



---

Nama: Khoirul Rijal Wicaksono (122140234), Sakinah Aulia (122140210) Tugas: Tugas Besar

Mata Kuliah: Multimedia (IF4021)

Tanggal: 31-05-24

---

## 1 Deskripsi Projek

filter interaktif menggunakan Python yang menciptakan suasana layaknya berada di stadion sepak bola. Sistem akan mendeteksi suara teriakan pengguna —jika volume atau frekuensinya tinggi, maka secara otomatis akan diputar suara sorakan supporter lain yang bergemuruh. Selain itu, filter juga mengenali gestur tangan seperti tepuk tangan dan sorakan tangan melalui kamera. Ketika gerakan tersebut terdeteksi, animasi penonton ikut bertepuk tangan disertai suara crowd.

## 2 Teknologi yang Digunakan pada Sistem

Pada proyek Filter Riu Stadion, digunakan beberapa library Python yang masing-masing memiliki peran spesifik untuk mengolah video, audio, serta deteksi gerakan. Berikut adalah penjelasan dari setiap library yang digunakan:

### 2.1 cv2 (OpenCV)

OpenCV adalah library untuk membaca, memproses, dan menganalisis gambar atau video. Dalam proyek ini, cv2 digunakan untuk:

- Mengakses kamera secara real-time.
- Membaca dan mengubah ukuran gambar latar (`stadion.jpg`, `teriak.jpg`, dll).
- Menampilkan output akhir berupa video dengan latar yang telah dimanipulasi.

**Contoh Penggunaan:**

```
1 img_idle = cv2.resize(cv2.imread('stadion.jpg'), (640, 480))
2 cv2.imshow("Filter Stadion", frame_out)
```

### 2.2 numpy

NumPy digunakan untuk perhitungan matematis, khususnya operasi array. Dalam proyek ini, digunakan untuk:

- Menghitung RMS (Root Mean Square) dari input suara.
- Membuat dan mengolah mask 3-channel untuk manipulasi gambar.

**Contoh Penggunaan:**

```
1 rms = np.sqrt(np.mean(indata**2))
2 mask_fg_3c = np.repeat(mask_fg[:, :, np.newaxis], 3, axis=2)
```

### 2.3 sounddevice

`sounddevice` menyediakan akses ke input/output audio secara real-time. Dalam proyek ini, digunakan untuk:

- Mendapatkan input suara dari mikrofon.
- Mendeteksi volume suara secara real-time dan mengubah status sistem berdasarkan intensitas suara.

**Contoh Penggunaan:**

```
1 with sd.InputStream(callback=audio_callback, channels=1, samplerate=44100):
2     ...
```

### 2.4 mediapipe

`MediaPipe` adalah framework open-source dari Google untuk pemrosesan video berbasis AI secara real-time. Dalam proyek ini, digunakan untuk dua hal utama:

- **Segmentasi tubuh:** untuk memisahkan pengguna dari latar belakang.
- **Deteksi tangan:** untuk mengidentifikasi gesture seperti tepuk tangan.

**Contoh Penggunaan:**

```
1 segmentor = mp_selfie.SelfieSegmentation(model_selection=0)
2 hands = mp_hands.Hands(max_num_hands=2, min_detection_confidence=0.7)
```

### 2.5 threading

`Threading` memungkinkan dua proses berjalan secara paralel. Dalam proyek ini, digunakan untuk:

- Input suara dan deteksi gesture dijalankan secara bersamaan tanpa mengganggu satu sama lain.

**Contoh Penggunaan:**

```
1 threading.Thread(target=audio_thread, daemon=True).start()
```

### 2.6 time

Library `time` merupakan library standar python, pada proyek ini digunakan untuk:

- Menghitung waktu sejak suara terakhir terdeteksi.
- Membuat delay singkat untuk stabilitas suara dan deteksi gesture.

**Contoh Penggunaan:**

```
1 now = time.time()
2 time.sleep(0.1)
```

## 2.7 pygame

Pygame adalah library yang biasa digunakan untuk membuat game, namun dalam proyek ini digunakan hanya untuk:

- Memutar file audio seperti suara sorakan (**teriak.wav**) dan tepuk tangan (**tepunktangan.wav**).

