

LAPORAN PRAKTIKUM 13 MEDIPIPE HAND

Minggu Ke-1

Nama : Fadia Hamida Hidayat

NIM : 234308069

Kelas : TKA-6C

Akun Github (Tautan) : <https://github.com/fadiahamida>

A. Pendahuluan

MediaPipe merupakan sebuah framework yang digunakan untuk membuat pipeline yang akan digunakan pada machine learning untuk pemrosesan data dalam bentuk video, audio, dsb. Salah satu modul yang tersedia pada MediaPipe adalah MediaPipe Hands, yang berfungsi untuk mendeteksi dan melacak tangan manusia menggunakan kamera. MediaPipe Hands mampu mengenali titik-titik penting (landmark) pada tangan, sehingga posisi dan pergerakan tangan dapat dianalisis dengan baik. MediaPipe terbukti mampu mendeteksi tangan secara real-time dengan tingkat akurasi yang cukup baik, terutama jika didukung oleh kondisi pencahayaan dan jarak kamera yang sesuai.

MediaPipe Hands dapat diintegrasikan dengan OpenCV untuk membangun sistem pengenalan gestur tangan. OpenCV berperan dalam pengambilan citra dari webcam dan menampilkan hasil pemrosesan, sedangkan MediaPipe digunakan untuk mendeteksi tangan pada citra tersebut. Kombinasi kedua library ini banyak digunakan, seperti pada sistem pengenalan bahasa isyarat dan sistem kendali berbasis gerakan tangan, karena mudah diimplementasikan dan memiliki performa yang stabil. Pada praktikum ini dilakukan pembuatan sistem deteksi tangan menggunakan MediaPipe Hands dan OpenCV. Sistem yang dirancang bertujuan untuk mendeteksi keberadaan tangan melalui webcam dan menampilkan hasilnya secara real-time pada layar.

B. Tujuan :

Tujuan dari praktikum MediaPipe Hand ini adalah sebagai berikut:

1. Mempelajari konsep dasar deteksi dan pelacakan tangan menggunakan MediaPipe Hands.
2. Mengimplementasikan MediaPipe Hands dan OpenCV untuk mendeteksi tangan secara real-time melalui webcam.

3. Memahami proses pengambilan citra, pengolahan citra, dan penampilan hasil deteksi tangan pada layar.
4. Mengetahui cara kerja MediaPipe dalam mengenali landmark tangan serta hubungannya dengan gerakan tangan.

C. Manfaat :

Manfaat yang diharapkan dari praktikum MediaPipe Hand ini adalah sebagai berikut:

1. Memberikan pemahaman praktis mengenai penerapan teknologi computer vision dalam mendeteksi gerakan tangan.
2. Menambah wawasan tentang penggunaan MediaPipe sebagai framework pendekripsi tangan berbasis machine learning.
3. Meningkatkan keterampilan dalam menggunakan OpenCV untuk pengolahan citra digital.
4. Menjadi dasar pengembangan aplikasi lanjutan berbasis gestur tangan, seperti sistem kendali, pengenalan bahasa isyarat, atau antarmuka tanpa sentuhan.

D. Hasil Running Program

1. Pendahuluan A

```

1 import cv2 #import module opencv
2 import mediapipe as mp
3
4 capture = cv2.VideoCapture(0) #video capture pada device kamera nomer 0
5
6 mp_hand = mp.solutions.hands #inisialisasi deteksi tangan
7
8 tangan = mp_hand.Hands() #variable tangan untuk menyimpan konfigurasi deteksi tangan
9
10 while True:
11
12     success, img = capture.read() #menyimpan citra tangan
13
14     imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB) #merubah citra ke RGB
15
16     result = tangan.process(imgRGB) #melakukan proses
17
18     if result.multi_hand_landmarks:
19
20         print("tangan") #ketika tangan terdeteksi muncul pesan

```

2. Pendahuluan B

The screenshot shows the PyCharm IDE interface. The left sidebar displays the project structure under 'PyCharmMiscProject' with files like 'coba.py', 'P13 B.py', 'P13 LAT3.py', and 'script.py'. The main editor window shows Python code for hand tracking using OpenCV and Mediapipe. The code reads frames from a video capture, processes them with Mediapipe's hands solution, and draws landmarks on the detected hands. The bottom run tab shows the output of 'P13 B.py' which includes numerical values and a count of 20. To the right of the editor is a preview window titled 'webcam' showing a video feed of a person's hand with red circular landmarks and white connecting lines indicating the detected hand joints.

```
1 > import ...
2 capture = cv2.VideoCapture(0)
3 mediapipehand = mediapipe.solutions.hands
4 tangan = mediapipehand.Hands(max_num_hands=1)
5 mpdraw = mediapipe.solutions.drawing_utils
6
7
8 while True:
9     success, img = capture.read() # Read video frame by frame
10    imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
11    results = tangan.process(imgRGB)
12    if results.multi_hand_landmarks:
13        for titiktangan in results.multi_hand_landmarks:
14            mpdraw.draw_landmarks(img,titiktangan,mediapipehand.HAND_CONNECTIONS)
15            for id, titik in enumerate(titiktangan.landmark):
16                print(id)
17                print(titik.x)
18                print(titik.y)
19                cv2.imshow("webcam",img)
20                cv2.waitKey(10)
21    if cv2.waitKey(10) & 0xFF == ord('q'):
```

Run P13 B

19
0.4941567249888306
0.35114922046661377
20
0.49481967091560364
0.2817535698413849

