



Nama: Muhammad Fabil|121140189 - Muhammad Bintang Erlangga Handiananta|121140171
Tugas Ke: Final Project

Mata Kuliah: Sistem/Teknologi Multimedia (IF4021)

Tanggal: 24 Desember 2024

1 Pendahuluan

FingerFacts merupakan inovasi dalam dunia pembelajaran interaktif yang mengintegrasikan teknologi pengenalan gerakan tangan dengan konsep permainan kuis edukatif. Proyek ini dikembangkan sebagai solusi untuk menciptakan pengalaman belajar yang lebih menarik dan interaktif melalui pemanfaatan teknologi computer vision.

1.1 Latar Belakang

Perkembangan teknologi computer vision membuka peluang baru dalam dunia pendidikan dan hiburan. FingerFacts hadir sebagai solusi inovatif yang menggabungkan aspek pembelajaran dengan interaksi fisik melalui pengenalan gerakan tangan.

1.2 Tujuan

- Mengembangkan permainan edukatif yang interaktif
- Memanfaatkan teknologi computer vision untuk pembelajaran
- Menciptakan pengalaman belajar yang menyenangkan

2 Landasan Teori

2.1 Teknologi yang Digunakan

2.1.1 MediaPipe Hands

Framework untuk deteksi dan pelacakan tangan yang memungkinkan pengenalan gerakan jari secara real-time.

2.1.2 OpenCV

Library computer vision untuk pemrosesan gambar yang digunakan untuk pengambilan dan pemrosesan input kamera.

2.2 Komponen Sistem

- Sistem Pengenalan Gerakan
- Antarmuka Pengguna
- Basis Data Pertanyaan

- Sistem Penilaian

3 Analisis Detail Kode Program

3.1 Struktur dan Konfigurasi Awal

3.1.1 Import Libraries dan Inisialisasi

```

1 import cv2
2 import mediapipe as mp
3 import random
4 import time
5 from utils import count_fingers, load_question_image, draw_custom_text
6 from questions import questions
7
8 mp_hands = mp.solutions.hands
9 mp_drawing = mp.solutions.drawing_utils
10 hands = mp_hands.Hands(max_num_hands=1,
11                         min_detection_confidence=0.7,
12                         min_tracking_confidence=0.7)

```

Kode 1: Import dan Inisialisasi

Penjelasan:

- **cv2**: Library OpenCV untuk pemrosesan gambar dan video
- **mediapipe**: Framework untuk deteksi tangan
- **random**: Untuk pemilihan pertanyaan secara acak
- **time**: Untuk manajemen waktu dan timer
- Konfigurasi MediaPipe:
 - **max_num_hands=1**: Hanya mendekripsi satu tangan
 - **min_detection_confidence=0.7**: Threshold deteksi
 - **min_tracking_confidence=0.7**: Threshold tracking

3.2 Fungsi-fungsi Utama

3.2.1 Inisialisasi Game State

```

1 def reset_game():
2     return {
3         'remaining_questions': questions.copy(),
4         'score': 0,
5         'wrong_attempts': 0,
6         'waiting_for_next_question': False,
7         'waiting_for_confirmation': False,
8         'current_question_number': 0,
9         'current_question': None,
10        'game_state': 'opening'
11    }

```

Kode 2: Reset Game Function

Penjelasan Setiap Komponen:

- **remaining_questions**: Salinan dari bank soal

- **score**: Nilai pemain, dimulai dari 0
- **wrong_attempts**: Jumlah kesalahan
- **waiting_for_next_question**: Status menunggu soal berikutnya
- **waiting_for_confirmation**: Status menunggu konfirmasi
- **current_question_number**: Nomor soal saat ini
- **current_question**: Soal yang sedang aktif
- **game_state**: Status permainan ('opening', 'playing', 'game_over')

3.2.2 Pengambilan Pertanyaan

```

1 def get_next_question(game_data):
2     game_data['current_question_number'] += 1
3
4     if game_data['remaining_questions']:
5         question = random.choice(game_data['remaining_questions'])
6         game_data['remaining_questions'].remove(question)
7         return question
8     return None

```

Kode 3: Get Next Question Function

Proses:

1. Increment nomor pertanyaan saat ini
2. Cek ketersediaan pertanyaan
3. Pilih pertanyaan secara acak
4. Hapus pertanyaan yang telah dipilih
5. Kembalikan pertanyaan atau None jika habis

