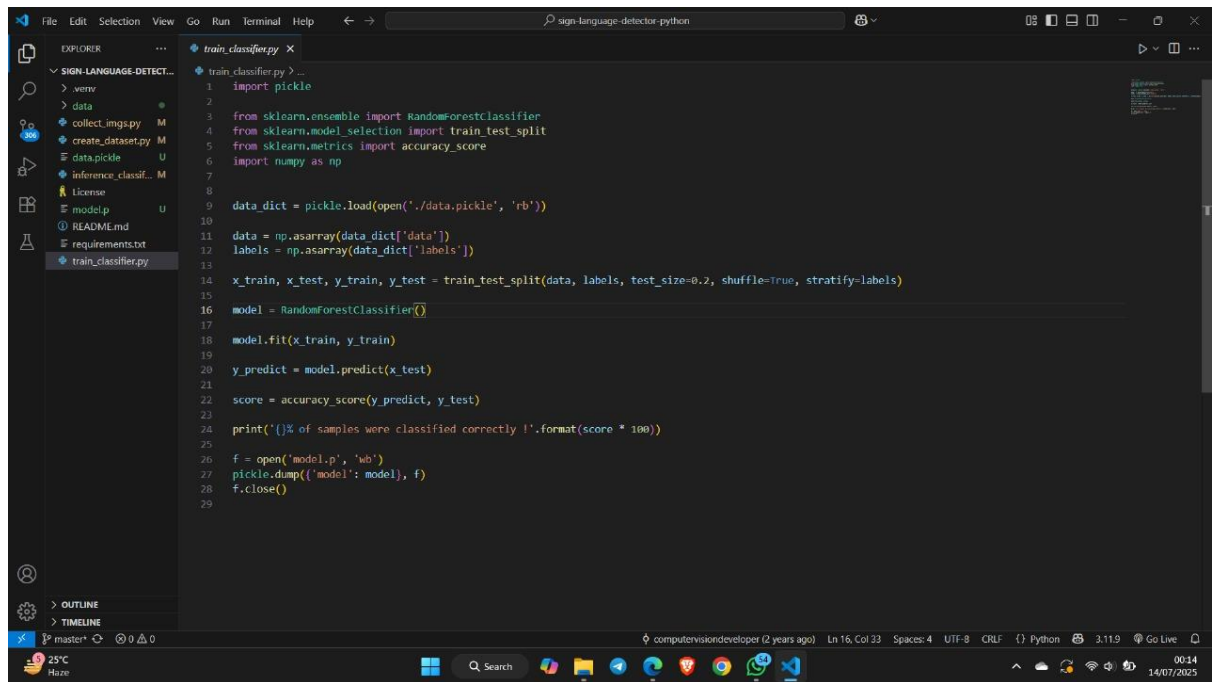
```
1 import pickle
2
3 import cv2
4 import mediapipe as mp
5 import numpy as np
6
7 model_dict = pickle.load(open('./model.p', 'rb'))
8 model = model_dict['model']
9
10 cap = cv2.VideoCapture(0)
11
12 mp_hands = mp.solutions.hands
13 mp_drawing = mp.solutions.drawing_utils
14 mp_drawing_styles = mp.solutions.drawing_styles
15
16 hands = mp_hands.Hands(static_image_mode=True, min_detection_confidence=0.3)
17
18 labels_dict = {0: 'A', 1: 'B', 2: 'L'}
19 while True:
20
21     data aux = []
22     x_ = []
23     y_ = []
24
25     ret, frame = cap.read()
26
27     H, W, _ = frame.shape
28
29     frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
30
31     results = hands.process(frame_rgb)
32     if results.multi_hand_landmarks:
33         for hand_landmarks in results.multi_hand_landmarks:
34             mp_drawing.draw_landmarks(
35                 frame, # image to draw
36                 hand_landmarks, # model output
37                 mp_hands.HAND_CONNECTIONS, # hand connections
```



```
1 import os
2
3 import cv2
4
5 DATA_DIR = './data'
6 if not os.path.exists(DATA_DIR):
7     os.makedirs(DATA_DIR)
8
9 number_of_classes = 3
10 dataset_size = 100
11
12 cap = cv2.VideoCapture(0)
13 for j in range(number_of_classes):
14     if not os.path.exists(os.path.join(DATA_DIR, str(j))):
15         os.makedirs(os.path.join(DATA_DIR, str(j)))
16     print('collecting data for class {}'.format(j))
17
18     done = False
19     while True:
20
21         ret, frame = cap.read()
22         cv2.putText(frame, 'Ready? Press "q" ! :)', (100, 50), cv2.FONT_HERSHEY_SIMPLEX, 1.3, (0, 255, 0), 3,
23                     cv2.LINE_AA)
24         cv2.imshow('frame', frame)
25         if cv2.waitKey(25) == ord('q'):
26             break
27
28     counter = 0
29     while counter < dataset_size:
30         ret, frame = cap.read()
31         cv2.imshow('frame', frame)
32         cv2.waitKey(25)
33         cv2.imwrite(os.path.join(DATA_DIR, str(j), '{}.jpg'.format(counter)), frame)
34         counter += 1
```

```
1 import os
2 import pickle
3
4 import mediapipe as mp
