



---

Mata Kuliah: **Multimedia (IF4021)**

Tugas : **Tugas Besar**

Nama Anggota:

1. Bayu Agaluh Wijaya (121140097)
  2. Bima Setiawan Sandi (121140162)
  3. Fatkhan Aziez Suffi (120140181)
- 

## 1 Pendahuluan

Dalam era digital saat ini, teknologi interaktif sangat diminati oleh masyarakat, terutama generasi muda atau generasi Z. Salah satu bentuk teknologi interaktif yang menarik adalah filter berbasis wajah yang mampu memberikan pengalaman menyenangkan bagi pengguna. Filter semacam ini sering ditemukan pada media sosial, terutama Tiktok. Aplikasi Tiktok banyak diminati oleh banyak orang, terutama remaja atau generasi Z, yaitu seseorang yang lahir di kisaran tahun 1995-2010.[1]

Teknologi seperti Python, OpenCV, dan Mediapipe telah membuka peluang besar untuk menciptakan aplikasi yang mampu melacak dan mengenali ekspresi wajah secara real-time. Computer Vision (CV) adalah teknologi yang membuat komputer dapat melihat dan mengenali bentuk yang meniru otak dan mata manusia. Sedangkan, mediapipe adalah sebuah kerangka kerja yang dikembangkan oleh Google untuk membangun rangkaian proses dalam memproses data persepsi, baik itu dalam format audio maupun video. Teknologi ini memungkinkan pengembangan filter yang canggih, cepat, dan responsif terhadap berbagai ekspresi pengguna.[2]

Salah satu proyek yang dibuat dengan menggunakan bahasa Python, OpenCV, dan Mediapipe adalah Filter Emote Challenge. Filter ini merupakan pengenalan ekspresi wajah untuk meniru emoji yang bertujuan untuk menghibur pengguna.[3]

## 2 Implementasi

Pada bagian ini, akan dijelaskan langkah-langkah implementasi proyek pembuatan filter *Emote Challenge* menggunakan teknologi Python, OpenCV, dan Mediapipe. Implementasi ini meliputi pengaturan lingkungan pengembangan, pembuatan algoritma untuk pelacakan wajah, hingga penerapan filter ekspresi yang interaktif.

### 2.1 Persiapan Lingkungan Pengembangan

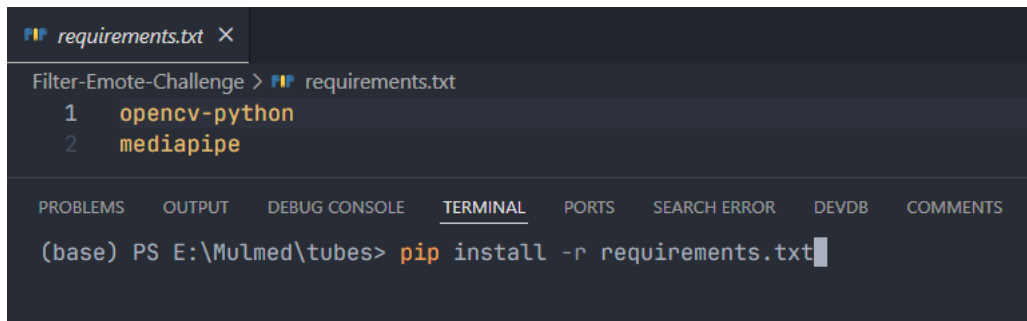
Langkah awal dalam implementasi adalah menyiapkan perangkat lunak dan *library* yang dibutuhkan. Berikut adalah spesifikasi yang dibutuhkan dalam proyek ini:

1. Bahasa Pemrograman Python  
Bahasa yang digunakan pada proyek ini adalah Python yang merupakan bahasa inti dalam pengembangan filter.
2. *Library* OpenCV  
Pustaka OpenCV digunakan untuk mengelola gambar dan melacak wajah pengguna.

### 3. Mediapipe

Pustaka Mediapipe digunakan untuk deteksi fitur wajah dan pengenalan ekspresi secara *real-time*.

