

LAPORAN SEMENTARA

Nama : Dwi Alfianto

Kelas : TKA-6D

NIM : 234308096

A. PENDAHULUAN

1. Latar Belakang dan Dasar Teori

Perkembangan teknologi **computer vision** dan **kecerdasan buatan (Artificial Intelligence)** memungkinkan komputer untuk mengenali serta memahami objek dan gerakan manusia secara visual. Salah satu penerapan yang banyak dikembangkan adalah **pengenalan tangan (hand tracking)**, yang dapat digunakan dalam berbagai bidang seperti *human-computer interaction*, sistem kendali tanpa sentuhan, augmented reality, serta aplikasi edukasi dan kesehatan.

Pada praktikum ini digunakan **MediaPipe Hand**, yaitu sebuah *framework* yang dikembangkan oleh Google untuk mendeteksi dan melacak tangan secara **real-time**. MediaPipe Hand mampu mengenali posisi tangan serta menentukan titik-titik penting (*landmarks*) pada tangan manusia hanya dengan menggunakan kamera, tanpa memerlukan perangkat tambahan seperti sensor khusus.

2. Computer Vision

Computer vision adalah cabang ilmu komputer yang mempelajari bagaimana komputer dapat memperoleh, memproses, dan memahami informasi dari citra atau video digital. Dalam konteks praktikum ini, computer vision digunakan untuk:

- Mengambil citra tangan dari kamera (webcam),
- Memproses citra tersebut,
- Mendeteksi bentuk dan posisi tangan secara otomatis.

Teknologi ini umumnya melibatkan proses seperti *image acquisition*, *preprocessing*, *feature extraction*, dan *object detection*.

3. MediaPipe Hand

MediaPipe Hand merupakan solusi *hand tracking* berbasis *machine learning* yang mampu mendeteksi satu atau dua tangan dalam sebuah frame video. MediaPipe Hand bekerja melalui dua tahap utama:

1. **Palm Detection** – mendeteksi area telapak tangan dalam citra.
2. **Hand Landmark Detection** – menentukan 21 titik landmark pada tangan, meliputi ujung jari, ruas jari, dan pergelangan tangan.

Setiap landmark direpresentasikan dalam koordinat **x, y, dan z**, yang memungkinkan analisis posisi tangan secara lebih akurat, termasuk kedalaman relatif.

4. Hand Landmark

Dalam MediaPipe Hand, tangan direpresentasikan oleh **21 titik landmark**, di mana setiap titik memiliki fungsi tertentu, seperti pergelangan tangan, sendi jari, ujung jari (thumb, index, middle, ring, dan pinky). Landmark ini menjadi dasar untuk mendeteksi gerakan tangan, menghitung sudut atau jarak antar jari, mengembangkan sistem pengenalan gesture.

5. Python sebagai Bahasa Pemrograman

Python merupakan bahasa pemrograman tingkat tinggi yang banyak digunakan dalam bidang data science, machine learning, dan computer vision. Keunggulan Python dalam praktikum ini antara lain:

- Sintaks yang sederhana dan mudah dipahami,
- Dukungan pustaka yang luas,
- Kompatibilitas tinggi dengan MediaPipe dan OpenCV.

Dalam praktikum ini, Python digunakan untuk mengimplementasikan MediaPipe Hand serta mengolah data hasil deteksi tangan.

6. OpenCV

OpenCV (Open Source Computer Vision Library) adalah pustaka yang digunakan untuk pengolahan citra dan video. OpenCV berperan dalam:

- Mengakses kamera,
- Menampilkan hasil deteksi tangan,
- Menggambar landmark dan garis koneksi pada tangan.

OpenCV bekerja sebagai penghubung antara kamera dan MediaPipe dalam sistem yang dibangun.

7. PyCharm sebagai Lingkungan Pengembangan

PyCharm adalah *Integrated Development Environment (IDE)* yang digunakan untuk menulis, menjalankan, dan melakukan *debugging* program Python. Penggunaan PyCharm mempermudah praktikum karena menyediakan:

- Penulisan kode yang terstruktur,
- Deteksi kesalahan sintaks,
- Manajemen pustaka dan *virtual environment*.