5 import cv2
6 import matplotlib.pyplot as plt
7
8
9 mp_hands = mp.solutions.hands
10 mp_drawing = mp.solutions.drawing_utils
11 mp_drawing_styles = mp.solutions.drawing_styles
12
13 hands = mp_hands.Hands(static_image_mode=True, min_detection_confidence=0.3)
14
15 DATA_DIR = './data'
16
17 data = []
18 labels = []
19 for dir_ in os.listdir(DATA_DIR):
20     for img_path in os.listdir(os.path.join(DATA_DIR, dir_)):
21         data_aux = []
22
23         x_ = []
24         y_ = []
25
26         img = cv2.imread(os.path.join(DATA_DIR, dir_, img_path))
27         img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
28
29         results = hands.process(img_rgb)
30         if results.multi_hand_landmarks:
31             for hand_landmarks in results.multi_hand_landmarks:
32                 for i in range(len(hand_landmarks.landmark)):
33                     x = hand_landmarks.landmark[i].x
34                     y = hand_landmarks.landmark[i].y
35
36                     x_.append(x)
37                     y_.append(y)
```

```
1 import os
2 import pickle
3
4 import mediapipe as mp
5 import cv2
6 import matplotlib.pyplot as plt
7
8
9 mp_hands = mp.solutions.hands
10 mp_drawing = mp.solutions.drawing_utils
11 mp_drawing_styles = mp.solutions.drawing_styles
12
13 hands = mp_hands.Hands(static_image_mode=True, min_detection_confidence=0.3)
14
15 DATA_DIR = './data'
16
17 data = []
18 labels = []
19 for dir_ in os.listdir(DATA_DIR):
20     for img_path in os.listdir(os.path.join(DATA_DIR, dir_)):
21         data_aux = []
22
23         x_ = []
24         y_ = []
25
26         img = cv2.imread(os.path.join(DATA_DIR, dir_, img_path))
27         img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
28
29         results = hands.process(img_rgb)
30         if results.multi_hand_landmarks:
31             for hand_landmarks in results.multi_hand_landmarks:
32                 for i in range(len(hand_landmarks.landmark)):
33                     x = hand_landmarks.landmark[i].x
34                     y = hand_landmarks.landmark[i].y
35
36                     x_.append(x)
37                     y_.append(y)
```

