



---

Nama: **Handayogi Tambunan (121140114)**

Tugas Ke: **5**

Mata Kuliah: **Sistem/ Teknologi Multimedia (IF4021)**

Tanggal: 24 Desember 2024

---

## 1 Drawing With Hands

Proyek ini memungkinkan pengguna untuk menggambar pada bingkai video secara real-time menggunakan gerakan tangan sebagai input. Proyek ini menggunakan Pustaka Python seperti OpenCV, MediaPipe, dan NumPy. Aplikasi ini mendeteksi gerakan tangan dan jari, yang memungkinkan pengguna untuk membuat dan memanipulasi garis secara langsung pada bingkai video. Selain itu, program ini menyertakan fitur untuk mengubah warna garis dan menghapus gambar menggunakan gerakan tangan tertentu. Program ini menawarkan cara yang interaktif dan intuitif untuk terlibat dengan pemrosesan video real-time melalui gerakan jari yang sederhana.

### 1.1 Latar Belakang

Dalam era digital saat ini, teknologi interaktif semakin menjadi bagian integral dari kehidupan sehari-hari. Salah satu bentuk teknologi interaktif yang berkembang pesat adalah pengenalan gerakan tangan untuk mengontrol perangkat dan aplikasi. Penggunaan gerakan tangan sebagai metode input memberikan pengalaman yang lebih alami dan intuitif dibandingkan dengan perangkat keras tradisional seperti mouse atau keyboard.

Proyek ini hadir sebagai solusi inovatif yang memanfaatkan teknologi pengenalan gerakan tangan untuk menggambar secara real-time pada bingkai video. Dengan menggunakan pustaka Python seperti OpenCV, MediaPipe, dan NumPy, aplikasi ini memungkinkan pengguna untuk menciptakan karya visual dengan hanya menggerakkan tangan mereka di depan kamera. Proyek ini tidak hanya mendeteksi gerakan tangan dan jari, tetapi juga menawarkan kemampuan untuk mengganti warna garis dan menghapus gambar melalui gerakan tertentu, sehingga menambah fleksibilitas dan keseruan dalam penggunaannya.

Pengembangan proyek ini didorong oleh kebutuhan akan alat kreatif yang lebih mudah diakses, interaktif, dan menyenangkan. Selain itu, aplikasi ini berpotensi untuk digunakan dalam berbagai bidang, termasuk pendidikan, seni digital, hiburan, dan terapi fisik, di mana interaksi berbasis gerakan tangan dapat meningkatkan efektivitas dan daya tarik suatu aktivitas.

Dengan mengintegrasikan teknologi terkini dan pendekatan intuitif, proyek ini diharapkan dapat memberikan kontribusi pada pengembangan teknologi pemrosesan video real-time yang lebih interaktif dan memudahkan masyarakat dalam berkreasi secara digital.

## 1.2 Teknologi yang dipakai

Proyek tersebut menggunakan beberapa teknologi utama untuk mendukung fungsionalitasnya, yaitu: OpenCV (Open Source Computer Vision Library)

1. OpenCV adalah pustaka Python yang populer untuk pemrosesan gambar dan video. Dalam proyek ini, OpenCV digunakan untuk menangkap dan memproses bingkai video secara real-time. OpenCV juga mempermudah penggambaran garis atau bentuk di atas bingkai video.
2. MediaPipe adalah framework yang dirancang untuk membangun pipeline pemrosesan multimedia. Dalam konteks proyek ini, MediaPipe digunakan untuk mendeteksi gerakan tangan dan jari secara real-time. Teknologi ini memungkinkan aplikasi untuk mengenali posisi dan orientasi tangan, yang menjadi dasar dari input pengguna.
3. NumPy adalah pustaka Python yang digunakan untuk komputasi numerik. Dalam proyek ini, NumPy digunakan untuk manipulasi data matriks dan array, yang berguna dalam pemrosesan data video atau koordinat tangan.
4. Kamera digunakan untuk menangkap video secara langsung, yang menjadi media utama di mana interaksi real-time dengan gerakan tangan dilakukan.
5. Algoritma Pendeteksi Gerakan Proyek ini mengintegrasikan algoritma pendeteksi gerakan yang dikembangkan berdasarkan kerangka MediaPipe untuk memahami posisi jari dan tangan, serta mengasosiasikan gerakan tertentu dengan fungsi seperti menggambar, menghapus, atau mengubah warna.

