

**PRAKTIKUM 13**  
**MEDIAPIPE HAND**  
**Laporan Sementara**

Nama : Muchamad Yusuf Revaldo

Nim : 234308078

Kelas : TKA-6C

**Pendahuluan**

Pada praktikum ini digunakan modul MediaPipe Hands yang berfungsi untuk mendeteksi posisi serta titik-titik landmark pada tangan secara real-time. Setiap tangan memiliki 21 titik landmark yang dapat digunakan untuk menganalisis posisi, orientasi, dan pergerakan tangan. Mediapipe merupakan sebuah framework yang digunakan untuk membuat pipeline yang akan digunakan pada machine learning untuk pemrosesan data dalam bentuk video, audio, dsb. Mediapipe pada python dapat diinstall dengan menggunakan command `pip install mediapipe` pada terminal. Implementasi program dilakukan menggunakan bahasa pemrograman Python dengan bantuan software PyCharm. PyCharm digunakan untuk menulis, menjalankan, serta melakukan debugging program, library MediaPipe pada Python dapat diinstal melalui terminal.

**Tujuan**

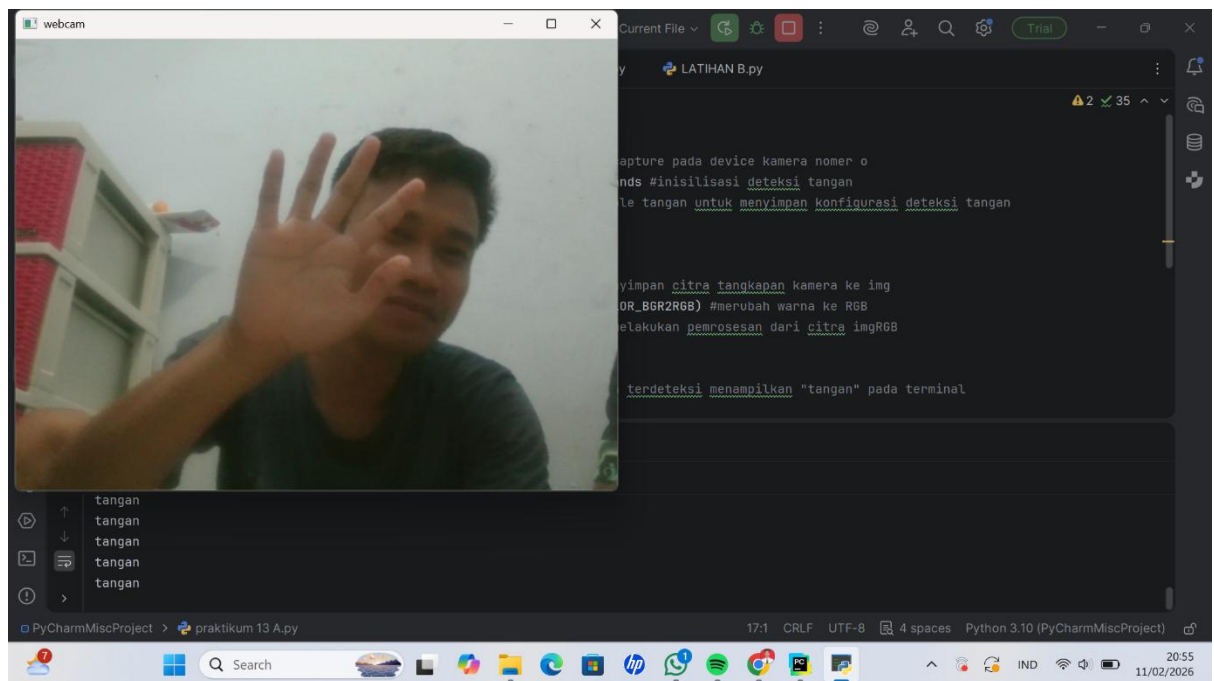
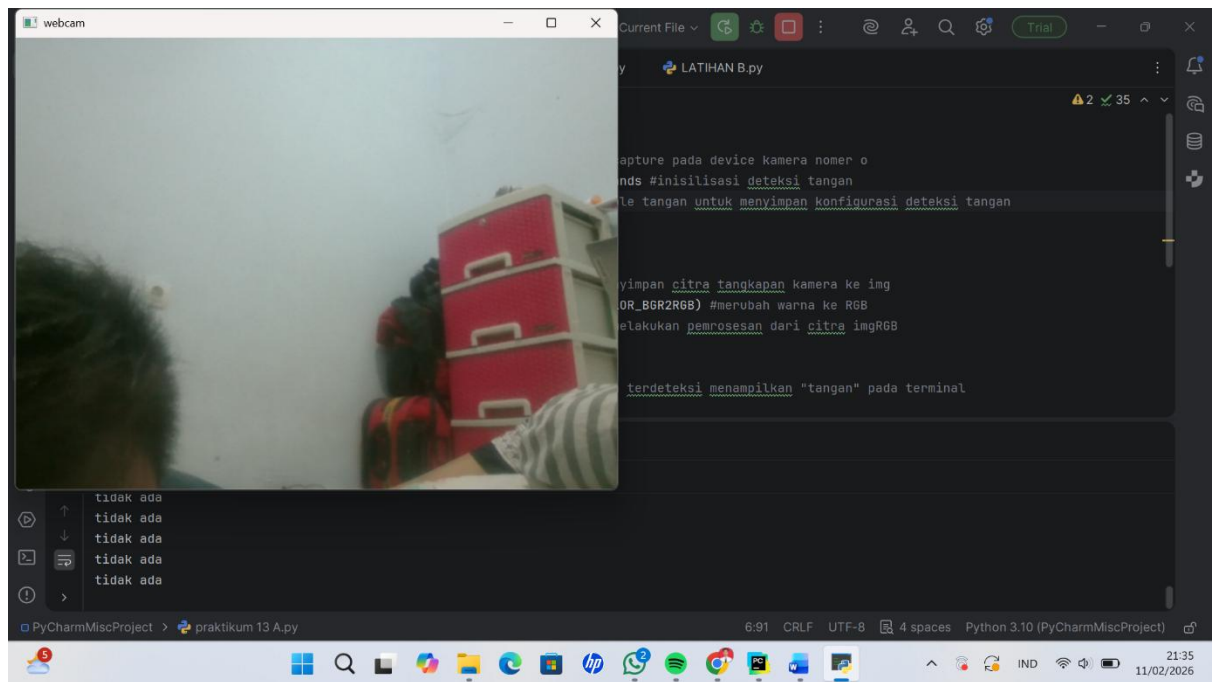
1. Mahasiswa mempelajari cara instalasi dan konfigurasi library MediaPipe pada pemrograman Python.
2. Mahasiswa memahami konsep landmark pada modul MediaPipe Hands sebagai titik acuan dalam mendeteksi posisi dan pergerakan tangan.
3. Mahasiswa mengimplementasikan program deteksi tangan kanan dan kiri menggunakan fitur klasifikasi bawaan MediaPipe.

**Manfaat**

1. Mahasiswa dapat pemahaman mengenai konsep dasar menggunakan Software Pycharm.
2. Mahasiswa dapat mengimplementasikan deteksi tangan secara real-time menggunakan bahasa pemrograman Python dan webcam.
3. Mahasiswa dapat memahami serta memanfaatkan titik *landmark* sebagai acuan analisis posisi dan orientasi tangan.