B. TUJUAN DAN MANFAAT

1. Tujuan Praktikum

Tujuan dari praktikum MediaPipe Hand ini adalah sebagai berikut:

1. Memahami konsep dasar **computer vision** dalam pendeteksian dan pelacakan tangan secara real-time.
2. Mengetahui cara kerja **MediaPipe Hand** dalam mendeteksi tangan dan menentukan titik-titik landmark pada tangan.
3. Mengimplementasikan **MediaPipe Hand** menggunakan bahasa pemrograman **Python**.
4. Mengintegrasikan MediaPipe dengan **OpenCV** untuk menampilkan hasil deteksi tangan melalui kamera.
5. Menggunakan **PyCharm** sebagai lingkungan pengembangan untuk menulis, menjalankan, dan melakukan *debugging* program.
6. Menganalisis hasil deteksi tangan berdasarkan posisi dan pergerakan landmark yang dihasilkan.

2. Manfaat Praktikum

Manfaat yang diperoleh dari praktikum MediaPipe Hand ini antara lain:

1. Meningkatkan pemahaman mahasiswa terhadap penerapan **computer vision** dalam dunia nyata.
2. Memberikan pengalaman praktis dalam penggunaan **MediaPipe** dan **OpenCV** untuk pengolahan citra dan video.
3. Melatih kemampuan pemrograman **Python** dalam pengembangan aplikasi berbasis visi komputer.
4. Menambah wawasan mengenai teknologi **hand tracking** yang dapat dikembangkan menjadi sistem pengenalan gesture.
5. Menjadi dasar pengembangan aplikasi lanjutan seperti **kontrol tanpa sentuhan**, **augmented reality**, dan **human-computer interaction**.

C. PROGRAM & ANALISA

1. Hand Detection

```
import cv2
import mediapipe
capture = cv2.VideoCapture(0)
```

```

mediapipehand = mediapipe.solutions.hands
tangan = mediapipehand.Hands()

while True:
    success, frame = capture.read()
    imRGB = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = tangan.process(imRGB)
    if results.multi_hand_landmarks:
        print("tangan")
    else: print("tidak ada")

    cv2.imshow("webcam", frame)
    cv2.waitKey(10)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
capture.release()
cv2.destroyAllWindows()

```

Program ini digunakan untuk mendeteksi keberadaan tangan secara **real-time** menggunakan kamera dengan bantuan pustaka **MediaPipe Hand**. Program diawali dengan mengaktifkan webcam sebagai input video:

```
capture = cv2.VideoCapture(0)
```

Webcam digunakan untuk menangkap citra tangan secara langsung. Selanjutnya, modul **MediaPipe Hand** diinisialisasi:

```

mediapipehand = mediapipe.solutions.hands
tangan = mediapipehand.Hands()

```

Bagian ini berfungsi untuk memanggil model pendeteksi tangan yang akan digunakan selama program berjalan. Setiap frame yang ditangkap kamera kemudian dikonversi dari format **BGR** ke **RGB**:

```
imRGB = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
```

Konversi ini diperlukan karena MediaPipe memproses citra dalam format RGB. Proses deteksi tangan dilakukan dengan

```
results = tangan.process(imRGB)
```

Hasil deteksi disimpan dalam variabel `results`.

Untuk mengetahui apakah tangan terdeteksi, dilakukan pengecekan sebagai berikut:

```

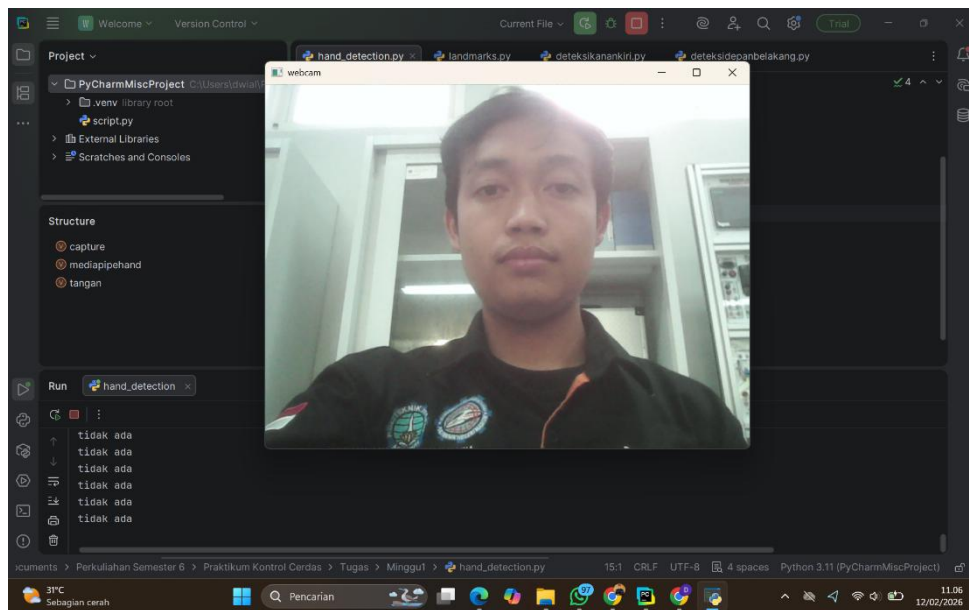
if results.multi_hand_landmarks:
    print("tangan")
else:
    print("tidak ada")