3.3 Manajemen Gambar dan Visual

3.3.1 Overlay Image Function

```

1 def overlay_image(frame, overlay, alpha_channel=True,
2                   x_pos=0, y_pos=0):
3     if overlay is None:
4         return frame
5
6     overlay_h, overlay_w = overlay.shape[:2]
7     overlay_x = x_pos
8     overlay_y = y_pos
9
10    if all(v > 0 for v in [overlay_x, overlay_y,
11                           overlay_w, overlay_h]):
12        for c in range(0, 3):
13            frame_slice = frame[overlay_y:overlay_y + overlay_h,
14                               overlay_x:overlay_x + overlay_w, c]
15            if alpha_channel:
16                alpha = overlay[..., 3] / 255.0
17            else:
18                alpha = 1.0

```

```

19         frame[overlay_y:overlay_y + overlay_h,
20             overlay_x:overlay_x + overlay_w, c] = \
21             frame_slice * (1 - alpha) + overlay[..., c] * alpha
22
23     return frame

```

Kode 4: Overlay Image Function

Penjelasan Detail:**• Parameter:**

- **frame**: Frame video utama
- **overlay**: Gambar yang akan ditambahkan
- **alpha_channel**: Flag untuk transparansi
- **x_pos, y_pos**: Posisi penempatan

• Proses:

- Ekstrak dimensi gambar overlay
- Hitung posisi penempatan
- Terapkan alpha blending untuk setiap channel warna
- Gabungkan gambar dengan frame

3.4 Loop Utama Game

3.4.1 Inisialisasi Loop

```

1 while cap.isOpened():
2     ret, frame = cap.read()
3     if not ret:
4         print("Tidak dapat membaca frame dari kamera.")
5         break
6
7     rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
8     results = hands.process(rgb_frame)

```

Kode 5: Main Game Loop Initialization

Proses:

- Baca frame dari kamera
- Konversi warna ke RGB untuk MediaPipe
- Proses frame untuk deteksi tangan

3.4.2 State Opening

```

1 if game_data['game_state'] == 'opening':
2     x_pos = (frame.shape[1] - game_start_image.shape[1]) // 2
3     y_pos = (frame.shape[0] - game_start_image.shape[0]) // 2
4     frame = overlay_image(frame, game_start_image,
5                           alpha_channel=True,
6                           x_pos=x_pos, y_pos=y_pos)

```

Kode 6: Opening State Handler

Fungsi:

- Menampilkan layar pembuka
- Menempatkan gambar start di tengah layar
- Menunggu input spasi untuk memulai

3.4.3 State Playing

```

1 elif game_data['game_state'] == 'playing':
2     if timer_active:
3         elapsed_time = time.time() - start_time
4         remaining_time = max(0, 7 - elapsed_time)
5
6         if remaining_time <= 0:
7             game_data['wrong_attempts'] += 1
8             timer_active = False
9
10        if game_data['wrong_attempts'] >= 3:
11            game_data['game_state'] = 'game_over'
12        else:
13            game_data['waiting_for_confirmation'] = True

```

Kode 7: Playing State Handler

Komponen:

- Timer:
 - Hitung waktu tersisa (7 detik)
 - Increment kesalahan jika waktu habis
 - Game over setelah 3 kesalahan
- Deteksi Jawaban:
 - Proses landmark tangan
 - Hitung jari yang terangkat
 - Validasi jawaban
- Visual Feedback:
 - Tampilkan gambar pertanyaan
 - Overlay skor dan timer
 - Feedback jawaban benar/salah

3.4.4 State Game Over

```

1 elif game_data['game_state'] == 'game_over':
2     if len(game_data['remaining_questions']) == 0 and \
3         game_data['wrong_attempts'] < 3:
4         well_done_text = "Well Done!"
5         score_text = f"Skor Anda: {game_data['score']}"
6         restart_text = "Tekan 'Backspace' untuk main lagi"
7
8         # Display completion messages
9     else:
10        # Display game over image
11        frame = overlay_image(frame, game_over_image_resized,

```

```

12         alpha_channel=True,
13         x_pos=x_pos, y_pos=y_pos)

```

Kode 8: Game Over State Handler

Kondisi:

- Menang:
 - Semua pertanyaan terjawab
 - Kesalahan kurang dari 3
 - Tampilkan skor akhir
- Kalah:
 - 3 kesalahan tercapai
 - Tampilkan gambar game over