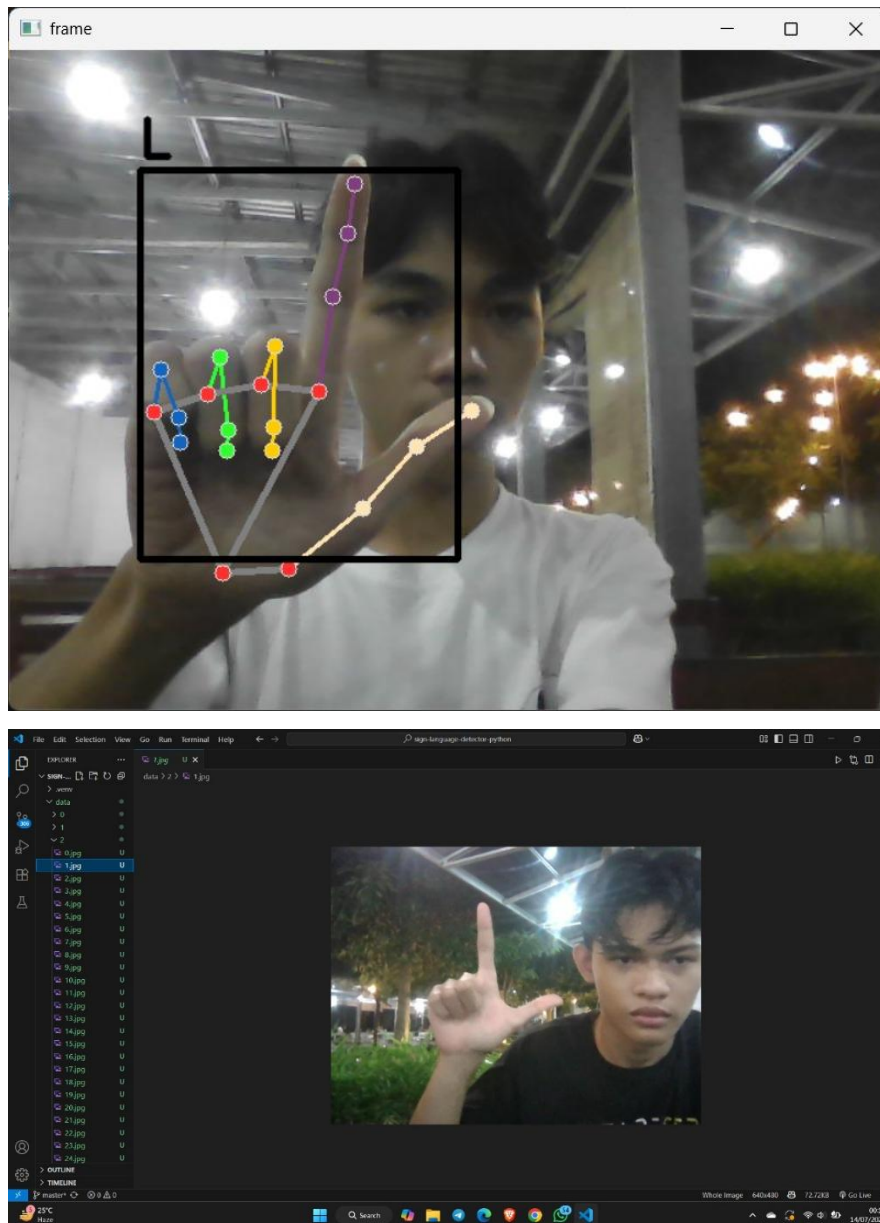


```
1 import pickle
2
3 from sklearn.ensemble import RandomForestClassifier
4 from sklearn.model_selection import train_test_split
5 from sklearn.metrics import accuracy_score
6 import numpy as np
7
8 data_dict = pickle.load(open('./data.pickle', 'rb'))
9
10 data = np.asarray(data_dict['data'])
11 labels = np.asarray(data_dict['labels'])
12
13 x_train, x_test, y_train, y_test = train_test_split(data, labels, test_size=0.2, shuffle=True, stratify=labels)
14
15 model = RandomForestClassifier()
16
17 model.fit(x_train, y_train)
18
19 y_predict = model.predict(x_test)
20
21 score = accuracy_score(y_predict, y_test)
22
23 print('{}% of samples were classified correctly !'.format(score * 100))
24
25 f = open('model.p', 'wb')
26 pickle.dump({'model': model}, f)
27 f.close()
28
29
```

