



Nama: **Yusup Pandu Putra Wibowo (121140124)**  
Mata Kuliah: **Sistem / Teknologi Multimedia (IF4021)**

Tugas Ke: **Tugas Besar**  
Tanggal: 24 Desember 2024

## 1 Pendahuluan

### 1.1 Latar Belakang

Kemajuan teknologi yang begitu pesat telah mengubah berbagai aspek kehidupan manusia, termasuk dalam bidang komunikasi dan hiburan. Perkembangan teknologi ini menghadirkan berbagai inovasi yang mempermudah manusia dalam berinteraksi dan mengekspresikan diri. Salah satu dampak terbesar terlihat pada media sosial, di mana teknologi canggih memungkinkan pengguna untuk berkreasi dengan berbagai cara, termasuk menggunakan fitur-fitur seperti filter gambar. Hal ini mencerminkan peran besar teknologi dalam memberikan pengalaman visual yang lebih kaya dan interaktif bagi penggunanya[1].

Generasi saat ini, khususnya kalangan muda, sangat akrab dengan penggunaan filter di aplikasi media sosial seperti Instagram, Snapchat, dan TikTok. Filter ini memungkinkan pengguna untuk mengubah tampilan foto atau video secara instan, baik untuk tujuan estetika, hiburan, maupun personalisasi. Kebiasaan ini menunjukkan bagaimana kemajuan teknologi multimedia telah menjadi bagian dari keseharian generasi digital. Namun, ketergantungan pada filter juga membawa tantangan, seperti bagaimana tetap menjaga apresiasi terhadap keaslian visual di tengah maraknya penggunaan teknologi manipulasi gambar.

Hal ini tidak lepas dari kontribusi kemajuan teknologi multimedia yang terus berkembang. Dengan dukungan teknologi seperti augmented reality (AR), machine learning, dan kecerdasan buatan, filter digital kini tidak hanya menjadi alat dekoratif, tetapi juga mampu menghadirkan pengalaman interaktif yang lebih mendalam. Perkembangan ini memungkinkan filter untuk menjadi lebih realistis dan personal, memberikan pengaruh yang besar dalam dunia kreatif dan komunikasi digital modern. Dengan demikian, teknologi multimedia telah menjadi pendorong utama dalam menciptakan solusi visual yang tidak hanya menarik tetapi juga relevan untuk kebutuhan pengguna masa kini.

### 1.2 Maksud dan Tujuan

1. **Membuat filter yang dapat memanipulasi gambar:** Tujuan utama adalah merancang filter yang mampu mengubah tampilan gambar atau video secara kreatif, baik untuk keperluan estetika, hiburan, maupun edukasi. Filter ini dirancang agar mudah digunakan dan mampu memberikan pengalaman visual yang menarik bagi pengguna.
2. **Menguji kefokuskan serta keakuratan pengguna:** Melalui penggunaan filter, diharapkan pengguna dapat melatih kemampuan observasi mereka terhadap detail visual. Hal ini juga dapat menjadi sarana untuk mengevaluasi seberapa baik pengguna dapat beradaptasi dengan teknologi manipulasi visual dalam konteks kehidupan sehari-hari maupun profesional.

## 2 Deskripsi Program dan Algoritma

Filter "Reaction Time" adalah sebuah inovasi interaktif yang dirancang untuk menguji tingkat keakuratan dan kecepatan reaksi pengguna dalam menyelaraskan gerakan visual. Saat filter ini diaktifkan, layar akan menampilkan sebuah gambar yang terbagi menjadi dua bagian: sisi kiri yang bersifat statis (diam), dan sisi kanan yang bergerak dengan rotasi searah jarum jam. Tugas pengguna adalah menghentikan rotasi gambar pada sisi kanan dengan cara menganggukkan kepala pada momen yang dianggap paling tepat. Tujuan utamanya adalah menyelaraskan posisi gambar di sisi kanan dengan gambar statis di sisi kiri. Setelah pengguna melakukan tindakan tersebut, filter akan menghitung dan menampilkan tingkat keakuratan atau presisi pengguna dalam bentuk skor, sehingga pengguna dapat mengetahui sejauh mana kemampuan mereka dalam bereaksi secara cepat dan tepat.