3.5 Penanganan Input

```

1 key = cv2.waitKey(1) & 0xFF
2 if key == ord('q'):
3     break
4 elif key == ord(' ') and game_data['game_state'] == 'opening':
5     game_data['game_state'] = 'playing'
6     start_time = time.time()
7     timer_active = True
8 elif key == ord('n') and game_data['waiting_for_next_question']:
9     next_question = get_next_question(game_data)
10    if next_question:
11        game_data['current_question'] = next_question
12        game_data['waiting_for_next_question'] = False
13        start_time = time.time()
14        timer_active = True
15    else:
16        game_data['game_state'] = 'game_over'

```

Kode 9: Input Handler

Fungsi Tombol:

- **q**: Keluar dari permainan
- **space**: Mulai permainan dari opening
- **n**: Lanjut ke pertanyaan berikutnya
- **l**: Konfirmasi lanjut setelah kesalahan
- **backspace**: Restart game saat game over

3.6 questions.py

3.6.1 Struktur Data Pertanyaan

```

1 questions = [
2     {"question": "Apa ibu kota Indonesia?",
3      "answer": 2,
4      "image": "assets/q1_main.png"},
5     {"question": "Siapa penemu bola lampu?",
6      "answer": 1,
7      "image": "assets/q2_main.png"},
8     # ... more questions ...
9 ]

```

Kode 10: Struktur Data Questions

Komponen Data:

- **question**: String pertanyaan yang akan ditampilkan
- **answer**: Nilai integer yang merepresentasikan jawaban (jumlah jari)
- **image**: Path ke file gambar yang terkait dengan pertanyaan

Karakteristik:

- Format data menggunakan list of dictionaries
- Setiap pertanyaan memiliki gambar ilustrasi
- Jawaban berupa angka 1-5 sesuai dengan jumlah jari
- Total 10 pertanyaan yang mencakup berbagai bidang pengetahuan

3.7 utils.py

3.7.1 Fungsi Penghitung Jari

```

1 def count_fingers(hand_landmarks):
2     finger_tips = [8, 12, 16, 20]
3     finger_pips = [6, 10, 14, 18]
4
5     count = 0
6     for tip, pip in zip(finger_tips, finger_pips):
7         if hand_landmarks.landmark[tip].y < hand_landmarks.landmark[pip].y:
8             count += 1
9
10    thumb_tip = hand_landmarks.landmark[4]
11    thumb_mcp = hand_landmarks.landmark[2]
12    if abs(thumb_tip.x - thumb_mcp.x) > 0.1:
13        count += 1
14
15    return count

```

Kode 11: Fungsi Count Fingers

Mekanisme Deteksi:

- Menggunakan indeks landmark MediaPipe untuk ujung jari dan sendi
- Deteksi jari terangkat berdasarkan posisi vertikal (y)
- Penanganan khusus untuk ibu jari menggunakan posisi horizontal (x)
- Threshold 0.1 untuk deteksi ibu jari terangkat

3.7.2 Fungsi Load Question Image

```

1 def load_question_image(image_path, frame, width=300,
2                         height=261, y_offset=50):
3     question_image = Image.open(image_path).convert("RGBA")
4     question_image = question_image.resize((width, height))
5     question_image_np = np.array(question_image)
6
7     r, g, b, a = cv2.split(question_image_np)
8     bgr_image = cv2.merge([b, g, r])
9     alpha_channel = a
10
11    x_offset = (frame.shape[1] - width) // 2
12    y_offset = max(0, y_offset)

```

Kode 12: Fungsi Load Question Image

Proses Pengolahan Gambar:

- Konversi gambar ke format RGBA untuk transparansi
- Resize gambar ke ukuran standar
- Penanganan kanal warna BGR dan alpha
- Penempatan gambar di tengah frame
- Penanganan boundary checking untuk mencegah error

3.7.3 Fungsi Draw Custom Text

```

1 def draw_custom_text(frame, text, position, font_path,
2                      font_size, color):
3     frame_pil = Image.fromarray(cv2.cvtColor(frame,
4                                         cv2.COLOR_BGR2RGB))
5     draw = ImageDraw.Draw(frame_pil)
6     font = ImageFont.truetype(font_path, font_size)
7
8     draw.text(position, text, font=font, fill=color)
9
10    return cv2.cvtColor(np.array(frame_pil),
11                        cv2.COLOR_RGB2BGR)