```

Jika tangan terdeteksi, program akan menampilkan teks “*tangan*”, dan jika tidak terdeteksi akan menampilkan teks “*tidak ada*” pada terminal. Hasil video tetap ditampilkan secara real-time dan program dapat dihentikan dengan menekan tombol **q**:

```
cv2.imshow("webcam", frame)
```

Program ini mampu mendeteksi keberadaan tangan secara real-time menggunakan MediaPipe Hand. Deteksi ditunjukkan melalui informasi teks pada terminal, sehingga program ini dapat digunakan sebagai dasar pengembangan sistem hand tracking yang lebih lanjut.



Gambar 1. Kondisi tangan tidak terdeteksi di kamera



Gambar 2. Kondisi tangan terdeteksi di kamera

2. Landmarks

```
import cv2
import mediapipe as mp

# Inisialisasi kamera
capture = cv2.VideoCapture(0)
```

```

# Inisialisasi MediaPipe
mp_hands = mp.solutions.hands
tangan = mp_hands.Hands(max_num_hands=1)
mpDraw = mp.solutions.drawing_utils

while True:
    success, img = capture.read()
    if not success:
        break

    # Konversi ke RGB
    imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    results = tangan.process(imgRGB)

    # Jika tangan terdeteksi
    if results.multi_hand_landmarks:
        for titiktangan in results.multi_hand_landmarks:
            mpDraw.draw_landmarks(
                img,
                titiktangan,
                mp_hands.HAND_CONNECTIONS
            )

            # Ambil koordinat tiap titik
            for id, titik in enumerate(titiktangan.landmark):
                print(id, titik.x, titik.y)

    # Tampilkan kamera
    cv2.imshow("Hand Detection", img)

    # Tekan q untuk keluar
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

capture.release()
cv2.destroyAllWindows()

```

Program ini merupakan pengembangan dari program sebelumnya, dengan tujuan menampilkan **titik-titik landmark tangan** serta **garis penghubung antar landmark** secara real-time menggunakan MediaPipe Hand. Program diawali dengan inisialisasi kamera dan modul MediaPipe:

```

capture = cv2.VideoCapture(0)
mp_hands = mp.solutions.hands
tangan = mp_hands.Hands(max_num_hands=1)
mpDraw = mp.solutions.drawing_utils

```

Parameter `max_num_hands=1` digunakan untuk membatasi deteksi hanya pada satu tangan.

Setiap frame yang diperoleh dari kamera dikonversi ke format RGB sebelum diproses:

```

imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
results = tangan.process(imgRGB)

```

Konversi warna diperlukan karena MediaPipe bekerja dengan format RGB. Apabila tangan terdeteksi, landmark tangan akan digambar pada citra:

```

mpDraw.draw_landmarks(
    img,
    titiktangan,
    mp_hands.HAND_CONNECTIONS
)

```

Bagian ini menampilkan **21 titik landmark** beserta **garis koneksi** antar titik pada tangan. Selain visualisasi, program juga mengambil koordinat setiap landmark:

```

for id, titikt in enumerate(titiktangan.landmark):
    print(id, titikt.x, titikt.y)

```

Koordinat x dan y dinyatakan dalam bentuk nilai relatif (0–1) terhadap ukuran frame. Hasil deteksi ditampilkan secara real-time melalui jendela kamera:

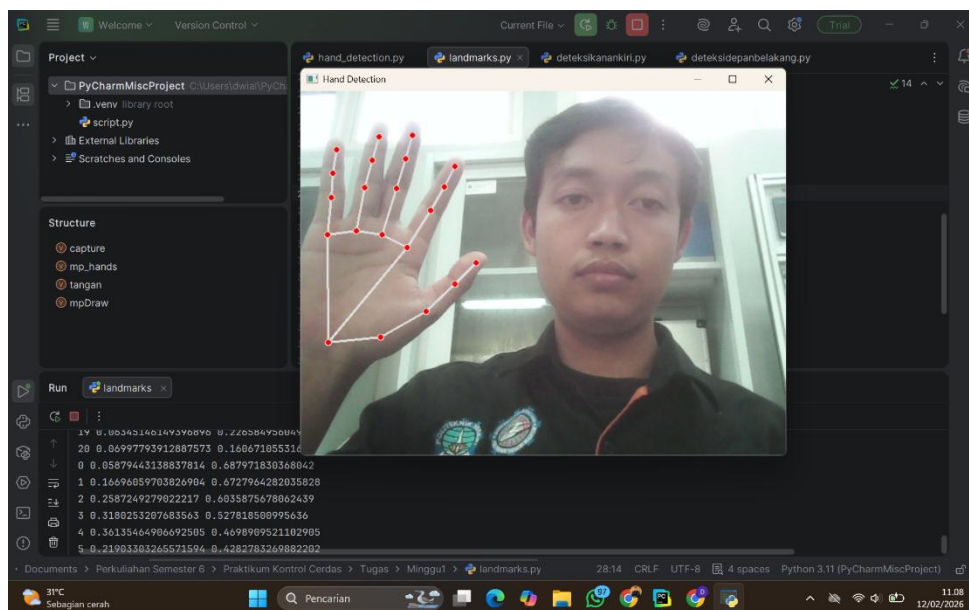
```

cv2.imshow("Hand Detection", img)

```

Program dapat dihentikan dengan menekan tombol **q** pada keyboard.

Program ini berhasil menampilkan landmark tangan berupa titik dan garis penghubung secara real-time. Selain visualisasi, program juga mampu menampilkan koordinat setiap landmark, sehingga dapat digunakan sebagai dasar untuk analisis gerakan tangan dan pengembangan sistem pengenalan gesture.



Gambar 3. Tampilan 21 titik landmark beserta garis koneksi

3. Deteksi Tangan Kanan dan Kiri

```

import cv2
import mediapipe
capture = cv2.VideoCapture(0)
mediapipehand = mediapipe.solutions.hands
tangan = mediapipehand.Hands(max_num_hands=1)

```

```

mpdraw = mediapipe.solutions.drawing_utils
while True:
    success, img = capture.read() #read video frame by frame
    image=cv2.flip(img,1)
    if not success:
        print("Webcam tidak terbaca")
        break
    imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    results = tangan.process(imgRGB)
    if results.multi_hand_landmarks:
        for titiktangan in results.multi_hand_landmarks:
            mpdraw.draw_landmarks(
img,titiktangan,mediapipehand.HAND_CONNECTIONS)
            for id, titik in enumerate(titiktangan.landmark):
                print(id, titik.x, titik.y)
    if results.multi_handedness:
        for hand in results.multi_handedness:
            if hand.classification[0].index == 1:
                cv2.putText(img, "Left", (200, 50),
                    cv2.FONT_HERSHEY_PLAIN, 3, (255, 0, 0), 3)
            else:
                cv2.putText(img, "Righth", (200, 50),
                    cv2.FONT_HERSHEY_PLAIN, 3, (0, 0, 255), 3)
    cv2.imshow("webcam", img)
    if cv2.waitKey(10) & 0xFF == ord('q'):
        break #TUTUP webcam dan jendela saat q tampilan ditekan
capture.release()
cv2.destroyAllWindows()

```

Program ini bertujuan untuk mendeteksi **tangan kanan (Right)** dan **tangan kiri (Left)** secara real-time menggunakan MediaPipe Hand, sekaligus menampilkan landmark tangan. Program diawali dengan inisialisasi kamera dan modul MediaPipe:

```

capture = cv2.VideoCapture(0)
mediapipehand = mediapipe.solutions.hands
tangan = mediapipehand.Hands(max_num_hands=1)
mpdraw = mediapipe.solutions.drawing_utils