### 2.1 Python

Python adalah bahasa pemrograman tingkat tinggi yang dirancang untuk kemudahan penggunaan dan pembacaan [2]. Bahasa ini memiliki sintaksis yang sederhana dan elegan, yang membuatnya sangat cocok untuk pemula. Python mendukung berbagai paradigma pemrograman, termasuk pemrograman berorientasi objek, fungsional, dan prosedural.

### 2.2 Numpy

NumPy adalah library Python yang digunakan untuk komputasi numerik dan manipulasi array[3]. Library ini sangat efisien dalam menangani array multidimensi, operasi matematika, dan manipulasi data dalam bentuk matriks.

### 2.3 OpenCV

OpenCV (Open Source Computer Vision) adalah library yang berfokus pada pemrosesan gambar dan video secara real-time[4]. OpenCV digunakan untuk aplikasi seperti deteksi objek, pengenalan wajah, pengolahan gambar, dan pengenalan pola.

### 2.4 MediaPipe

MediaPipe adalah library open-source dari Google yang dirancang untuk mempermudah pengembangan aplikasi berbasis Machine Learning dalam pengolahan video, audio, atau data lainnya. MediaPipe banyak digunakan untuk aplikasi real-time, seperti pelacakan tangan, deteksi wajah, dan pose tubuh[5].

### 2.5 time

time adalah library bawaan Python yang digunakan untuk bekerja dengan waktu, seperti pencatatan waktu, pengukuran durasi, atau penundaan proses.

## 3 Desain Filter

### 3.1 Textbox dan font

Desain textbox menggunakan perpaduan warna hijau dan hijau gelap, dipadukan dengan font berwarna putih, dirancang untuk memberikan pengalaman visual yang nyaman bagi pengguna. Kombinasi warna ini bertujuan untuk mengurangi ketegangan pada mata, sehingga pengguna dapat membaca instruksi dengan lebih mudah dan fokus tanpa merasa terganggu oleh elemen visual yang terlalu mencolok

### 3.2 Textbox dan font

Penggunaan gambar klub sepak bola yang diputar dirancang untuk memberikan kesan yang menarik dan dinamis. Pemilihan gambar klub sepak bola bukan hanya untuk meningkatkan daya tarik visual, tetapi juga karena sepak bola memiliki banyak penggemar di seluruh dunia. Dengan elemen yang familiar ini, pengguna akan lebih mudah untuk mempresisikan dari gambar yang berputar.

## 4 Analisis Hasil

Pada filter ini, pengerjaannya dibagi menjadi tiga **section**, yaitu:

### 4.1 Section 1: Instruksi

Bagian ini berisi instruksi yang ditampilkan dalam sebuah *textbox*, dilengkapi dengan *countdown timer* selama 5 detik di bawahnya. Setelah waktu habis, filter secara otomatis akan melanjutkan ke **section** berikutnya.

- **Hasil Pengujian:**

Timer dan instruksi berhasil muncul dengan baik, berjalan sesuai durasi yang ditentukan, dan secara otomatis melanjutkan ke **section** berikutnya tanpa kendala.

### 4.2 Section 2: Gambar Berputar

Pada bagian ini, gambar secara otomatis terbagi menjadi dua sisi: sisi kiri yang statis dan sisi kanan yang bergerak berputar. Pengguna diminta untuk menganggukkan kepala sebagai **trigger** untuk menghentikan pergerakan sisi kanan dan menghasilkan nilai **Akurasi**. Setelah itu, filter akan berlanjut ke **section** terakhir.

- **Hasil Pengujian:**

Proses pembagian gambar berjalan dengan baik, di mana sisi kiri tetap statis dan sisi kanan bergerak secara dinamis. Namun, **trigger** berupa deteksi anggukan belum berjalan dengan sempurna, sehingga memerlukan perbaikan lebih lanjut.

### 4.3 Section 3: Nilai Akurasi

Bagian ini menampilkan hasil berupa persentase nilai **Akurasi** yang telah dihitung berdasarkan interaksi pengguna di **section** sebelumnya.

- **Hasil Pengujian:**

Dengan melakukan bypass pada proses di **section 2**, sistem berhasil menampilkan nilai **Akurasi** di layar tanpa kendala.