```

Kode 13: Fungsi Draw Custom Text

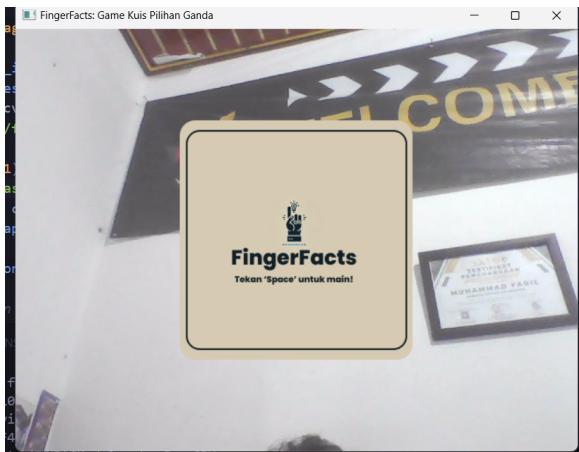
Fitur Rendering Teks:

- Penggunaan PIL untuk rendering teks custom
- Dukungan font TTF untuk tampilan yang lebih baik
- Konversi warna antara OpenCV dan PIL
- Posisi dan ukuran teks yang dapat dikustomisasi
- Penanganan warna RGB/BGR untuk kompatibilitas

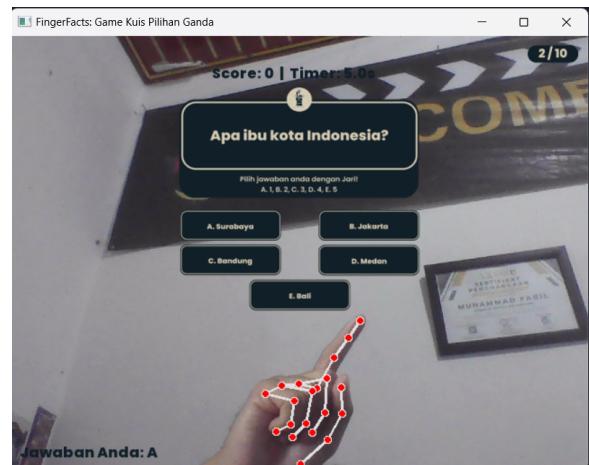
4 Tampilan Program

4.1 Antarmuka Aplikasi

Berikut adalah tampilan dari aplikasi FingerFacts yang menunjukkan berbagai state dan interaksi dalam permainan:

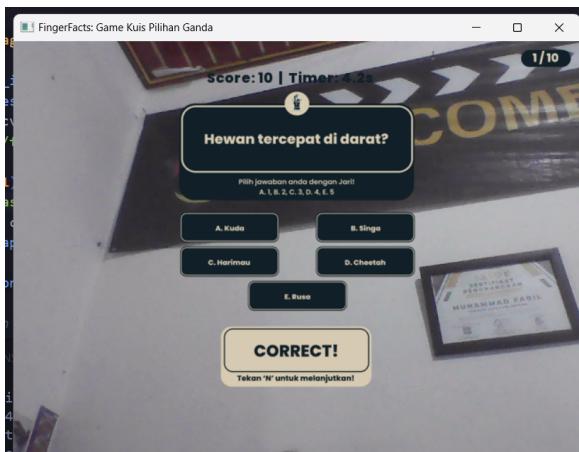


(a) Tampilan Awal Program

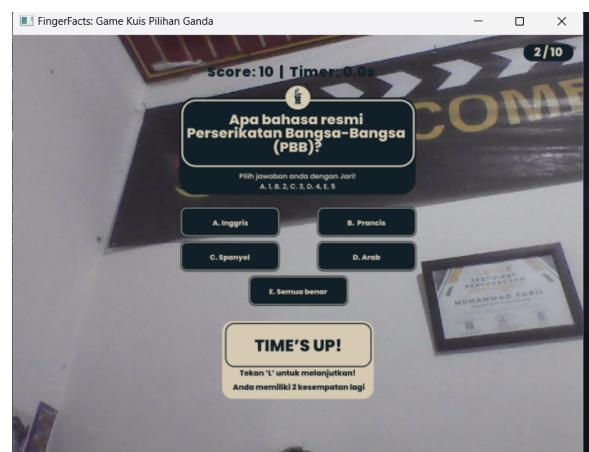


(b) Tampilan Saat Deteksi Jari

Gambar 1: Tampilan Awal Program

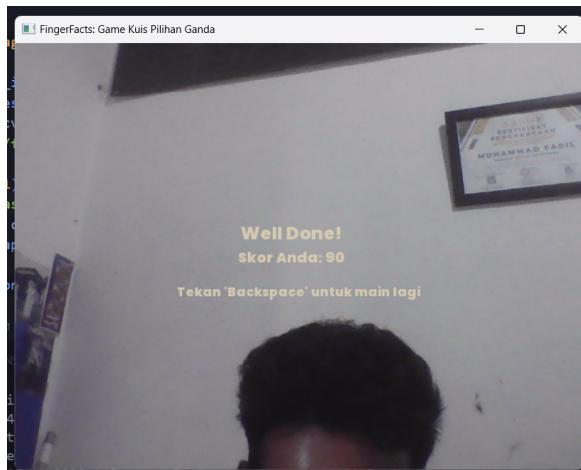


(a) Tampilan Ketika Menjawab Benar

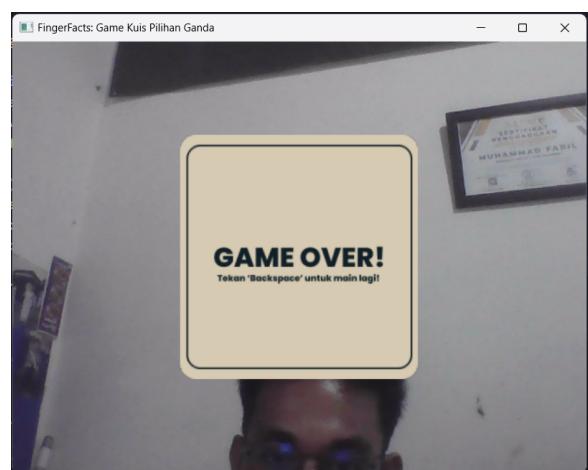


(b) Tampilan Ketika Menjawab Salah

Gambar 2: Tampilan Gameplay



(a) Tampilan Skor Akhir



(b) Tampilan Saat Game Over

Gambar 3: Tampilan Akhir

Penjelasan Tampilan:

- Gambar 1a menunjukkan tampilan awal program ketika dijalankan akan menampilkan logo serta instruksi untuk menekan 'space' untuk mulai bermain.
- Gambar 1b memperlihatkan tahap deteksi jari ketika mulai bermain.
- Gambar 2a menampilkan tampilan ketika menjawab salah.
- Gambar 2a menampilkan tampilan ketika menjawab benar.
- Gambar 3a menampilkan tampilan jumlah score ketika selesai bermain.
- Gambar 3b menampilkan tampilan gameover.

5 Kesimpulan

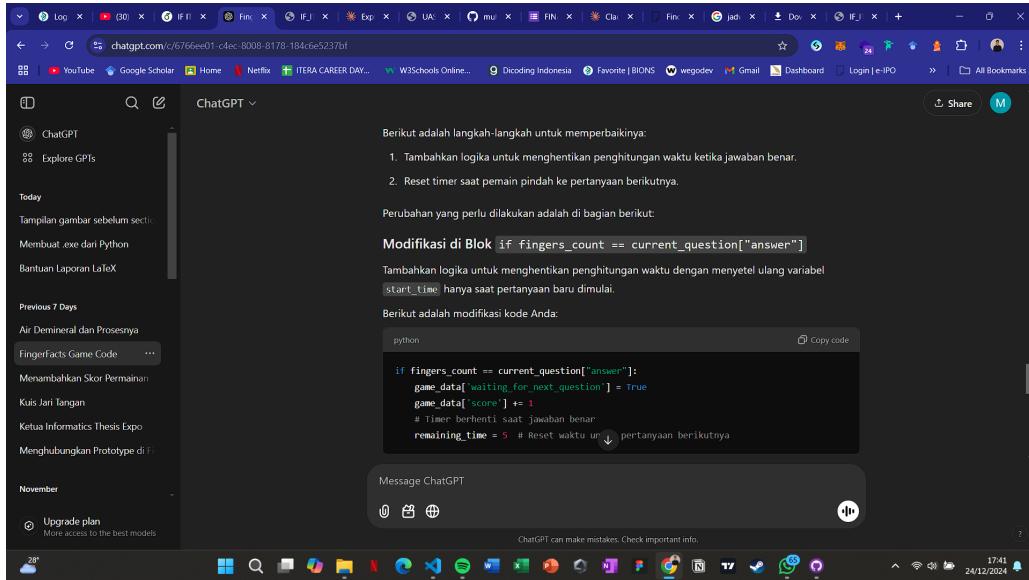
FingerFacts berhasil mengimplementasikan konsep pembelajaran interaktif melalui teknologi pengenalan gerakan tangan. Aplikasi ini mendemonstrasikan potensi integrasi computer vision dalam pendidikan, menciptakan pengalaman belajar yang lebih menarik dan efektif.