## 1.3 Cara Kerja Sistem

Proyek ini bekerja dengan memanfaatkan teknologi pemrosesan video real-time dan pengenalan gerakan tangan. Berikut adalah penjelasan langkah demi langkah cara kerja sistem:

1. Penerimaan Input Video
  - Sistem menggunakan kamera (misalnya, kamera laptop atau webcam) untuk menangkap video secara real-time.
  - Bingkai video yang diterima diproses sebagai data untuk mendeteksi gerakan tangan pengguna.
2. Pemrosesan Bingkai Video
  - Setiap bingkai video diambil oleh OpenCV untuk diolah.
  - Bingkai ini kemudian dikonversi ke format yang kompatibel untuk deteksi tangan menggunakan MediaPipe.
3. Deteksi Gerakan Tangan dan Jari
  - MediaPipe mendeteksi keberadaan tangan di dalam bingkai video dan memetakan landmark (titik-titik referensi) pada tangan dan jari.
  - Sistem mengenali posisi dan orientasi landmark tersebut untuk mengidentifikasi gerakan tertentu, seperti menggambar, menghapus, atau mengganti warna.
4. Pemetaan Gerakan ke Fungsi Gerakan tangan atau jari tertentu diasosiasikan dengan fungsi spesifik dalam aplikasi:

- Menggambar: Jika jari telunjuk terdeteksi dalam posisi tertentu, aplikasi menggambar garis mengikuti jalur gerakan jari tersebut.
- Menghapus: Jika gerakan tangan tertentu (misalnya, membuat lingkaran dengan jari) terdeteksi, sistem menghapus bagian tertentu dari gambar.
- Mengganti Warna: Sistem mengenali gestur lain (misalnya, dua jari diarahkan ke atas) untuk mengganti warna garis yang akan digambar.

## 5. Rendering Output

- Sistem menambahkan elemen visual, seperti garis atau bentuk, langsung pada bingkai video yang sedang ditampilkan.
- Bingkai video yang telah dimodifikasi ditampilkan kepada pengguna secara real-time di jendela aplikasi.

## 1.4 Penjelasan Code

### 1.4.1 function.py

Kode ini merupakan implementasi sederhana untuk mendeteksi dan memproses gerakan tangan menggunakan MediaPipe dan OpenCV. Fungsi utama yang tersedia meliputi menggambar garis, mengganti warna berdasarkan gerakan tertentu, dan menghapus gambar pada lapisan tertentu. Berikut adalah penjelasan lebih rinci dari setiap fungsi:

#### 1. Inisialisasi MediaPipe

```
1  
2 mp_hands = mp.solutions.hands  
3 mp_drawing = mp.solutions.drawing_utils
```

Kode 1: Inisialisasi MediaPipe

- `mp.solutions.hands`: Library MediaPipe digunakan untuk mendeteksi tangan dan landmark pada tangan.
- `mp.solutions.drawing_utils`: Digunakan untuk menggambar landmark pada frame video

#### 2. Fungsi draw.

```
1 def draw(layer, last_pos, curr_pos, color, thickness=6):  
2
```

Kode 2: Fungsi draw

Fungsi ini menggambar garis pada frame antara posisi terakhir tangan (last pos) dan posisi saat ini (curr pos).

#### Parameter

- `layer`: Lapisan tempat garis akan digambar.
- `last pos` dan `curr pos`: Titik koordinat tangan saat ini dan sebelumnya.
- `color`: Warna garis yang akan digambar dalam format RGB.
- `thickness`: Ketebalan garis.

#### Logika

- Garis akan digambar hanya jika kedua posisi (last dan current) tidak bernilai None.
- Fungsi mengembalikan koordinat curr pos sebagai last pos untuk iterasi berikutnya.

### 3. Fungsi change color

```
1 def change_color(landmarks, curr_color):
```

Kode 3: Change color

Fungsi ini mengganti warna garis berdasarkan jarak antara ujung jari manis dan ibu jari.

#### Parameter

- landmarks: Koordinat landmark tangan hasil deteksi MediaPipe.
- curr color: Warna saat ini yang digunakan.

#### Logika

- Jarak antara jari dihitung menggunakan rumus Euclidean:

```
1 distance = np.sqrt((thumb_tip.x - ring_finger_tip.x) ** 2 + (thumb_tip.y - ring_finger_tip.y)
    ** 2)
```

- Jika jarak  $< 0.05$ , warna berubah sesuai urutan: Putih  $\rightarrow$  Merah Merah  $\rightarrow$  Hijau Hijau  $\rightarrow$  Biru Biru  $\rightarrow$  Putih

### 4. Fungsi clear

```
1 def clear(landmarks, layer):
```

Kode 4: Fungsi clear

Fungsi ini menghapus semua gambar pada lapisan jika jari kelingking dan ibu jari bersentuhan.

#### Parameter

- landmarks: Koordinat landmark tangan hasil deteksi MediaPipe.
- layer: Lapisan gambar yang akan dihapus.

#### Logika

- Jarak antara jari dihitung dengan rumus Euclidean.
- Jika jarak  $< 0.05$ , lapisan di-reset menjadi kosong menggunakan:

```
1 return np.zeros_like(layer, dtype=np.uint8)
```

## 1.4.2 main.py

Kode ini adalah implementasi dari aplikasi Hand Gesture Drawing, di mana pengguna dapat menggambar, mengubah warna, atau menghapus layer gambar hanya dengan menggunakan gerakan tangan yang terdeteksi oleh kamera secara real-time. Aplikasi memanfaatkan library MediaPipe untuk deteksi tangan dan OpenCV untuk pengolahan gambar.