D:\ SEMESTER 6 > P13 B.py

1:1 CRLF UTF-8 4 spaces Python 3.11 (PyCharmMiscProject) 11/02/2026

3. Latihan A

```
R_BGR2RGB)

ts.multi_handedness);
dex == 1; # klasifikasi tangan kanan
men, org (200, 50),
SHEY_PLAIN, fontScale: 5, color: (255, 0, 0), thickness: 3) # memberikan teks pada tampilan jendela
index == 0; # klasifikasi tangan kiri
, org (200, 50),
AIN, fontScale: 5, color: (0, 0, 255), thickness: 3)

on", img)

INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
N0000 00:00:1770815515.148104 8504 inference feedback_manager.cc:114] Feedback manager requires a model with a single signature inference. Disabling support for feedback to
N0000 00:00:1770815515.181364 22082 inference_feedback_manager.cc:114] Feedback manager requires a model with a single signature inference. Disabling support for feedback to
C:\Users\user\AppData\Local\Programs\Python\Python311\lib\site-packages\google\protobuf\symbol_database.py:55: UserWarning: SymbolDatabase.GetPrototype() is deprecated. Please
    warnings.warn('SymbolDatabase.GetPrototype() is deprecated. Please '
```

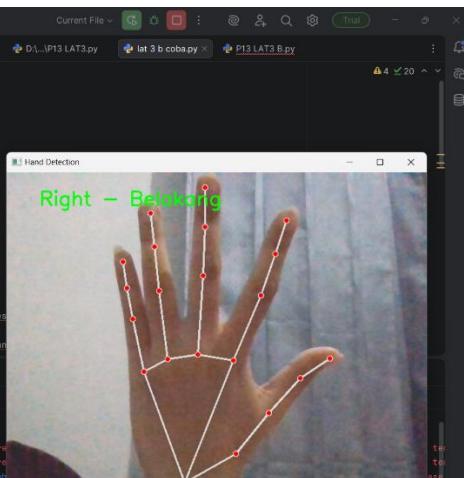
```
    r_BGR2RGB)

    ts.multi_handedness);
    index == 1; # klasifikasi tangan kanan
    "kanan", org (200, 50),
    SHEY_PLAIN, fontScale:5, color (255, 0, 0), thickness: 3) # memberikan teks pada tampilan jendela
    index == 0; # klasifikasi tangan kiri
    "kiri", org (200, 50),
    AIN, fontScale:5, color (0, 0, 255), thickness: 3)
    on", img)

on
```

INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
#0000 00:00:1270815515.16R10A 8504 inference_feedback_manager.cc:114] Feedback manager requires a model with a single signature inference. Disabling support for feedback to
#0000 00:00:1270815515.181364 22037 inference_feedback_manager.cc:114] Feedback manager requires a model with a single signature inference. Disabling support for feedback to
C:\Users\user\appdata\local\temp\1\blisite_packages\google\protobuf\symbol_database.pb: UserWarning: SymbolDatabase.GetPrototype() is deprecated. Please
warnings.warn('SymbolDatabase.GetPrototype() is deprecated. Please '

4. Latihan B



Screenshot of PyCharm showing the code for hand detection and a video feed window titled "Hand Detection". The video shows a right hand with red dots at the detected landmarks. The text "Right - Belakang" is displayed in green at the top left of the video frame.

```
import cv2
import mediapipe as mp
...
cap = cv2.VideoCapture(0)

mp_hands = mp.solutions.hands
mp_drawing = mp.solutions.drawing_utils
hands = mp_hands.Hands()

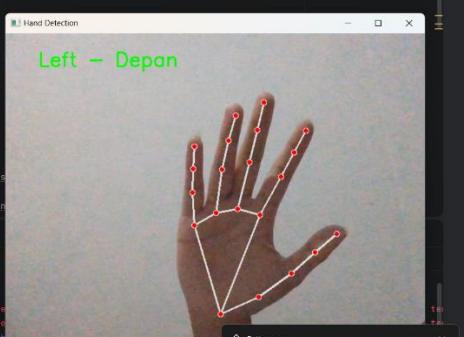
while True:
    success, img = cap.read()
    if not success:
        break

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

    if results.multi_hand_landmarks and results.multi_handedness:
        for hand_landmarks, hand_label in zip(results.multi_hand_landmarks, results.multi_handedness):
            mp_drawing.draw_landmarks(img, hand_landmarks, mp_hands.HAND_CONNECTIONS)

    cv2.imshow("Hand Detection", img)

    if cv2.waitKey(1) == ord('q'):
        break
```



Screenshot of PyCharm showing the same hand detection code and a video feed window titled "Hand Detection". The video shows a left hand with red dots at the detected landmarks. The text "Left - Depan" is displayed in green at the top left of the video frame.

```
import cv2
import mediapipe as mp
...
cap = cv2.VideoCapture(0)

mp_hands = mp.solutions.hands
mp_drawing = mp.solutions.drawing_utils
hands = mp_hands.Hands()

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

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

    if results.multi_hand_landmarks and results.multi_handedness:
        for hand_landmarks, hand_label in zip(results.multi_hand_landmarks, results.multi_handedness):
            mp_drawing.draw_landmarks(img, hand_landmarks, mp_hands.HAND_CONNECTIONS)

    cv2.imshow("Hand Detection", img)

    if cv2.waitKey(1) == ord('q'):
        break
```