```

Parameter `max_num_hands=1` digunakan agar sistem hanya mendeteksi satu tangan dalam satu waktu. Frame video yang diperoleh kemudian dibalik secara horizontal:

```

image = cv2.flip(img, 1)

```

Proses *flip* ini bertujuan agar tampilan kamera menyerupai cermin, sehingga pergerakan tangan terlihat lebih natural. Selanjutnya, frame dikonversi ke format RGB dan diproses oleh MediaPipe:

```

imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
results = tangan.process(imgRGB)

```

Jika landmark tangan terdeteksi, maka titik-titik landmark dan garis penghubungnya akan digambar:

```

mpdraw.draw_landmarks(
    img,

```



```

titiktangan,
mediapipehand.HAND_CONNECTIONS
)

```

Selain itu, koordinat setiap landmark juga ditampilkan pada terminal:

```

for id, titik in enumerate(titiktangan.landmark):
    print(id, titik.x, titik.y)

```

Untuk membedakan tangan kanan dan kiri, digunakan informasi `multi_handedness`:

```

if results.multi_handedness:
    for hand in results.multi_handedness:
        if hand.classification[0].index == 1:
            cv2.putText(img, "Left", (200, 50),
                        cv2.FONT_HERSHEY_PLAIN, 3, (255, 0, 0), 3)
        else:
            cv2.putText(img, "Right", (200, 50),
                        cv2.FONT_HERSHEY_PLAIN, 3, (0, 0, 255), 3)

```

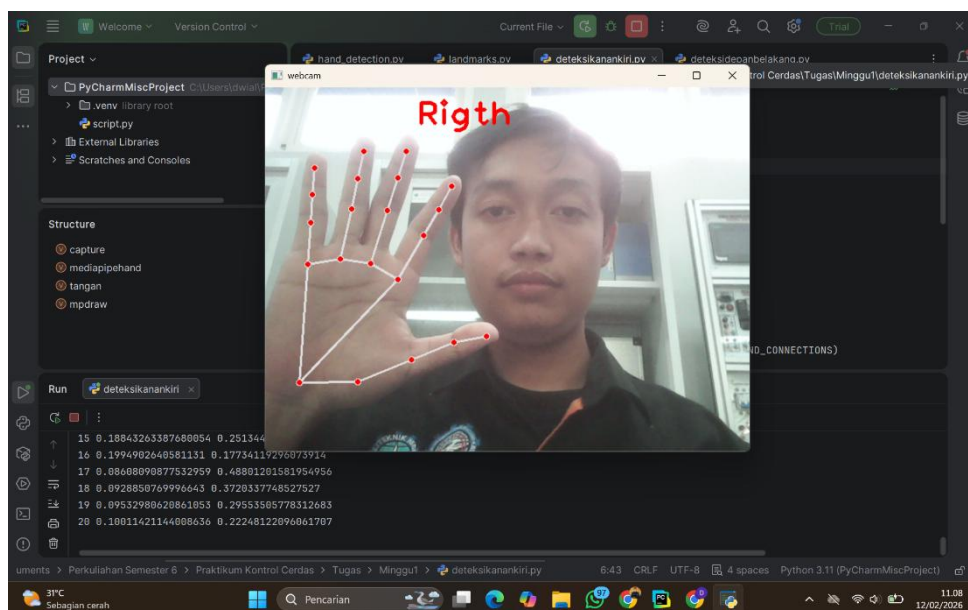
Teks **Left** atau **Right** akan ditampilkan pada layar sesuai dengan tangan yang terdeteksi. Hasil akhir ditampilkan melalui jendela webcam, dan program dapat dihentikan dengan menekan tombol **q**:

```

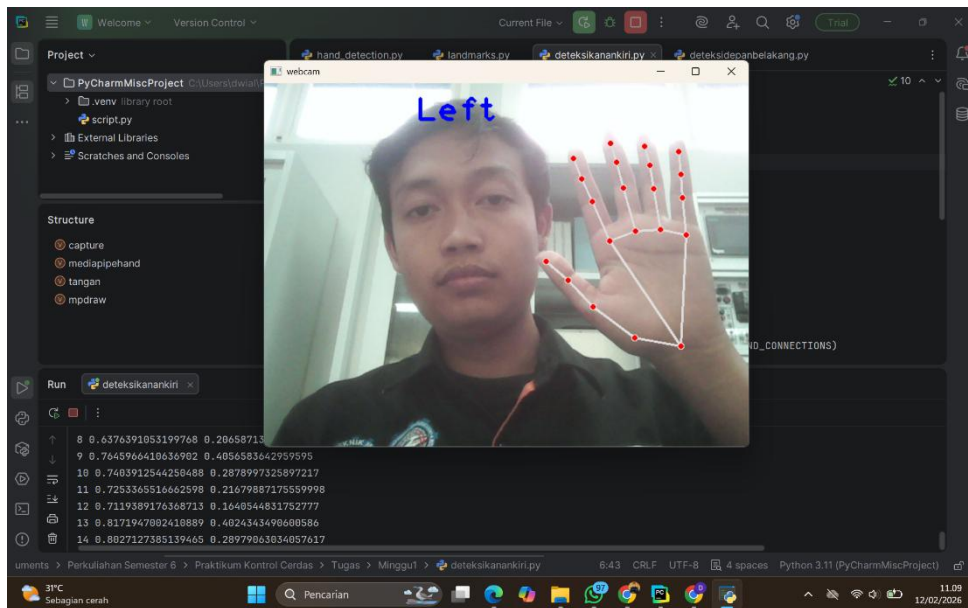
cv2.imshow("webcam", img)

