



Nama: **Hendry Kurniawan (118140094)**

Tugas Ke: **Final Project**

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

Tanggal: 24 Desember 2024

1 Deskripsi

Filter ini adalah aplikasi real-time yang menambahkan efek gambar katak ke wajah pengguna dengan menggunakan teknologi MediaPipe dan OpenCV. Menggunakan deteksi wajah, gambar katak disesuaikan dengan posisi dan pembukaan mulut pengguna, menciptakan efek interaktif. Dengan transparansi gambar dan ukuran yang menyesuaikan ekspresi wajah, aplikasi ini menawarkan pengalaman filter wajah yang halus dan personal. Proyek ini cocok untuk hiburan, media sosial, dan eksperimen kreatif, serta dapat dengan mudah dikembangkan untuk efek wajah lainnya. Untuk menggunakan header, anda cukup membuat `\section` pada bagian script anda. Penomoran pada header akan otomatis dibuat. Jika anda membutuhkan line break, anda dapat membubuhkan dua garis miring dengan backslash seperti berikut `\\`

2 Instalasi dan penggunaan

2.1 Persyaratan Sistem

Pastikan Anda memiliki perangkat dengan sistem operasi Windows, macOS, atau Linux, serta kamera untuk menangkap gambar real-time.

2.2 Instalasi Paket yang Dibutuhkan

Program ini menggunakan beberapa pustaka Python yang perlu diinstal terlebih dahulu. Ikuti langkah-langkah berikut untuk menginstalnya:

- Install Python

Pastikan Anda sudah menginstal Python (versi yang digunakan dalam pengembang filter ini adalah 3.12.4). Jika belum, Anda bisa mengunduhnya dari situs resmi Python.

- Install Pustaka yang dibutuhkan

Buka terminal atau command prompt dan jalankan perintah berikut untuk menginstal pustaka yang diperlukan: **`pip install mediapipe opencv-python numpy`** Pustaka yang diinstal: mediapipe: Untuk mendeteksi wajah dan landmark wajah. opencv-python: Untuk menangani pengolahan gambar dan video. numpy: Untuk manipulasi array dan gambar.

2.3 Mempersiapkan Gambar Katak

Siapkan gambar katak yang memiliki saluran alpha (transparansi). Pastikan gambar tersebut berformat PNG. Gambar katak dapat diunduh pada folder data, dengan nama katak.png Buat folder data, lalu simpan gambar katak didalamnya

2.4 Menjalankan Program

Setelah semua pustaka diinstal dan gambar hewan disiapkan, Anda dapat menjalankan program dengan langkah berikut:

- Menyiapkan kode program

Salin seluruh kode program yang ada pada file `main.ipynb` ke dalam sebuah file Python, misalnya `filter_suap.py`.

- Menjalankan Program

Buka terminal atau command prompt, arahkan ke folder tempat file Python disimpan, dan jalankan perintah berikut: **`python filter_suap.py`** Program akan membuka jendela kamera dan menampilkan efek gambar hewan yang mengikuti pergerakan mulut Anda.

2.5 Menggunakan Program

Filter Wajah: Saat program berjalan, Anda akan melihat gambar katak yang muncul di area mulut Anda dalam jendela kamera. Keluar dari Program: Untuk keluar dari program, tekan tombol ESC pada keyboard. Program akan berhenti dan menutup jendela kamera.

2.6 Troubleshooting

Jika kamera tidak terdeteksi, pastikan perangkat kamera Anda terhubung dengan baik dan aktif. Jika program tidak berjalan, pastikan semua pustaka telah terinstal dengan benar.

Dengan mengikuti instruksi ini, Anda akan dapat menjalankan program filter wajah dengan gambar hewan secara mudah dan menyenangkan!

3 Kode Program

3.1 Import Library

mediapipe: Digunakan untuk deteksi dan pelacakan wajah menggunakan FaceMesh untuk mendapatkan landmark wajah.

cv2: Pustaka OpenCV digunakan untuk menangani pengolahan gambar dan video, serta interaksi dengan kamera.

numpy: Digunakan untuk manipulasi array, yang sering digunakan dalam pengolahan gambar.

os: Digunakan untuk mengelola jalur file, terutama untuk mengakses gambar hewan yang akan ditambahkan ke wajah.

```
1 import mediapipe as mp
2 import cv2
3 import numpy as np
4 import os
5
```

Kode 1: import library

3.2 Inisialisasi Mediapipe FaceMesh

Mengakses modul FaceMesh dari pustaka MediaPipe yang digunakan untuk mendeteksi wajah dan landmarks dengan parameter yang akan di deteksi adalah gambar real-time, bukan gambar statis, hanya satu wajah yang akan di deteksi.

```

1 # Inisialisasi MediaPipe FaceMesh
2 mp_face_mesh = mp.solutions.face_mesh
3 face_mesh = mp_face_mesh.FaceMesh(static_image_mode=False, max_num_faces=1, refine_landmarks=True)
4