**Contoh Penggunaan:**

```
1 pygame.mixer.init()
2 sound_cheer = pygame.mixer.Sound('teriak.wav')
3 sound_cheer.play(loops=-1)
```

## 3 Cara Menjalankan Program dan Penggunaan Filter Riuhan Stadion

### 3.1 Langkah Menjalankan Program

- Buka file **stadion.ipynb**.
- Jalankan program secara berurutan, atau bisa langsung run all.
- Jendela kamera akan muncul dengan latar virtual.



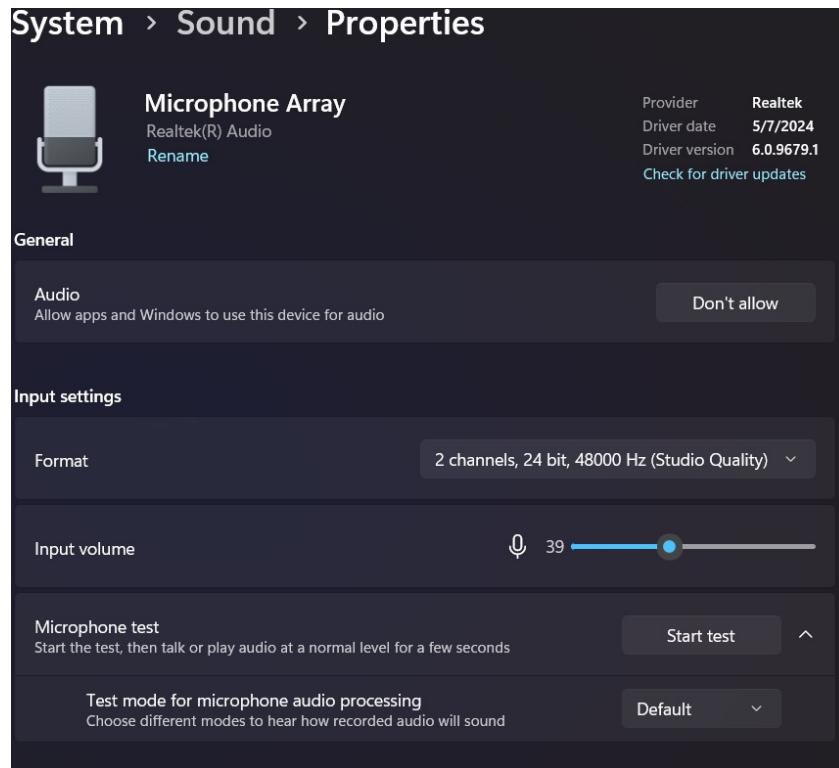
*Gambar: Tampilan awal filter stadion*

- Tekan tombol **q** pada jendela kamera untuk keluar.

### 3.2 Cara Penggunaan Program

#### 3.2.1 Deteksi Sorakan Melalui Suara

- Pastikan mikrofon aktif dan berada dalam jarak 30–50 cm dari wajah.
- Atur input volume mikrofon pada rentang 30–40% dari pengaturan sistem agar deteksi suara lebih akurat.



Gambar: Contoh pengaturan volume mikrofon

- Program akan merespons jika suara melebihi ambang batas ( $\sim -20$  dB).
- Untuk memicu efek sorakan:
  - Teriakkan secara **berulang dan penuh semangat**, misalnya:  
"GOAL! GOAL! GOAL!"



Gambar: Tampilan efek sorakan

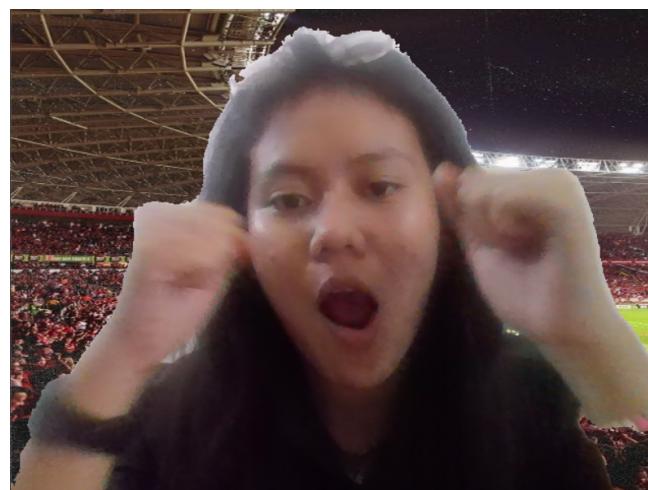
- Semakin emosional dan heboh, semakin realistik efek yang ditampilkan.