Proses instalasi pustaka yang terdapat pada requirements.txt dapat dilakukan dengan menjalankan perintah pada PowerShell seperti berikut:



```
requirements.txt X
Filter-Emote-Challenge > requirements.txt
1  opencv-python
2  mediapipe

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR  DEVDB  COMMENTS

(base) PS E:\Mulmed\tubes> pip install -r requirements.txt
```

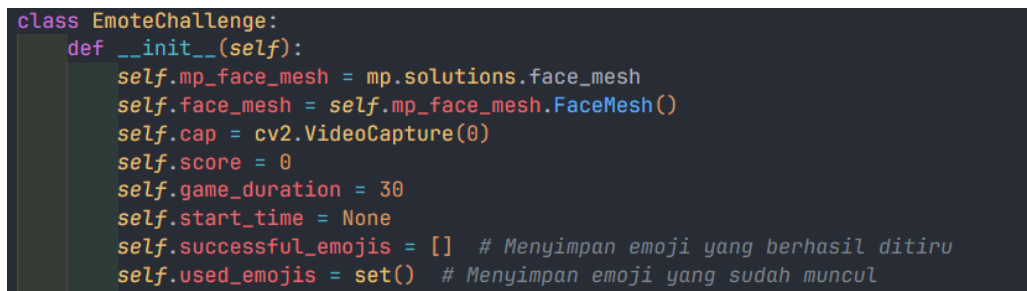
Gambar 1: Instalasi Pustaka

## 2.2 Pembuatan Algoritma Deteksi Wajah

Untuk Mendeteksi wajah pengguna secara *real-time* dapat menggunakan fitur *Face Mesh* yang disediakan dari pustaka Mediapipe, dan berikut adalah tahapan implementasi:

### 1. Inisialisasi *Face Mesh*

Mediapipe menyediakan model *pre-trained* atau model yang telah dilatih sebelumnya pada dataset tertentu sehingga dapat langsung digunakan untuk tugas spesifik, seperti pelacakan wajah, tanpa perlu melatih ulang dari awal. Model ini diinisialisasi untuk mendeteksi titik-titik penting pada wajah.



```
class EmoteChallenge:
    def __init__(self):
        self.mp_face_mesh = mp.solutions.face_mesh
        self.face_mesh = self.mp_face_mesh.FaceMesh()
        self.cap = cv2.VideoCapture(0)
        self.score = 0
        self.game_duration = 30
        self.start_time = None
        self.successful_emojis = [] # Menyimpan emoji yang berhasil ditiru
        self.used_emojis = set() # Menyimpan emoji yang sudah muncul
```

Gambar 2: Inisialisasi *Face Mesh*

### 2. Pelacakan Titik Wajah

Algoritma akan membaca input dari kamera dan melacak posisi fitur wajah seperti mata, hidung, dan bibir.

```
def detect_expression(self, face_landmarks):  
    # Mengambil landmark yang diperlukan  
    landmarks = face_landmarks.landmark  
  
    # Landmark untuk bibir, mata, dan alis  
    left_lip_corner = landmarks[61]  
    right_lip_corner = landmarks[291]  
    upper_lip = landmarks[13]  
    lower_lip = landmarks[14]  
    left_eye_top = landmarks[159]  
    left_eye_bottom = landmarks[145]  
    right_eye_top = landmarks[386]  
    right_eye_bottom = landmarks[374]
```