```

Program ini mampu mendeteksi tangan serta membedakan antara tangan kanan dan tangan kiri secara real-time. Informasi handedness yang dihasilkan MediaPipe dapat dimanfaatkan untuk pengembangan aplikasi lanjutan, seperti sistem kontrol berbasis gesture atau interaksi manusia dan komputer.



Gambar 4. Deteksi tangan kanan



Gambar 5. Deteksi kanan kiri

4. Deteksi Depan dan Belakang Tangan

```
import cv2
import mediapipe
capture = cv2.VideoCapture(0)
mediapipehand = mediapipe.solutions.hands
tangan = mediapipehand.Hands(max_num_hands=1)
mpdraw = mediapipe.solutions.drawing_utils
while True:
    success, img = capture.read()
    if not success:
        print("Webcam tidak terbaca")
        break
    # mirror webcam
    image = cv2.flip(img, 1)
    imgRGB = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    results = tangan.process(imgRGB)

    if results.multi_hand_landmarks and results.multi_handedness:
        for titiktangan, hand in zip(results.multi_hand_landmarks,
                                     results.multi_handedness):
            mpdraw.draw_landmarks(image, titiktangan,
            mediapipehand.HAND_CONNECTIONS)
            # landmark penting
            thumb = titiktangan.landmark[4] # ujung jempol
            wrist = titiktangan.landmark[0] # pergelangan
            # jenis tangan (Left / Right)
            hand_label = hand.classification[0].label
            # LOGIKA DEPAN / BELAKANG
            status = ""
            if hand_label == "Right":
                if thumb.x < wrist.x:
                    status = "DEPAN"
                else:
                    status = "BELAKANG"
            elif hand_label == "Left":
                if thumb.x > wrist.x:
```

```

        status = "DEPAN"
    else:
        status = "BELAKANG"
    # TAMPILKAN HASIL
    cv2.putText(
        image,
        f"{hand_label.upper()} - {status}", (100,
50), cv2.FONT_HERSHEY_PLAIN, 2.5, (0, 255, 0), 3 )
    cv2.imshow("webcam", image)
    if cv2.waitKey(10) & 0xFF == ord('q'):
        break
capture.release()
cv2.destroyAllWindows()

```

Program ini bertujuan untuk mendeteksi apakah tangan yang tertangkap kamera berada pada posisi **bagian depan (telapak tangan)** atau **bagian belakang tangan** dengan memanfaatkan informasi landmark dan handedness dari MediaPipe Hand. Program diawali dengan inisialisasi kamera dan modul MediaPipe:

```

capture = cv2.VideoCapture(0)
mediapipehand = mediapipe.solutions.hands
tangan = mediapipehand.Hands(max_num_hands=1)
mpdraw = mediapipe.solutions.drawing_utils

```

Frame video yang diperoleh kemudian dibalik secara horizontal (*mirror*) agar tampilan kamera lebih natural:

```

image = cv2.flip(img, 1)