## 5 Kesimpulan

Hasil pengujian filter menunjukkan bahwa sebagian besar fungsi telah berjalan sesuai rencana. Pada section pertama, instruksi beserta timer berhasil tampil dan berfungsi dengan baik, memastikan transisi ke section berikutnya berjalan mulus tanpa kendala. Hal ini menunjukkan bahwa bagian awal filter sudah bekerja secara optimal dalam memberikan panduan kepada pengguna.

Namun, pada section kedua, meskipun pembagian gambar menjadi dua sisi (statis dan dinamis) berjalan sesuai harapan, deteksi anggukan sebagai trigger belum berfungsi dengan sempurna, sehingga memerlukan pengembangan lebih lanjut. Meskipun demikian, dengan melewati section 2, dapat dibuktikan pada section ini berhasil memastikan bahwa hasil akhir berupa nilai akurasi dapat ditampilkan di section ketiga, yang menunjukkan potensi implementasi filter secara keseluruhan sudah mendekati siap untuk digunakan setelah perbaikan minor dilakukan.

## 6 Kode Program

```

1 import mediapipe as mp
2 import numpy as np
3 import time
4
5 mp_face_mesh = mp.solutions.face_mesh
6 face_mesh = mp.solutions.face_mesh.FaceMesh(static_image_mode=False, max_num_faces=1,
7         refine_landmarks=True)
8
9 def calculate_similarity(img_left, img_right):
10     height, width = img_left.shape[:2]
11     left_gray = cv2.cvtColor(img_left, cv2.COLOR_BGR2GRAY)
12     right_gray = cv2.cvtColor(img_right, cv2.COLOR_BGR2GRAY)
13     right_resized = cv2.resize(right_gray, (width, height))
14     diff = cv2.absdiff(left_gray, right_resized)
15     score = 100 - np.mean(diff) * 100 / 255
16     return max(0, min(100, round(score)))
17
18 def draw_instruction_box(frame, text, countdown):
19     h, w, _ = frame.shape
20     box_width, box_height = 610, 150
21     top_left_x = (w - box_width) // 2
22     top_left_y = max(0, int(h * 0.2))
23     bottom_right_x = top_left_x + box_width
24     bottom_right_y = top_left_y + box_height
25     overlay = frame.copy()
26     cv2.rectangle(overlay, (top_left_x, top_left_y), (bottom_right_x, bottom_right_y), (0, 128, 0),
27         -1)
28     cv2.addWeighted(overlay, 0.7, frame, 0.3, 0, frame)
29     cv2.rectangle(frame, (top_left_x, top_left_y), (bottom_right_x, bottom_right_y), (0, 255, 0),
30         2)
31     if len(text) > 50:
32         split_index = text[:50].rfind(" ")
33         text1 = text[:split_index]
34         text2 = text[split_index+1:]
35         cv2.putText(frame, text1, (top_left_x + 20, top_left_y + 50), cv2.FONT_HERSHEY_SIMPLEX,
36             0.6, (255, 255, 255), 2)
37         cv2.putText(frame, text2, (top_left_x + 20, top_left_y + 80), cv2.FONT_HERSHEY_SIMPLEX,
38             0.6, (255, 255, 255), 2)
39     else:
40         cv2.putText(frame, text, (top_left_x + 20, top_left_y + 50), cv2.FONT_HERSHEY_SIMPLEX, 0.6,
41             (255, 255, 255), 2)
42     cv2.putText(frame, str(countdown), (top_left_x + box_width // 2 - 20, top_left_y + 120), cv2.
43         FONT_HERSHEY_SIMPLEX, 1.5, (255, 255, 255), 2)
44
45 image_paths = ["data\\1.png"]
46 current_image_index = 0
47 cap = cv2.VideoCapture(0)
48 session, countdown, countdown_start_time = 1, 5, None
49 stop_time, is_nodding, accuracy_displayed = None, False, False
50 nilaiAkurasi, frame_count, sesi2_start_time = None, 0, None
51 process_frame_interval, process_frame_counter = 5, 0
52
53 while cap.isOpened():
54     ret, frame = cap.read()
55     if not ret:
56         break
57     frame = cv2.flip(frame, 1)
58     height, width, _ = frame.shape
59     rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
60     process_frame_counter += 1