Gambar 3: Fungsi Untuk Menetapkan *Landmark* Wajah

### 3. Pengelompokan Ekspresi Wajah

Berdasarkan posisi relatif dari titik-titik wajah, filter akan mengidentifikasi ekspresi wajah yang sesuai dengan emoji.

```
# Deteksi ekspresi berdasarkan threshold  
if lip_width > thresholds["smile_width"] and lip_height > thresholds["smile_height"]:  
    print("Detected expression: emoji5 (senyum lebar)")  
    return "emoji5" # Emoji senyum lebar hingga terlihat gigi  
elif (left_eye_height < thresholds["eye_closed"] or right_eye_height < thresholds["eye_closed"]) and lip_height > thresholds["mouth_open"]:  
    print("Detected expression: emoji2 (mata mendedip 1 dan lidah melet)")  
    return "emoji2" # Emoji mata mendedip 1 dan lidah melet  
elif lip_height > thresholds["mouth_open"] and lip_width < thresholds["smile_width"]:  
    print("Detected expression: emoji11 (bengong)")  
    return "emoji11" # Emoji bengong hingga mulut terbuka  
elif (left_gradient < thresholds["lip_corner_down"] or right_gradient < thresholds["lip_corner_down"]):  
    print("Detected expression: emoji14 (sedih/murung)")  
    return "emoji14" # Emoji sedih/murung  
elif (lip_height < thresholds["mouth_open"] and lip_width < thresholds["smile_width"] and  
    left_gradient > thresholds["lip_corner_down"] and left_gradient < thresholds["lip_corner_up"] and  
    right_gradient > thresholds["lip_corner_down"] and right_gradient < thresholds["lip_corner_up"]):  
    print("Detected expression: emoji3 (ekspresi netral)")  
    return "emoji3" # Emoji ekspresi netral  
  
print("No expression detected")  
return None # Tidak ada ekspresi yang terdeteksi
```

Gambar 4: Fungsi Untuk Deteksi Ekspresi Dengan *Thresholds*

## 2.3 Penerapan Filter Ekspresi

Setelah wajah berhasil dilacak, filter ekspresi diterapkan dengan langkah berikut:

1. Pemetaan Emoji ke Ekspresi Wajah  
Setiap ekspresi wajah, seperti tersenyum, marah, atau terkejut, dikaitkan dengan representasi emoji yang relevan.
2. Overlay Emoji pada Layar  
Proses ini dilakukan secara real-time menggunakan pustaka OpenCV, pada layar saat permainan dimulai emoji akan ditampilkan dan harus ditiru pengguna untuk mendapatkan skor saat berhasil.

## 2.4 Pengujian dan Optimasi

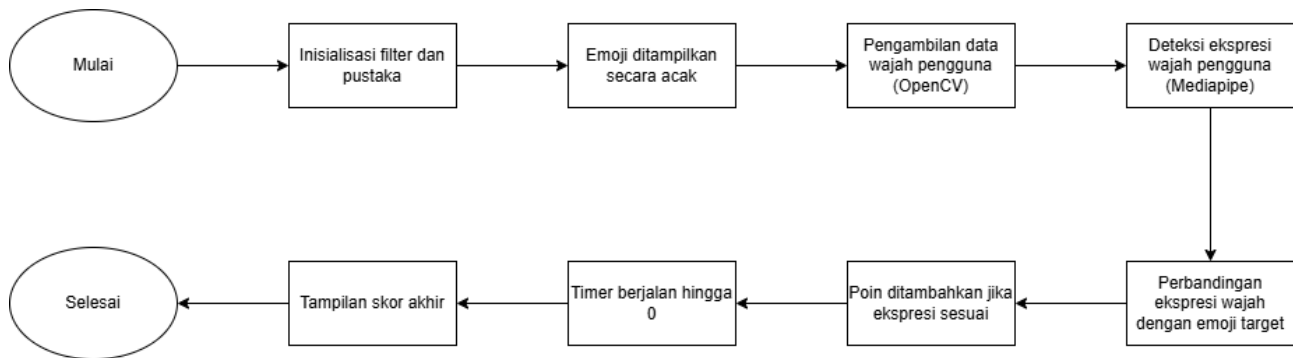
Setelah filter selesai dibuat, dilakukan pengujian untuk memastikan:

- Deteksi wajah bekerja dengan cepat dan akurat.
- Ekspresi yang dikenali sesuai dengan hasil yang diharapkan.
- Kinerja aplikasi stabil pada perangkat dengan spesifikasi standar.

Hasil pengujian menunjukkan bahwa teknologi Python, OpenCV, dan Mediapipe mampu memberikan kinerja yang baik dalam penerapan filter interaktif ini.