```

Selanjutnya, citra dikonversi ke format RGB dan diproses oleh MediaPipe:

```

imgRGB = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
results = tangan.process(imgRGB)

```

Jika tangan terdeteksi, landmark tangan akan digambar dan dianalisis:

```

mpdraw.draw_landmarks(
    image,
    titiktangan,
    mediapipehand.HAND_CONNECTIONS )

```

Untuk menentukan bagian depan atau belakang tangan, digunakan dua landmark penting, yaitu:

```

thumb = titiktangan.landmark[4] # ujung jempol
wrist = titiktangan.landmark[0] # pergelangan tangan

```

Jenis tangan diperoleh dari informasi handedness:

```

hand_label = hand.classification[0].label

```

Logika penentuan posisi tangan dilakukan dengan membandingkan posisi sumbu **x** antara jempol dan pergelangan tangan:

```

if hand_label == "Right":
    if thumb.x < wrist.x:

```

```

        status = "DEPAN"
    else:
        status = "BELAKANG"
elif hand_label == "Left":
    if thumb.x > wrist.x:
        status = "DEPAN"
    else:
        status = "BELAKANG"

```

Logika ini bekerja berdasarkan orientasi tangan kanan dan kiri yang berbeda terhadap kamera.

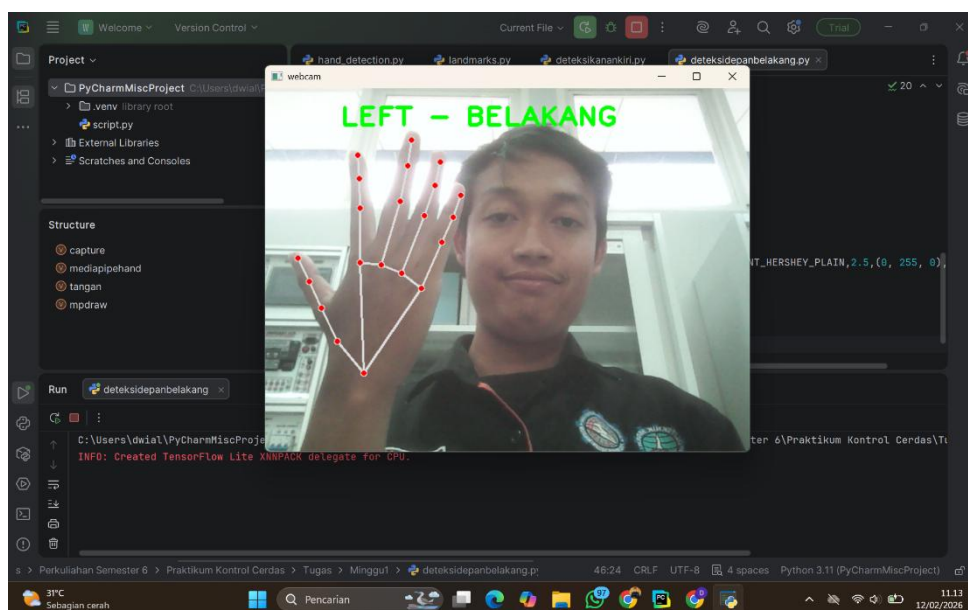
Hasil deteksi kemudian ditampilkan pada layar:

```

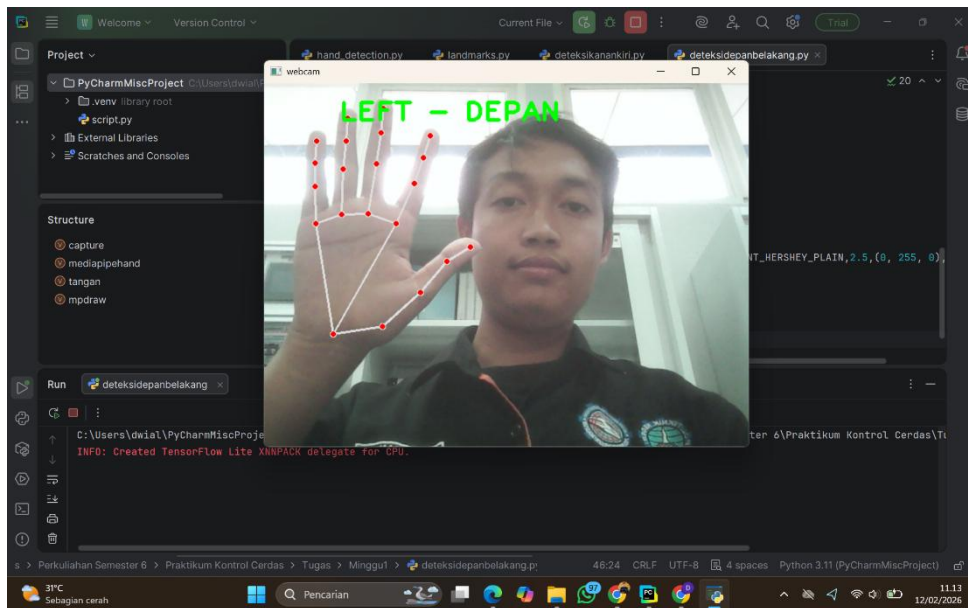
cv2.putText(
    image,
    f"{hand_label.upper()} - {status}",
    (100, 50),
    cv2.FONT_HERSHEY_PLAIN,
    2.5,
    (0, 255, 0),
    3 )