```

```

54
55 if process_frame_counter % process_frame_interval == 0:
56     results = face_mesh.process(rgb_frame)
57 if session == 1:
58     if countdown_start_time is None:
59         countdown_start_time = time.time()
60     elapsed_time = time.time() - countdown_start_time
61     if elapsed_time >= 1:
62         countdown -= 1
63         countdown_start_time = time.time()
64     draw_instruction_box(frame, "Anggukan Kepalamu Untuk Mengetes Seberapa Cepat Kamu",
countdown)
65     if countdown <= 0:
66         session, sesi2_start_time = 2, time.time()
67
68 elif session == 2 and current_image_index < len(image_paths):
69     image = cv2.imread(image_paths[current_image_index])
70     if image is None:
71         print(f"Gambar {image_paths[current_image_index]} tidak ditemukan.")
72         current_image_index += 1
73         continue
74     scale_factor = 0.4
75     img_height = int(height * scale_factor)
76     img_width = int(image.shape[1] * img_height / image.shape[0])
77     image = cv2.resize(image, (img_width, img_height))
78     mid_x = image.shape[1] // 2
79     img_left, img_right = image[:, :mid_x], image[:, mid_x:]
80
81     if results.multi_face_landmarks:
82         for face_landmarks in results.multi_face_landmarks:
83             forehead_y = max(0, int(face_landmarks.landmark[10].y * height) - int(img_height *
1.2))
84             left_start_x = max(0, int(face_landmarks.landmark[10].x * width) - mid_x)
85             right_start_x = int(face_landmarks.landmark[10].x * width)
86
87             if forehead_y + img_height <= height and left_start_x + mid_x <= width:
88                 frame[forehead_y:forehead_y+img_height, left_start_x:left_start_x+mid_x] =
img_left
89
90             if forehead_y + img_height <= height and right_start_x + mid_x <= width:
91                 rotation_matrix = cv2.getRotationMatrix2D((mid_x // 2, img_height // 2),
frame_count % 360, 1.0)
92                 rotated_img_right = cv2.warpAffine(img_right, rotation_matrix, (mid_x,
img_height))
93                 frame[forehead_y:forehead_y+img_height, right_start_x:right_start_x+mid_x] =
rotated_img_right
94
95                 top_y = int(face_landmarks.landmark[10].y * height)
96                 chin_y = int(face_landmarks.landmark[152].y * height)
97                 if chin_y - top_y > int(0.05 * height):
98                     is_nodding, stop_time = True, time.time()
99                     nilaiAkurasi = int(calculate_similarity(img_left, rotated_img_right))
100             frame_count += 1
101             if is_nodding and stop_time and time.time() - stop_time >= 5:
102                 session = 3
103
104 elif session == 3 and nilaiAkurasi is not None:
105     accuracy_text = f"Nilai keakuratan kamu sebesar {nilaiAkurasi}%"
106     text_size = cv2.getTextSize(accuracy_text, cv2.FONT_HERSHEY_SIMPLEX, 1, 2)[0]
107     text_x, text_y = (width - text_size[0]) // 2, height // 2
108     cv2.putText(frame, accuracy_text, (text_x, text_y), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255,
255), 2)

```

```
109 cv2.imshow("Reaction Time", frame)
110 if cv2.waitKey(1) & 0xFF == 27:
111     break
112
113
114 cap.release()
115 cv2.destroyAllWindows()
116
```

## 7 Memuat Multi-Gambar

Hasil Penggunaan Filter.



(a) Section 1: Instruksi



(b) Section 2: Gambar Berputar



(c) Section 3: Nilai Akurasi

## References

- [1] J. F. Pimentel, L. Murta, V. Braganholo, and J. Freire, "Understanding and improving the quality and reproducibility of jupyter notebooks," *Empirical Software Engineering*, vol. 26, no. 4, p. 65, 2021.
- [2] R. Aurachman, "Visualization of google mobility data for provinces in indonesia using seaborn python programming package," in *Journal of Physics: Conference Series*, vol. 1833, no. 1. IOP Publishing, 2021, p. 012002.
- [3] F. Lundh, *Python standard library*. " O'Reilly Media, Inc.", 2001.
- [4] G. Van Rossum and F. L. Drake, "Python library reference," 1995.
- [5] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee *et al.*, "Mediapipe: A framework for building perception pipelines," *arXiv preprint arXiv:1906.08172*, 2019.