

Bahan Belajar OpenCV & MediaPipe (Panduan Praktis)

Deskripsi: Panduan ini ditujukan untuk pemula sampai menengah yang ingin belajar pemrosesan citra dan pelacakan/estimasi pose menggunakan OpenCV dan MediaPipe. Berisi penjelasan, langkah instalasi, contoh kode Python, latihan, dan mini-proyek.

Daftar Isi

1. Ringkasan & Tujuan
 2. Prasyarat
 3. Instalasi & Setup
 4. Dasar-dasar OpenCV
 5. Membaca & Menampilkan Gambar
 6. Manipulasi Warna & Konversi
 7. Resize, Crop, Rotate
 8. Thresholding & Blurring
 9. Edge Detection (Canny)
 10. Contours & Shape Detection
 11. Mengakses Webcam dan Video
 12. Dasar-dasar MediaPipe
 13. Arsitektur singkat
 14. Hands (deteksi & landmark)
 15. Face Mesh
 16. Pose
 17. Holistic
 18. Integrasi OpenCV + MediaPipe
 19. Menangkap frame OpenCV → MediaPipe → menggambar hasil
 20. Contoh: Deteksi tangan + teleoperasi (contoh tombol virtual)
 21. Contoh Project Mini
 22. Project 1: Counter jari sederhana (menghitung jari terangkat)
 23. Project 2: Pelacakan pose untuk push-up counter
 24. Project 3: Virtual mouse / klik menggunakan gesture
 25. Tips Debugging & Performance
 26. Latihan & Tugas
 27. Cheat Sheet perintah & fungsi penting
 28. Referensi & Sumber Belajar
-

1. Ringkasan & Tujuan

Setelah mempelajari materi ini, kamu bisa:

- Mengolah gambar dasar (OpenCV)
- Menggunakan model real-time MediaPipe untuk mendeteksi tangan, wajah, dan pose
- Menggabungkan output MediaPipe ke aplikasi berbasis webcam
- Membangun project kecil seperti penghitung jari atau kontrol gesture

2. Prasyarat

- Python 3.8+ (direkomendasikan 3.10 atau 3.11)
- PIP
- Pengetahuan dasar Python (fungsi, list, loop)

3. Instalasi & Setup

Rekomendasi: buat virtual environment.

```
python -m venv venv
# windows
venv\Scripts\activate
# linux / mac
source venv/bin/activate

pip install --upgrade pip
pip install opencv-python mediapipe numpy
```

Catatan: - `opencv-python` cukup untuk banyak kasus; jika butuh fitur tambahan (FFmpeg, contrib) pakai `pip install opencv-contrib-python`. - Untuk komputer dengan GPU khusus atau kebutuhan lain, instal library tambahan sesuai kebutuhan.

4. Dasar-dasar OpenCV

Membaca & menampilkan gambar

```
import cv2
img = cv2.imread('gambar.jpg')
cv2.imshow('Gambar', img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Konversi warna (BGR ↔ RGB / grayscale)

```
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```

Resize, Crop, Rotate

```
resized = cv2.resize(img, (640, 480))
cropped = img[100:400, 50:300]
# rotate 90 deg
rotated = cv2.rotate(img, cv2.ROTATE_90_CLOCKWISE)
```

Blur & Threshold

```
blur = cv2.GaussianBlur(gray, (5,5), 0)
ret, th = cv2.threshold(gray, 127, 255, cv2.THRESH_BINARY)
```

Edge detection

```
edges = cv2.Canny(gray, 50, 150)
```

Contours

```
contours, _ = cv2.findContours(th, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
for c in contours:
    x,y,w,h = cv2.boundingRect(c)
    cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2)
```

Webcam (real-time)

```
cap = cv2.VideoCapture(0)
while True:
    ret, frame = cap.read()
    if not ret: break
    cv2.imshow('Webcam', frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

5. Dasar-dasar MediaPipe

MediaPipe menyediakan solusi siap-pakai (hands, pose, face_mesh, holistic). Contoh umum: deteksi tangan dengan landmark.

Contoh deteksi tangan (sederhana)