```

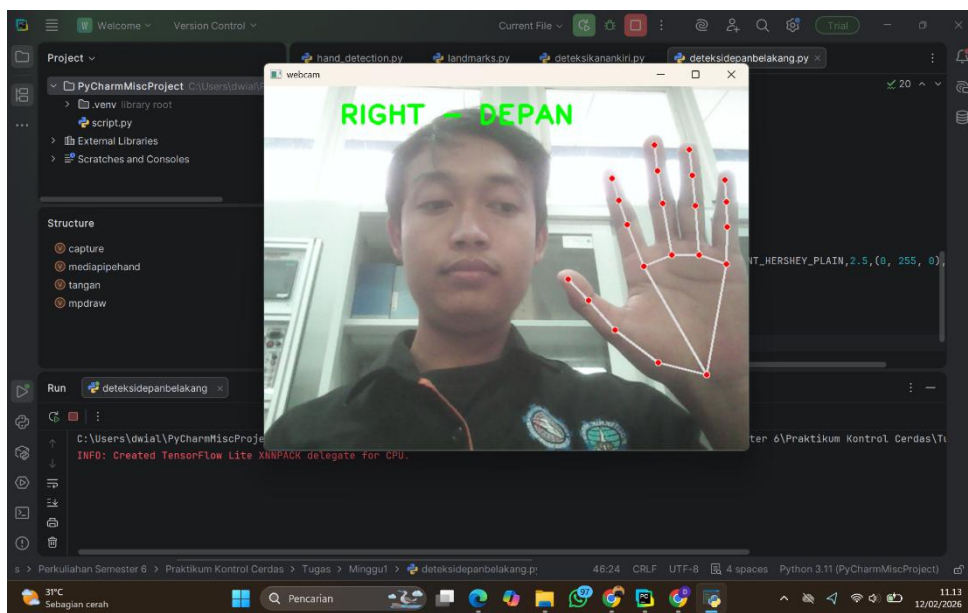
Program dapat dihentikan dengan menekan tombol **q** pada keyboard. Program ini berhasil mendeteksi orientasi tangan, yaitu bagian depan (telapak tangan) dan bagian belakang tangan secara real-time. Dengan memanfaatkan landmark jempol dan pergelangan tangan serta informasi handedness, sistem mampu menentukan arah tangan secara sederhana dan efektif. Program ini dapat dikembangkan lebih lanjut untuk aplikasi pengenalan gesture dan interaksi tanpa sentuhan.



Gambar 6. Deteksi tangan kiri bagian belakang



Gambar 7. Deteksi tangan kiri bagian depan



Gambar 8. Deteksi tangan kanan bagian depan



Gambar 9. Deteksi tangan kanan bagian belakang

D. REFERENSI

Belajar Python - Situs Open Source Tutorial Pemrograman Python Bahasa Indonesia - <https://belajarpython.com>

Python project : AI hand tracking using python (Media pipe) - <https://www.youtube.com/watch?v=TACu8y2T2ps>

Custom Hand Gesture Recognition with Hand Landmarks Using Google's Mediapipe + OpenCV in Python - https://www.youtube.com/watch?v=a99p_fAr6e4

Right and Left Hand Detection Using Python - <https://www.geeksforgeeks.org/python/right-and-left-hand-detection-using-python/>