### 3 Alur Proses

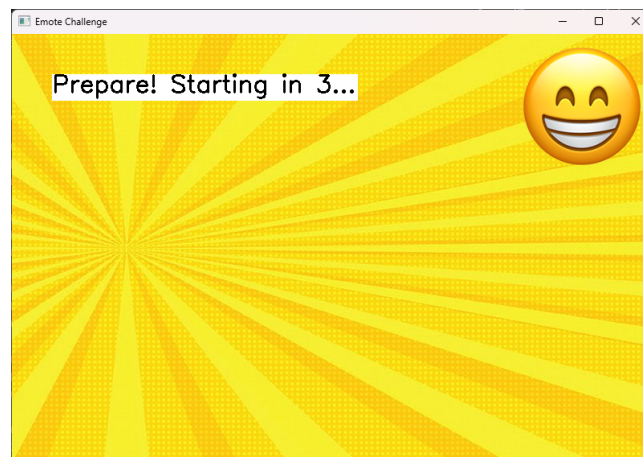
Proses kerja proyek Filter Emote Challenge diawali dengan penggunaan Python, OpenCV, dan Mediapipe untuk mengolah data video secara real-time, yang ditangkap melalui kamera. Data wajah pengguna kemudian dianalisis untuk mendeteksi ekspresi yang sesuai dengan emoji yang ditampilkan pada layar. Setiap tahapan melibatkan proses pemilihan emoji secara acak, pelacakan landmark wajah, dan perbandingan ekspresi untuk memastikan kesesuaian antara wajah pengguna dan emoji target. Pada bagian ini, akan dijelaskan langkah-langkah dari penampilan emoji hingga evaluasi skor pengguna.



Gambar 5: Diagram Alur Pemrosesan Filter

#### 3.1 Inisialisasi Filter

Sistem dimulai dengan memuat pustaka Python yang diperlukan, termasuk OpenCV untuk pengolahan gambar dan Mediapipe untuk pelacakan fitur wajah. Timer juga diinisialisasi dengan durasi 30 detik untuk mengatur waktu permainan.



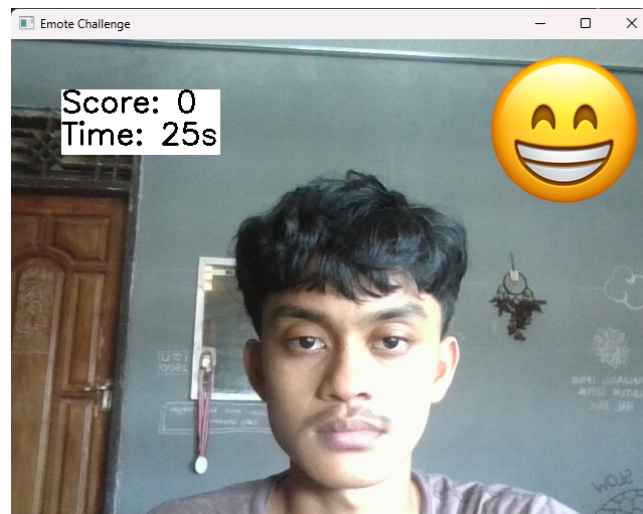
Gambar 6: Inisialisasi Filter

#### 3.2 Penampilan Emoji Secara Acak

Emoji akan ditampilkan secara acak pada layar menggunakan fungsi randomizer. Emoji ini menjadi target ekspresi yang harus ditiru oleh pengguna selama permainan.

#### 3.3 Pengambilan Data Wajah

Sistem menangkap data wajah pengguna secara real-time melalui kamera. Frame video yang diperoleh diolah menggunakan OpenCV untuk mendeteksi wajah.