```

Kode 2: Inisialisasi Mediapipe FaceMesh

3.3 Penyesuaian Gambar yang akan ditempelkan ke wajah

Pada blok program ini dilakukan beberapa hal yaitu penentuan landmark posisi bibir atas dan bibir bawah pengguna, menghitung jarak buka mulut pengguna, penyesuaian ukuran gambar yang akan di tempelkan, serta penempatannya pada wajah.

```

1 # Fungsi untuk menempatkan gambar hewan pada wajah
2 def overlay_animal(image, landmarks, animal_image, scale_factor=0.4, max_height=200):
3     h, w, _ = image.shape
4     ah, aw, _ = animal_image.shape
5
6     # Landmark untuk mulut
7     top_lip = landmarks[13] # Landmark bibir atas
8     bottom_lip = landmarks[14] # Landmark bibir bawah
9
10    # Koordinat mulut pengguna
11    mouth_top = (int(top_lip.x * w), int(top_lip.y * h))
12    mouth_bottom = (int(bottom_lip.x * w), int(bottom_lip.y * h))
13    mouth_center = (mouth_top[0], (mouth_top[1] + mouth_bottom[1]) // 2)
14
15    # Hitung jarak pembukaan mulut
16    mouth_opening = abs(mouth_bottom[1] - mouth_top[1])
17
18    # Pastikan pembukaan mulut cukup besar untuk mengubah ukuran gambar
19    if mouth_opening < 5: # Ambang batas pembukaan mulut yang terlalu kecil
20        mouth_opening = 5 # Setel ke nilai minimum yang valid
21
22    # Sesuaikan tinggi gambar katak dengan pembukaan mulut secara vertikal
23    new_height = int(ah + mouth_opening * scale_factor) # Tinggi bertambah seiring pembukaan mulut
24    new_width = int(aw * scale_factor) # Lebar tetap, hanya tinggi yang berubah
25
26    # Batasi tinggi gambar katak agar tidak terlalu besar
27    new_height = min(new_height, max_height) # Batasi tinggi maksimal
28
29    # Pastikan ukuran baru tidak kurang dari 1
30    new_height = max(new_height, 1)
31    new_width = max(new_width, 1)
32
33    resized_frog = cv2.resize(animal_image, (new_width, new_height)) # Sesuaikan ukuran
34    ah, aw, _ = resized_frog.shape
35
36    # Tempelkan gambar katak di sekitar mulut
37    y_offset = mouth_center[1] - ah // 2
38    x_offset = mouth_center[0] - aw // 2
39
40    # Validasi koordinat agar tidak keluar batas
41    y1, y2 = max(0, y_offset), min(h, y_offset + ah)
42    x1, x2 = max(0, x_offset), min(w, x_offset + aw)
43
44    # Sesuaikan ukuran overlay agar sesuai dengan frame
45    overlay_y1, overlay_y2 = max(0, -y_offset), ah - max(0, (y_offset + ah) - h)
46    overlay_x1, overlay_x2 = max(0, -x_offset), aw - max(0, (x_offset + aw) - w)
47
48    for c in range(3): # Untuk setiap channel warna (RGB)
49        alpha = resized_frog[overlay_y1:overlay_y2, overlay_x1:overlay_x2, 3] / 255.0

```

```

50     image[y1:y2, x1:x2, c] = (
51         alpha * resized_frog[overlay_y1:overlay_y2, overlay_x1:overlay_x2, c]
52         + (1 - alpha) * image[y1:y2, x1:x2, c]
53     )
54
55     return image
56

```

Kode 3: Penyesuaian Ukuran Gambar Filter

3.4 Import Gambar

Import gambar dari folder data

```

1 # Path ke gambar hewan (katak)
2 animal_image_path = os.path.join(os.getcwd(), 'data', 'frog.png')
3
4 # Muat gambar hewan (katak dengan transparansi)
5 animal_image = cv2.imread(animal_image_path, cv2.IMREAD_UNCHANGED)
6

```

Kode 4: Import Gambar

3.5 Menjalankan Filter

Dilakukan perulangan/looping pada program, sehingga jika wajah terdeteksi, gambar katak ditempelkan ke mulut pengguna. Program akan berhenti jika pengguna menekan tombol "Esc"

```

1 # Mulai kamera
2 cap = cv2.VideoCapture(0)
3
4 while cap.isOpened():
5     success, frame = cap.read()
6     if not success:
7         print("Tidak dapat mengakses kamera.")
8         break
9
10    # Konversi gambar ke RGB
11    rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
12    results = face_mesh.process(rgb_frame)
13
14    # Jika wajah terdeteksi
15    if results.multi_face_landmarks:
16        for face_landmarks in results.multi_face_landmarks:
17            landmarks = face_landmarks.landmark
18            frame = overlay_animal(frame, landmarks, animal_image)
19
20    # Tampilkan hasil
21    cv2.imshow("Animal Filter", frame)
22
23    # Tombol 'ESC' untuk keluar
24    if cv2.waitKey(5) & 0xFF == 27:
25        break
26
27 cap.release()
28 cv2.destroyAllWindows()
29

```

Kode 5: Menjalankan Filter