```
import cv2
import mediapipe as mp

mp_hands = mp.solutions.hands
mp_drawing = mp.solutions.drawing_utils

cap = cv2.VideoCapture(0)
with mp_hands.Hands(min_detection_confidence=0.5, min_tracking_confidence=0.5) as hands:
    while True:
        ret, frame = cap.read()
        if not ret: break
        # konversi BGR ke RGB
        image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        image.flags.writeable = False
        results = hands.process(image)
        image.flags.writeable = True
        image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

        if results.multi_hand_landmarks:
            for hand_landmarks in results.multi_hand_landmarks:
                mp_drawing.draw_landmarks(image, hand_landmarks,
                mp_hands.HAND_CONNECTIONS)

            cv2.imshow('Hands', image)
            if cv2.waitKey(1) & 0xFF == ord('q'):
                break
    cap.release()
    cv2.destroyAllWindows()
```

Penjelasan singkat landmark

- `results.multi_hand_landmarks` adalah list landmark per tangan.
- Landmark disimpan sebagai koordinat normalisasi (0..1) relatif ke ukuran gambar.

Modul lain

- `mp.solutions.pose` — pose estimation
- `mp.solutions.face_mesh` — mesh wajah (468 points)

- `mp.solutions.holistic` — gabungan face, pose, hands

6. Integrasi OpenCV + MediaPipe

Langkah umum: 1. Tangkap frame dengan OpenCV 2. Konversi BGR → RGB 3. Kirim ke MediaPipe (process) 4. Ambil hasil & gambar kembali di frame OpenCV

Contoh: menghitung jari terangkat (ide dasar) - Ambil landmark fingertip (mis. id 8 untuk index fingertip) - Bandingkan posisi y fingertip vs y pip joint untuk memutuskan terangkat atau tidak

7. Contoh Project Mini

Project 1: Counter jari sederhana

- Tujuan: Hitung berapa jari yang terangkat pada satu tangan
- Ide: Periksa posisi fingertip relatif ke joint di pangkal jari

Kode ringkas (pseudo):

```
# gunakan hand landmarks
fingers = []
# contoh untuk index: if tip.y < pip.y -> terangkat (koordinat y lebih kecil ke atas)
```

Latihan: implementasikan counter, tunjukkan angka pada frame.

Project 2: Push-up counter dengan pose

- Gunakan `mp.solutions.pose`
- Deteksi sudut siku (elbow) antara shoulder-elbow-wrist
- Hitung satu push-up ketika sudut turun melewati threshold lalu naik kembali

Project 3: Virtual mouse

- Gunakan posisi telunjuk sebagai kursor
- Lakukan smoothing (exponential moving average)
- Klik: deteksi gesture tertentu (mis. ibu jari menyentuh telunjuk)

8. Tips Debugging & Performance

- Kurangi resolusi input untuk meningkatkan FPS (mis. 640x480)
- Gunakan `image.flags.writeable = False` sebelum memanggil `process()` untuk performa

- Batasi `max_num_hands=1` jika hanya butuh 1 tangan
 - Gunakan `threading` jika `capture` dan `processing` membuat lag
-

9. Latihan & Tugas

1. Tulis script untuk mengambil foto dari webcam saat menekan tombol `s`.
 2. Buat filter warna: hanya tampilkan warna merah dalam frame.
 3. Implementasikan counter jari (0-5).
 4. Buat aplikasi kecil yang mengambil screenshot saat gesture tertentu dikenali.
-

10. Cheat Sheet (fungsi penting)

- OpenCV
 - `cv2.imread`, `cv2.imwrite`, `cv2.imshow`, `cv2.VideoCapture`
 - `cv2.cvtColor`, `cv2.resize`, `cv2.GaussianBlur`, `cv2.Canny`
 - `cv2.findContours`, `cv2.boundingRect`
 - MediaPipe
 - `mp.solutions.hands.Hands()`, `.process(image)`
 - `results.multi_hand_landmarks`, `mp_drawing.draw_landmarks`
-

11. Referensi & Sumber Belajar

- Dokumentasi OpenCV (python)
 - Dokumentasi MediaPipe (hands, pose, face_mesh)
 - Tutorial YouTube (search: "MediaPipe hands tutorial Python")
-

Catatan penutup

File ini dibuat sebagai bahan awal. Jika kamu mau, saya bisa: - Tambahkan file contoh `.py` terpisah untuk tiap topik - Buat slide ringkasan (PowerPoint) - Buat video singkat langkah-demi-langkah