Secara keseluruhan, alur kerja ini dimulai dengan pengumpulan data gambar mentah untuk setiap abjad isyarat menggunakan skrip `collect_imgs.py`, yang kemudian diolah oleh `create_dataset.py` untuk mengekstraksi fitur-fitur numerik berupa landmark tangan menggunakan MediaPipe dan menyimpannya ke dalam file dataset terstruktur `data.pickle`. Dataset ini selanjutnya digunakan oleh skrip `train_classifier.py` untuk melatih model klasifikasi dari pustaka Scikit-learn, di mana model yang telah cerdas ini disimpan sebagai file `model.p`. Pada tahap akhir, skrip `inference_classifier.py` memuat `model.p` tersebut untuk melakukan penerjemahan gestur tangan secara real-time dari input kamera, membuktikan keberhasilan fungsionalitas aplikasi dari awal hingga akhir.

F. Hasil

Aplikasi yang dikembangkan berhasil mencapai tujuannya untuk mendeteksi dan menerjemahkan gestur abjad jari secara *real-time*. Ketika pengguna menunjukkan sebuah gestur tangan di depan kamera, sistem akan bekerja melalui serangkaian proses yang telah dijelaskan.



Seperti yang terlihat pada gambar di atas, sistem berhasil mendeteksi keberadaan tangan dan menandainya dengan kotak pembatas. Selanjutnya, sistem menganalisis bentuk gestur dengan memetakan titik-titik digital (landmarks) pada setiap sendi jari. Berdasarkan pola unik dari posisi landmark tersebut, model AI yang telah dilatih mengklasifikasikan gestur tersebut dan mencocokkannya dengan abjad yang telah dipelajari. Pada contoh ini, pola gestur dikenali sebagai huruf "L", yang kemudian ditampilkan di layar sebagai hasil akhir, memberikan umpan balik instan kepada pengguna.

G. Kesimpulan

Proyek pengembangan aplikasi “Penerjemah Bahasa Isyarat” ini telah berhasil diwujudkan menjadi sebuah prototipe fungsional yang mampu menerjemahkan gestur abjad jari secara real-time. Dengan memanfaatkan bahasa pemrograman Python dan

mengintegrasikan library canggih seperti OpenCV untuk interaksi kamera, MediaPipe untuk deteksi gestur tangan, dan Scikit-learn untuk melatih model klasifikasi, alur kerja proyek dari pengumpulan data hingga implementasi dapat berjalan dengan baik. Aplikasi ini menjadi bukti nyata bahwa inovasi teknologi AI, khususnya dalam bidang computer vision, dapat secara efektif menjembatani kesenjangan komunikasi serta menyediakan sarana pembelajaran praktis untuk mendorong terciptanya masyarakat yang lebih inklusif dan setara.

REFERENSI

- Algoritma. (2022). *7 Teknik Data Mining yang Paling Mendasar dan Penting*. Algorit.ma.
- Al-Barry, M. D. (n.d.). *Kamus Ilmiah Populer*. Arkola.
- Artificial Intelligence Center Indonesia. (2024). *Algoritma Machine Learning: Jenis dan Contoh*. Aici-umg.com.
- Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.
- Deriota. (2023). *Sistem Pendukung Keputusan: Pengertian, Cara Kerja dan Contohnya*.
- FTIK Teknokrat. (2023). *Bagaimana AI Dapat Membantu dalam Pengambilan Keputusan Bisnis yang Lebih Baik?*
- Google. (n.d.). *MediaPipe*. Google Open Source. Diakses 14 Juli 2025.
- Inixindo. (2023). *8 Contoh dan Manfaat Machine Learning yang membuat dunia lebih baik*.
- Jurnal Cendekia. (2024). *Implementasi Konsep Information Retrieval dengan Metode Case Insensitive Search pada Mesin Pencari Dokumen Quality*.
- Majelis Ulama Indonesia. (n.d.). *Fatwa-Fatwa tentang Teknologi Informasi dan Komunikasi*.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Oct), 2825-2830.
- Python Software Foundation. (n.d.). *Python Language Reference*. Diakses 14 Juli 2025, dari <https://www.python.org>
- Qardhawi, Y. (n.d.). *Fiqh Maqasid al-Shari'ah*. Pustaka Al-Kautsar.
- Repository UIN Suska. (2013). *Sistem Temu Kembali Informasi dengan Menerapkan Metode Probabilistik Binary Independence Model (BIM)*.

STEKOM. (n.d.). *Data Mining: Pengertian, Proses, Manfaat, Dan 3 Contoh Penerapannya*. Diakses 14 Juli 2025.

Universitas Mahakarya Asia. (n.d.). *Penggunaan Robotika dalam Industri dan Kehidupan Sehari-hari: Meningkatkan Efisiensi dan Kualitas Hidup*. Blog UNMAHA. Diakses 14 Juli 2025.

Widya Robotics. (2024). *Computer Vision dalam Dunia Medis: Inovasi dan Manfaatnya*.