## Hasil Percobaan

### A. Percobaan Deteksi Tampilan Tangan Program Dan Hasil Running Program



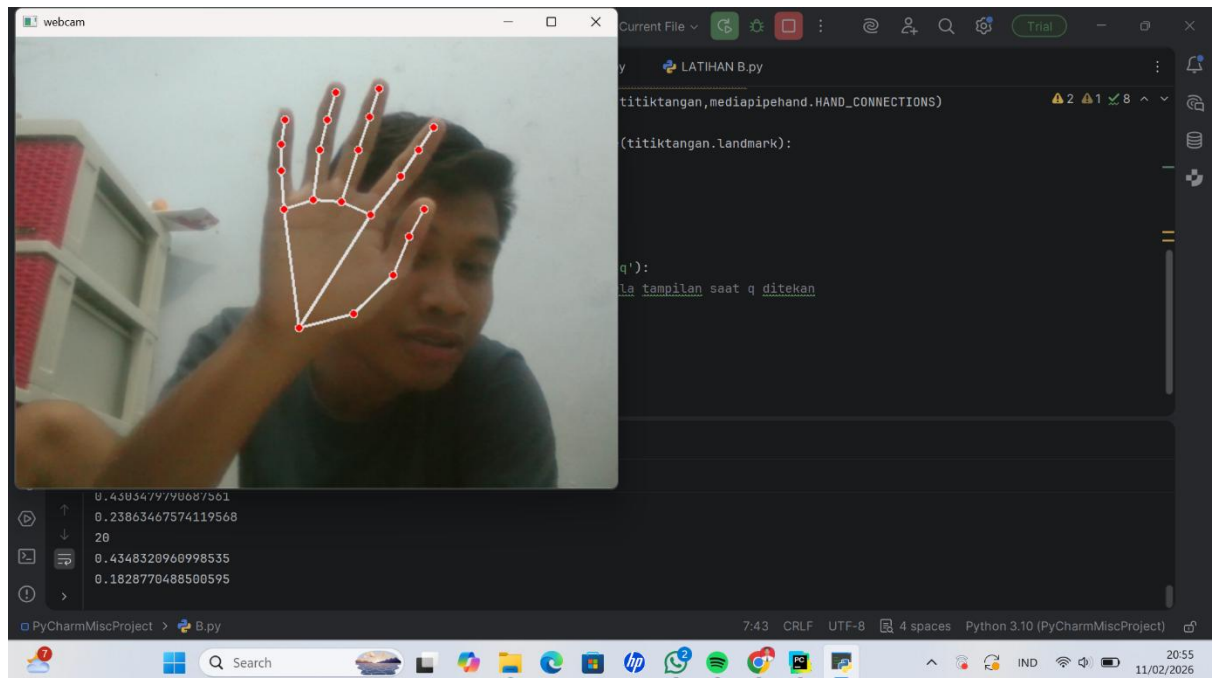
```

1  import cv2 #import module opencv
2  import mediapipe
3
4  capture = cv2.VideoCapture(0) #video capture pada device kamera nomer 0
5  mediapipehand = mediapipe.solutions.hands #inisialisasi deteksi tangan
6  tangan = mediapipehand.Hands() #variable tangan untuk menyimpan konfigurasi deteksi tangan
7
8  while True:
9
10     success, img = capture.read() #menyimpan citra tangkapan kamera ke img
11     imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB) #merubah warna ke RGB
12     result = tangan.process(imgRGB) #melakukan pemrosesan dari citra imgRGB
13
14     if result.multi_hand_landmarks:
15         print("tangan") #ketika tangan terdeteksi menampilkan "tangan" pada terminal
16     else:
17
18         print("tidak ada") #ketika tangan terdeteksi menampilkan "tidak ada" pada terminal
19     cv2.imshow( winname: "webcam", img)
20     cv2.waitKey(10)
21     if cv2.waitKey(10) & 0xFF == ord('q'):
22         break
23
24 capture.release()
25 cv2.destroyAllWindows()