### 3.2.2 Deteksi Gesture Tangan

- Program mendeteksi keberadaan **tangan apa pun di depan kamera**.
- Angkat satu atau dua tangan, bertepuk tangan, atau lakukan gerakan sorakan tangan seperti mengangkat dan mengepalkan tangan.
- Saat gesture tangan terdeteksi:
  - Latar belakang akan berubah menyesuaikan kondisi penonton yang borsorak/bertepuk tangan.
  - Efek audio tepuk tangan otomatis diputar selama tangan masih terlihat.
- **Tidak perlu tepuk tangan secara sempurna**, karena program mendeteksi keberadaan tangan, bukan suara tepuknya.



Gambar: Tampilan saat tangan terdeteksi



Gambar: Tampilan selebrasi dengan gesture tangan

### 3.3 Sumber Background Filter

#### 3.3.1 Background Suara (Sound Filter)

- <https://vt.tiktok.com/ZSk89GUDW/>
- <https://vt.tiktok.com/ZSk89gDdF/>

### 3.3.2 Background Latar (Visual Filter)

- <https://id.pinterest.com/pin/102738435243915039/>
- <https://id.pinterest.com/pin/59602395053574365/>
- <https://id.pinterest.com/pin/488851734574077692/>

## 4 Penjelasan Kode Python: Riuhan Stadion

### 4.1 Import Library

```

1 import cv2
2 import numpy as np
3 import sounddevice as sd
4 import mediapipe as mp
5 import threading
6 import time
7 import pygame

```

Kode ini mengimpor berbagai library yang digunakan untuk pengolahan video, audio, threading, dan tampilan antarmuka.

### 4.2 Memuat Gambar Background

```

1 IMG_SIZE = (640, 480)
2 img_idle = cv2.resize(cv2.imread('stadion.jpg'), IMG_SIZE)
3 img_cheer = cv2.resize(cv2.imread('teriak.jpg'), IMG_SIZE)
4 img_clap = cv2.resize(cv2.imread('tepuktangan.jpg'), IMG_SIZE)

```

Ketiga gambar ini akan digunakan sebagai latar belakang virtual tergantung status pengguna: diam, bersorak, atau tepuk tangan.

### 4.3 Variabel Status

```

1 status_lock = threading.Lock()
2 audio_status = "idle"
3 gesture_status = "idle"
4 current_status = "idle"

```

Digunakan untuk menyimpan status input pengguna, yang dikontrol dengan ‘threading.Lock()’ agar thread tidak bertabrakan.

### 4.4 Inisialisasi MediaPipe dan Pygame

```

1 mp_selfie = mp.solutions.selfie_segmentation
2 segmentor = mp_selfie.SelfieSegmentation(model_selection=0)
3 mp_hands = mp.solutions.hands
4
5 pygame.mixer.init()
6 sound_cheer = pygame.mixer.Sound('teriak.wav')
7 sound_clap = pygame.mixer.Sound('tepuktangan.wav')

```

MediaPipe digunakan untuk segmentasi tubuh dan deteksi tangan. Pygame digunakan untuk memainkan suara.

## 4.5 Debounce Audio dan Gesture

```

1 last_active_time = 0
2 debounce_time = 1.0
3
4 clap_sound_playing = False
5 last_hand_seen_time = 0
6 hand_timeout = 0.5

```

Debounce menghindari input palsu yang muncul karena delay atau noise dengan memberi jeda sebelum status berubah kembali ke **idle**.

## 4.6 Fungsi Pembaruan Status Audio

```

1 def update_status_audio(new_status):
2     ...

```

Memainkan suara sorakan jika deteksi suara melebihi ambang batas, dan menghentikannya saat kembali tenang.

## 4.7 Fungsi Pembaruan Status Gesture

```

1 def update_status_gesture(new_status):
2     ...

```

Memainkan suara tepuk tangan saat tangan terdeteksi, dan menghentikannya jika tidak ada tangan yang terdeteksi.

## 4.8 Kalkulasi Status Akhir

```

1 def compute_final_status():
2     ...

```

Menentukan status akhir berdasarkan prioritas: gesture > suara > idle.

## 4.9 Callback Audio Input

```

1 def audio_callback(indata, frames, time_, status):
2     ...

```

Mengukur volume input suara dan mengonversinya ke dB. Jika melebihi ambang, maka dianggap bersorak.