5.1 Saran Pengembangan

- Penambahan variasi jenis pertanyaan
- Penambahan sound effect

Lampiran Bukti Bantuan dengan LLM

Dalam penggerjaan proyek ini kami menggunakan bantuan LLM, berikut kami lampirkan beberapa bukti bantuan dengan LLM.



Berikut adalah langkah-langkah untuk memperbaikinya:

1. Tambahkan logika untuk menghentikan penghitungan waktu ketika jawaban benar.
2. Reset timer saat pemain pindah ke pertanyaan berikutnya.

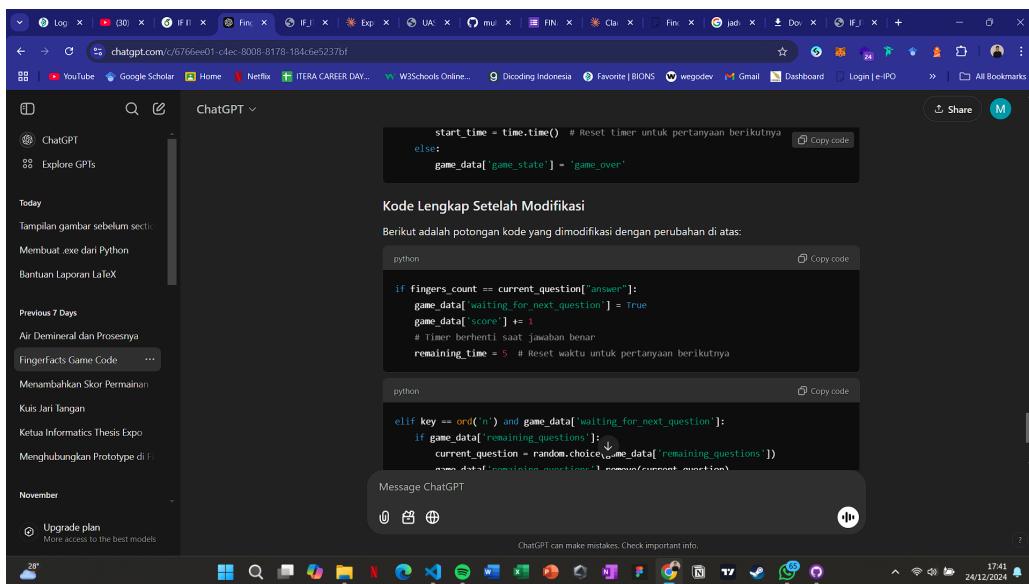
Perubahan yang perlu dilakukan adalah di bagian berikut:

```
Modifikasi di Blok if fingers_count == current_question["answer"]
```

Tambahkan logika untuk menghentikan penghitungan waktu dengan menyetel ulang variabel `start_time` hanya saat pertanyaan baru dimulai.

```
if fingers count == current_question["answer"]:
    game_data['waiting_for_next_question'] = True
    game_data['score'] += 1
    # Timer berhenti saat jawaban benar
    remaining_time = 5 # Reset waktu untuk pertanyaan berikutnya
```

Message ChatGPT



Berikut adalah potongan kode yang dimodifikasi dengan perubahan di atas:

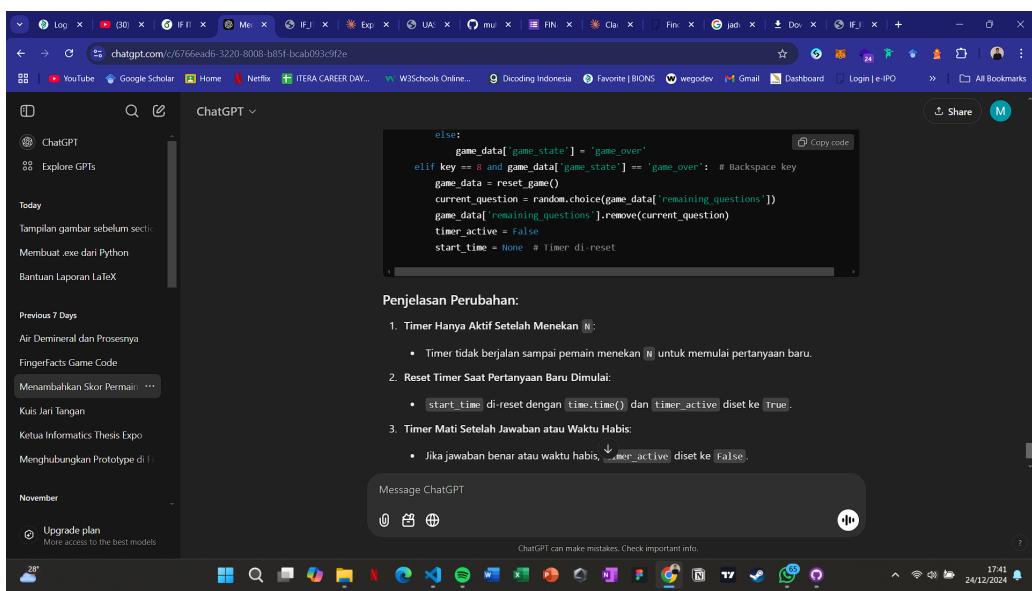
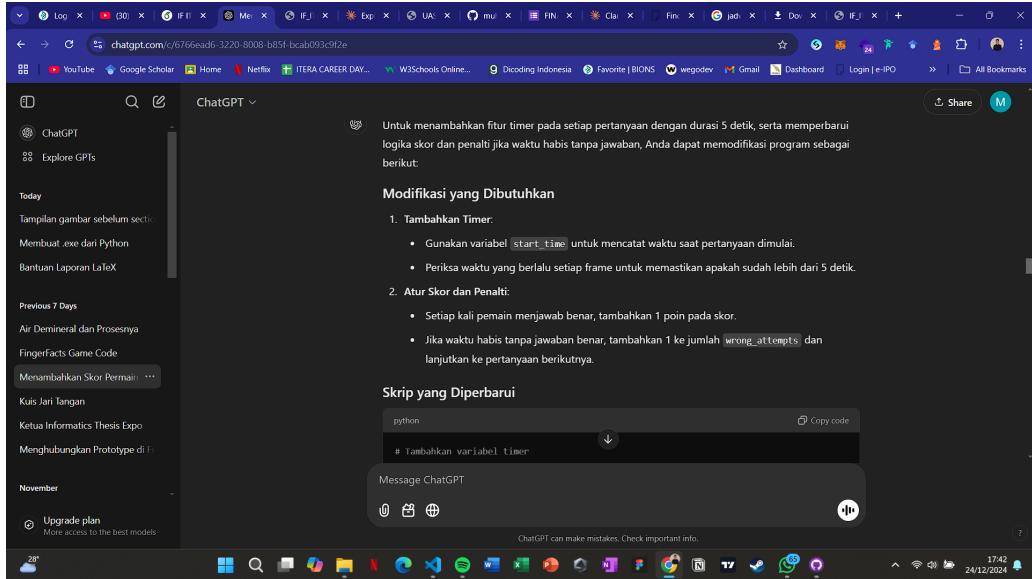
```
start_time = time.time() # Reset timer untuk pertanyaan berikutnya
else:
    game_data['game_state'] = 'game_over'
```

Kode Lengkap Setelah Modifikasi

```
if fingers_count == current_question["answer"]:
    game_data['waiting_for_next_question'] = True
    game_data['score'] += 1
    # Timer berhenti saat jawaban benar
    remaining_time = 5 # Reset waktu untuk pertanyaan berikutnya
```

```
elif key == ord('n') and game_data['waiting_for_next_question']:
    if game_data['remaining_questions'] > 0:
        current_question = random.choice(game_data['remaining_questions'])
        game_data['remaining_questions'].remove(current_question)
```

Message ChatGPT



Referensi

Referensi yang digunakan dalam pengembangan proyek ini: [Video Referensi](#)