```

## B. Percobaan Deteksi Tangan Titik Landmarks

### Program Dan Hasil Running Program



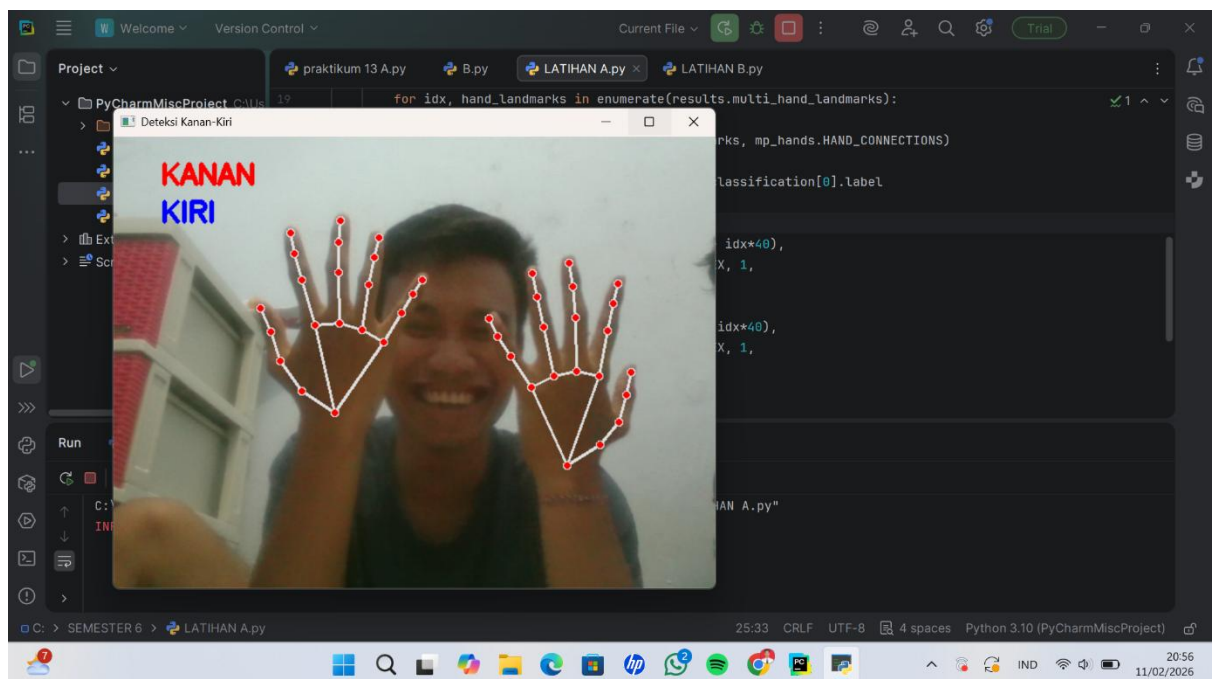
```

1  import cv2 #import module opencv
2  import mediapipe
3
4  capture = cv2.VideoCapture(0)
5  mediapipehand = mediapipe.solutions.hands
6  tangan = mediapipehand.Hands(max_num_hands=1)
7  mpdraw = mediapipe.solutions.drawing_utils
8
9  while True:
10
11     success, img = capture.read() # Read video frame by frame
12     imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
13     results = tangan.process(imgRGB)
14     if results.multi_hand_landmarks:
15         for titiktangan in results.multi_hand_landmarks:
16             mpdraw.draw_landmarks(img, titiktangan, mediapipehand.HAND_CONNECTIONS)
17
18             for id, titik in enumerate(titiktangan.landmark):
19                 print(id)
20                 print (titik.x)
21                 print (titik.y)
22
23     cv2.imshow( winname: "webcam",img)
24     cv2.waitKey(10)
25     if cv2.waitKey(10) & 0xFF == ord('q'):
26         break # Tutup webcam dan jendela tampilan saat q ditekan
27
28 capture.release()
29 cv2.destroyAllWindows()

```

### C. Percobaan Deteksi Posisi Tangan Kanan Dan Kiri

#### Program Dan Hasil Running Program



```

1  import cv2
2  import mediapipe as mp
3
4  cap = cv2.VideoCapture(0)
5
6  mp_hands = mp.solutions.hands
7  hands = mp_hands.Hands(max_num_hands=2)
8  mp_draw = mp.solutions.drawing_utils
9
10 while True:
11     success, img = cap.read()
12     if not success:
13         break
14
15     imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
16     results = hands.process(imgRGB)
17
18     if results.multi_hand_landmarks and results.multi_handedness:
19         for idx, hand_landmarks in enumerate(results.multi_hand_landmarks):
20
21             mp_draw.draw_landmarks(img, hand_landmarks, mp_hands.HAND_CONNECTIONS)
22
23             label = results.multi_handedness[idx].classification[0].label
24
25             if label == "Right":|