### 1. Inisialisasi MediaPipe dan Variabel Awal

```
1 mp_hands = mp.solutions.hands
2 hands = mp_hands.Hands(min_detection_confidence=0.7, min_tracking_confidence=0.2)
```

Kode 5: Inisialisasi MediaPipe dan Variabel Awal

- mp\_hands.Hands: Inisialisasi model deteksi tangan MediaPipe dengan tingkat kepercayaan minimum untuk deteksi (0.7) dan tracking (0.2).

```
1 last_pos = None
2 curr_color = (255, 255, 255)
```

- last pos: Menyimpan posisi terakhir ujung jari untuk menggambar garis kontinu.
- curr color: Warna saat ini untuk menggambar, diinisialisasi dengan warna putih (255, 255, 255).

## 2. Membuka Kamera

```
1 cap = cv2.VideoCapture(0)
2 _, frame = cap.read()
3 graff_layer = np.zeros_like(frame, dtype=np.uint8)
```

Kode 6: Membuka Kamera

- cv2.VideoCapture(0): Membuka kamera default untuk menangkap video.
- graff layer: Lapisan kosong dengan ukuran yang sama seperti frame video, digunakan untuk menggambar.

## 3. Loop Utama untuk Pemrosesan Real-Time

```
1 while cap.isOpened():
2     ret, frame = cap.read()
3     if not ret:
4         break
```

Kode 7: Loop Utama untuk Pemrosesan Real-Time

- Program akan membaca frame secara terus-menerus dari kamera hingga program dihentikan.

## 4. Pemrosesan Deteksi Tangan

```
1 rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
2 results = hands.process(rgb_frame)
```

Kode 8: Proses Deteksi Tangan

- Frame dikonversi dari BGR ke RGB agar kompatibel dengan MediaPipe.
- hands.process(): Memproses frame untuk mendeteksi tangan dan menghasilkan landmark.

## 5. Mengelola Gerakan Tangan Jika tangan terdeteksi:

```
1 for hand_landmarks in results.multi_hand_landmarks:
2     index_finger_tip = hand_landmarks.landmark[mp_hands.HandLandmark.INDEX_FINGER_TIP]
3     h, w, _ = frame.shape
4     curr_pos = (int(index_finger_tip.x * w), int(index_finger_tip.y * h))
```

Kode 9: Mengelola Gerakan Tangan

- Mendapatkan koordinat ujung jari telunjuk dalam piksel.

Fungsi yang dijalankan

- draw(): Menggambar garis antara posisi terakhir dan saat ini.
- change color(): Mengganti warna berdasarkan jarak ujung jari manis dan ibu jari.
- clear(): Menghapus layer jika jari kelingking dan ibu jari bersentuhan.

```
1 mp_drawing.draw_landmarks(frame, hand_landmarks, mp_hands.HAND_CONNECTIONS)
```

- Fungsi ini menggambar landmark dan koneksi antar jari pada frame untuk membantu visualisasi.

## 6. Menampilkan Output

```
1 output = cv2.addWeighted(frame, 0.8, graff_layer, 0.5, 0)
2 cv2.imshow('Drawing with Hands', output)
```

- `cv2.addWeighted()`: Menggabungkan frame asli dan layer gambar untuk ditampilkan sebagai output.

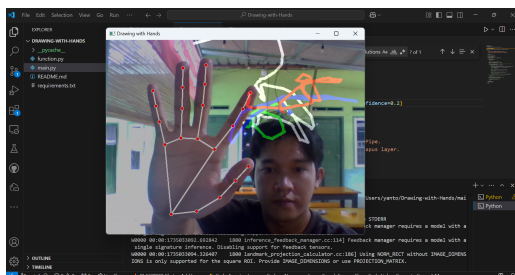
## 7. Penghentian Program

```
1 if cv2.waitKey(1) & 0xFF == ord('q'):
2     break
3 cap.release()
4 cv2.destroyAllWindows()
```

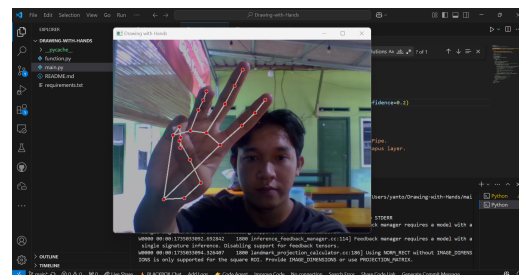
- Program berhenti jika tombol q ditekan. Kamera dilepas, dan semua jendela ditutup.

# 2 Dokumentasi Drawing with Hands

Berikut ini adalah dokumentasi dari proyek Drawing with Hands:



(a) Ganti Warna



(b) Hapus Tulisan

Gambar 1: Dokumentasi Drawing with Hands