Gambar 7: Menangkap data wajah pengguna

### 3.4 Deteksi Ekspresi Wajah

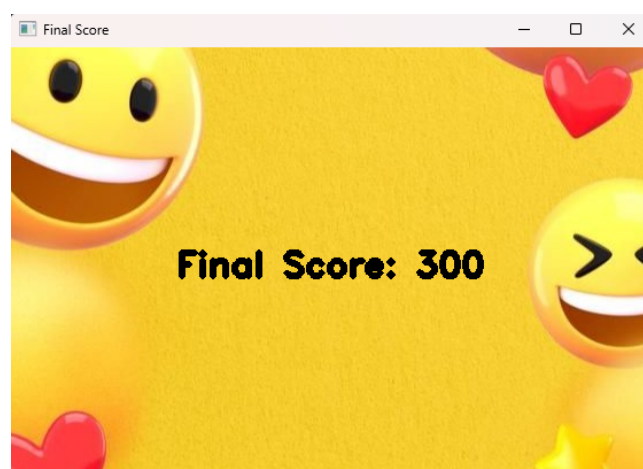
Mediapipe digunakan untuk melacak landmark pada wajah pengguna, seperti mata, hidung, dan mulut. Berdasarkan data landmark, sistem menganalisis ekspresi wajah pengguna untuk mendeteksi apakah ekspresi tersebut sesuai dengan emoji target. Setelah ekspresi wajah pengguna dianalisis, sistem membandingkannya dengan ekspresi emoji yang ditampilkan. Jika sesuai, pengguna akan mendapatkan poin sebesar 100.

### 3.5 Perhitungan Skor dan Timer

Setiap kali pengguna berhasil meniru ekspresi dengan benar, skor akan bertambah 100. Sistem juga terus mengurangi waktu permainan hingga timer mencapai 0, menandakan permainan berakhir.

### 3.6 Tampilan Hasil Akhir

Setelah permainan selesai, sistem menampilkan total skor pengguna bersama dengan laporan akurasi peniruan ekspresi wajah.



Gambar 8: Instalasi Pustaka

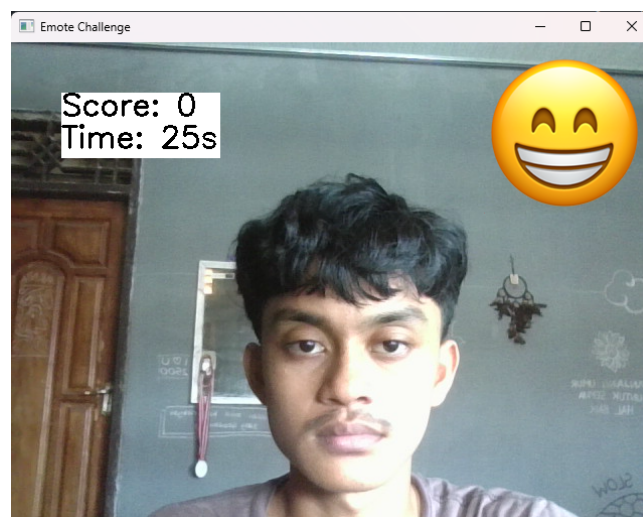
## 4 Hasil dan Pembahasan

Setelah melakukan implementasi dan pengujian pada program Emote Challenge, beberapa hasil yang telah diperoleh. Program berhasil mendeteksi berbagai ekspresi wajah pengguna berdasarkan landmark wajah yang dihasilkan oleh MediaPipe. Ekspresi yang berhasil dideteksi meliputi senyum lebar, mata mengedip dan lidah melet, bengong, sedih/murung, ekspresi netral, cium dengan mata mengedip, dan muka monyet senyum. Deteksi ekspresi ini dilakukan dengan menghitung jarak dan gradien antara berbagai landmark wajah seperti bibir, mata, dan alis, serta membandingkannya dengan threshold yang telah ditentukan.

Selain itu, emoji yang sesuai dengan ekspresi wajah pengguna berhasil ditampilkan di sudut kanan atas layar. Gambar emoji dimuat dari direktori assets dan ditampilkan menggunakan fungsi overlay emoji. Program juga memastikan bahwa setiap emoji hanya muncul sekali selama permainan berlangsung, dan permainan berhenti secara otomatis ketika semua emoji telah berhasil ditiru atau waktu permainan habis.