```

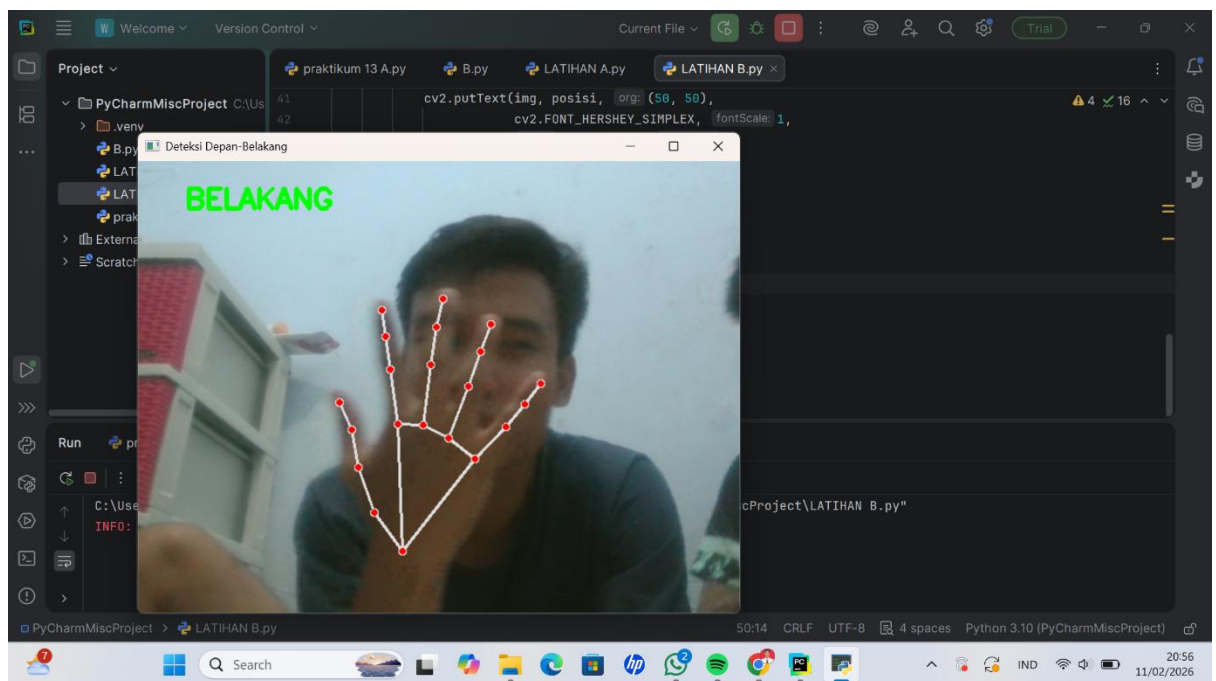
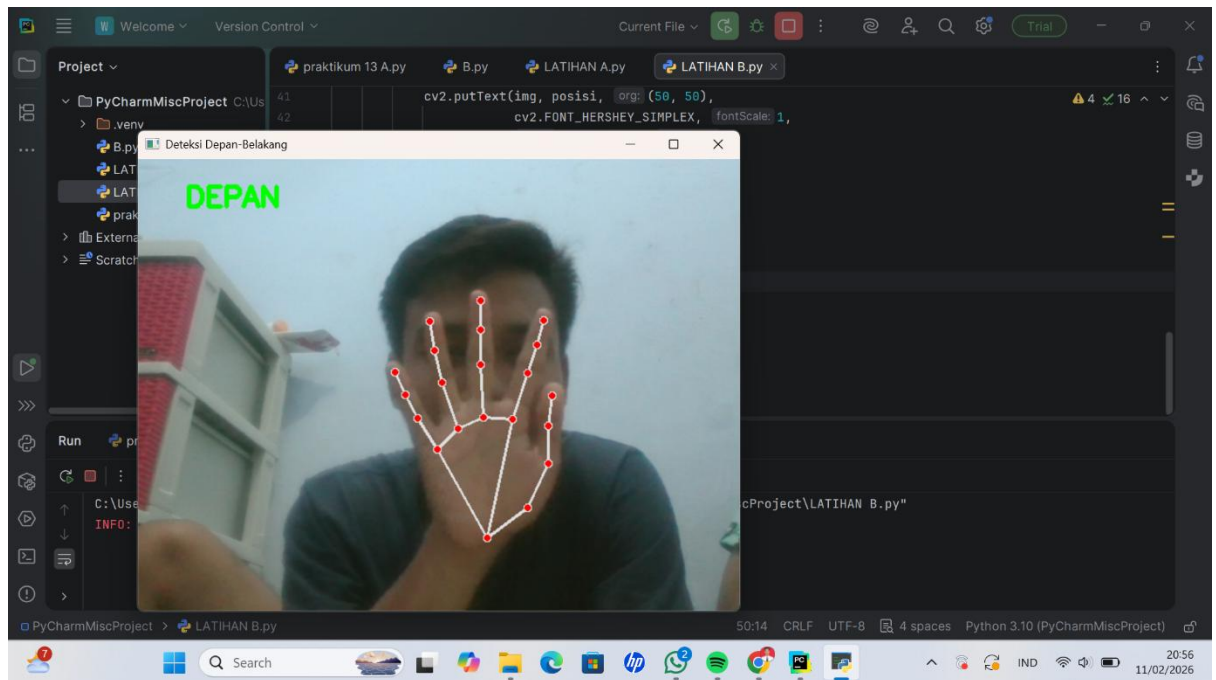
```

26                 cv2.putText(img, "KANAN", (50, 50 + idx*40),
27                               cv2.FONT_HERSHEY_SIMPLEX, 1,
28                               (0, 0, 255), 3)
29             else:
30                 cv2.putText(img, "KIRI", (50, 50 + idx*40),
31                               cv2.FONT_HERSHEY_SIMPLEX, 1,
32                               (255, 0, 0), 3)
33
34     cv2.imshow("Deteksi Kanan-Kiri", img)
35
36     if cv2.waitKey(1) & 0xFF == ord('q'):
37         break
38
39 cap.release()
40 cv2.destroyAllWindows()