## 4.10 Thread Input Suara

```

1 def audio_thread(device_index=None):
2     ...

```

Thread yang berjalan terus-menerus membaca input suara secara real-time.

#### 4.11 Virtual Background

```
1 def apply_virtual_background(frame, bg_image):
2     ...
```

Mengganti latar belakang gambar sesuai dengan status dan bantuan segmentasi tubuh dari MediaPipe.

#### 4.12 Loop Tampilan Webcam dan Gesture

```
1 def display_loop():
2     ...
```

Menampilkan webcam dan mendeteksi gesture menggunakan MediaPipe. Latar belakang berubah sesuai status pengguna.

#### 4.13 Program Utama

```
1 if __name__ == "__main__":
2     ...
```

Menjalankan thread audio dan loop utama tampilan webcam. Juga menampilkan daftar perangkat audio yang tersedia.

### 5 Hasil Analisis Program Filter Riuhan Stadion

#### 5.1 Fungsionalitas Program

Program ini merupakan filter interaktif yang merespons input suara (teriakan seperti euforia saat menonton pertandingan sepak bola) dan gesture tangan (tepuk tangan atau sorakan tangan) untuk menampilkan suasana stadion secara visual dan audio. Program menampilkan latar belakang stadion dan memutar efek suara sesuai input pengguna.

#### 5.2 Responsif terhadap Input

Responsifitas terhadap input suara cukup baik, dengan ambang batas -20 dB untuk mengenali sorakan. Sistem menggunakan teknik debounce agar tidak terlalu sensitif terhadap fluktuasi suara. Untuk input visual, gesture tangan dideteksi menggunakan MediaPipe dengan toleransi waktu timeout untuk memastikan deteksi stabil.

#### 5.3 Struktur Kode

Struktur kode dibagi menjadi beberapa fungsi utama:

- Deteksi dan penanganan input suara secara real-time (menggunakan thread).
- Deteksi gesture tangan melalui webcam.
- Penentuan status akhir (cheer, clap, atau idle).
- Penggantian latar belakang dengan segmentasi manusia.
- Sinkronisasi status dan pengelolaan thread menggunakan mekanisme lock.

## 5.4 Penggunaan Teknologi

Program memanfaatkan pustaka modern seperti:

- OpenCV untuk pemrosesan gambar dan tampilan webcam.
- NumPy untuk perhitungan suara.
- MediaPipe untuk deteksi gesture dan segmentasi manusia.
- SoundDevice untuk input audio.
- Pygame untuk pemutaran file audio looping.
- Threading dan sinkronisasi Python untuk manajemen paralel antar input.

## 5.5 Tampilan Visual

Tampilan antarmuka berbasis jendela OpenCV berukuran 640x480 piksel, menampilkan latar belakang stadion yang berubah dinamis berdasarkan input. Pengguna dapat melihat dirinya dengan latar virtual yang telah disesuaikan.

## 5.6 Pengelolaan Sumber Daya

Penggunaan webcam dan mikrofon dijalankan dalam thread terpisah, sehingga program tetap responsif. Penggunaan status global dan locking memastikan tidak terjadi konflik antar thread. Audio looping diberhentikan atau diganti dengan tepat ketika status berubah.

## 5.7 Kelebihan

- Responsif terhadap kombinasi input audio dan visual.
- Menggunakan segmentasi latar belakang yang cukup akurat.
- Interaktif dan mendukung ekspresi euforia penonton.
- Struktur modular memudahkan pengembangan.

## 5.8 Kekurangan

- Tidak ada sistem GUI yang ramah pengguna, hanya berbasis jendela OpenCV.
- Tidak menyimpan data aktivitas pengguna.
- Tidak ada opsi pengaturan sensitivitas suara atau gesture dari antarmuka.
- Ketergantungan pada pencahayaan dan kualitas kamera untuk deteksi gesture.

## 5.9 Kesimpulan

Program ini dapat digunakan untuk meningkatkan pengalaman menonton pertandingan dengan menciptakan suasana stadion secara virtual. Deteksi sorakan dan gesture tangan membuat program terasa imersif (Pengalaman yang membuat pengguna merasa hadir secara nyata di dalam suatu dunia virtual atau simulasi) dan menyenangkan. Cocok digunakan pada kegiatan nobar atau hiburan interaktif.