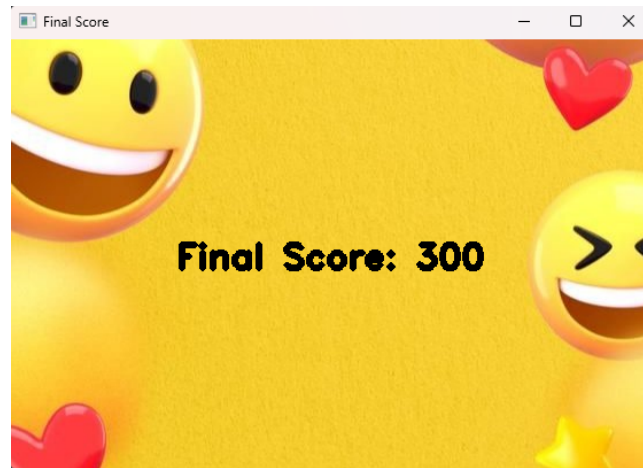
Skor pengguna dihitung berdasarkan jumlah ekspresi yang berhasil ditiru dengan benar, dengan setiap ekspresi yang berhasil ditiru menambah 100 poin pada skor. Permainan berlangsung selama 30 detik, dan waktu yang tersisa ditampilkan di layar untuk memberikan tantangan yang cukup bagi pengguna. Setelah permainan selesai, skor akhir ditampilkan untuk memberikan umpan balik kepada pengguna tentang kinerja mereka selama permainan.

Saran untuk para pengembang project ini kedepannya agar logika deteksi ekspresi dapat dibuat lebih sensitif dengan menambahkan lebih banyak faktor pendeteksi untuk setiap emoji. Misalnya, selain menggunakan jarak dan gradien, dapat ditambahkan analisis tekstur wajah atau perubahan warna kulit untuk mendeteksi ekspresi dengan lebih akurat. Menambahkan lebih banyak variasi ekspresi dan emoji untuk meningkatkan kompleksitas dan keseruan permainan. Ekspresi tambahan seperti marah, terkejut, atau tertawa terbahak-bahak dapat membuat permainan lebih menarik. Meningkatkan antarmuka pengguna dengan desain yang lebih menarik dan intuitif. Menambahkan animasi dan efek suara juga dapat meningkatkan pengalaman pengguna.



Gambar 9: Hasil Implementasi





Gambar 10: Hasil Implementasi

## 5 Kesimpulan

Proyek *Filter Emote Challenge* berhasil mengintegrasikan teknologi Python, OpenCV, dan Mediapipe untuk memproses filter interaktif berbasis pelacakan ekspresi wajah secara *real-time*. Sistem ini mampu menampilkan emoji secara acak, mendeteksi wajah pengguna, dan membandingkan kesesuaiannya dengan emoji target. Dengan adanya fitur penilaian otomatis dengan batas waktu 30 detik, aplikasi ini tidak hanya menghibur tetapi juga memberikan pengalaman interaktif yang responsif untuk pengguna. Proyek ini membuktikan potensi teknologi *Computer Vision* dalam menciptakan aplikasi aplikasi yang kreatif dan menarik untuk masyarakat, terutama generasi muda.

## References

- [1] O. V. N. Azizah and A. Zahid, “Identitas diri perempuan muslim generasi z (studi kasus pengguna tiktok mahasiswa febi 2019 uin satu tulungagung),” *Pute Waya: Sociology of Religion Journal*, vol. 4, no. 1, pp. 42–62, June 2023, received: 20 Maret 2023 / Accepted: 30 Mei 2023 / Published: 30 Juni 2023.
- [2] T. C. A.-S. Zulkhaidi, E. Maria, and Yulianto, “Pengenalan pola bentuk wajah dengan opencv,” *JURTI*, vol. 3, no. 2, pp. 181–186, December 2019, received June 1st, 2012; Revised June 25th, 2012; Accepted July 10th, 2012.
- [3] M. A. Pratama, M. R. Erfit, N. M. Farhani, I. A. Hartono, and M. Maryamah, “Klasifikasi abjad sibi (sistem bahasa isyarat indonesia) menggunakan mediapipe dengan metode deep learning,” *Seminar Nasional Sains Data 2023 (SENADA 2023)*, pp. 134–141, 2023. [Online]. Available: <https://arxiv.org/abs/2006.10214>