```

#### D. Percobaan Deteksi Posisi Tangan Depan Dan Belakang

##### Program Dan Hasil Running Program



```

1  import cv2
2  import mediapipe as mp
3
4  cap = cv2.VideoCapture(0)
5
6  mp_hands = mp.solutions.hands
7  hands = mp_hands.Hands(max_num_hands=1)
8  mp_draw = mp.solutions.drawing_utils
9
10 while True:
11     success, img = cap.read()
12     if not success:
13         break
14
15     imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
16     results = hands.process(imgRGB)
17
18     if results.multi_hand_landmarks and results.multi_handedness:
19         for idx, hand_landmarks in enumerate(results.multi_hand_landmarks):
20
21             mp_draw.draw_landmarks(img, hand_landmarks, mp_hands.HAND_CONNECTIONS)
22
23             # Ambil data tangan kanan/kiri
24             label = results.multi_handedness[idx].classification[0].label
25

```

```

26     thumb_tip = hand_landmarks.landmark[4]
27     pinky_tip = hand_landmarks.landmark[20]
28
29     # Logika depan / belakang
30     if label == "Right":
31         if thumb_tip.x < pinky_tip.x:
32             posisi = "DEPAN"
33         else:
34             posisi = "BELAKANG"
35     else: # Left hand
36         if thumb_tip.x > pinky_tip.x:
37             posisi = "DEPAN"
38         else:
39             posisi = "BELAKANG"
40
41     cv2.putText(img, posisi, org=(50, 50),
42                 cv2.FONT_HERSHEY_SIMPLEX, fontScale=1,
43                 color=(0, 255, 0), thickness=3)
44
45     cv2.imshow( winname: "Deteksi Depan-Belakang", img)
46
47     if cv2.waitKey(1) & 0xFF == ord('q'):
48         break
49

```