



Nama: **Kharisma Gumilang (121450142)**, **Ericson Chandra Sihombing (121450026)**, **Shula Talitha Ardhya Putri (121450087)**

Tugas Ke: **UAS**

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

Tanggal: Tanggalnya

1 Abstrak

Laporan ini menguraikan implementasi dari sebuah game interaktif berjudul "**Gesture Diagonal Obstacle Game**" yang dikontrol sepenuhnya oleh gerakan tangan pengguna. Proyek ini dibangun menggunakan Python, memanfaatkan pustaka **MediaPipe** untuk deteksi dan pelacakan tangan secara *real-time* dari input *webcam*, **OpenCV** untuk pemrosesan video dan antarmuka, serta **Pygame** untuk pengelolaan audio. Tujuan utama game adalah agar pemain mencocokkan gestur tangannya dengan gestur yang diminta oleh rintangan yang bergerak secara diagonal. Program ini dilengkapi dengan fitur seperti deteksi gestur spesifik (Open Hand, Peace, Metal, Fist, Pointing), sistem skor, penghitung kegagalan, mode koreksi (*retry*), serta antarmuka pengguna yang informatif termasuk layar menu, instruksi, dan *game over*. Implementasi ini menunjukkan aplikasi praktis dari visi komputer dan *machine learning* untuk menciptakan pengalaman bermain game yang imersif dan tanpa kontak fisik.

2 1. Pendahuluan

Perkembangan dalam bidang *Human-Computer Interaction* (HCI), khususnya dalam pengenalan gestur, telah membuka peluang baru untuk antarmuka yang lebih intuitif dan alami. Dalam konteks game, kontrol berbasis gestur memungkinkan pemain untuk berinteraksi dengan lingkungan virtual menggunakan gerakan tubuh mereka, yang dapat meningkatkan imersif dan pengalaman bermain.

Proyek "Gesture Diagonal Obstacle Game" ini merupakan demonstrasi dari konsep tersebut. Alih-alih menggunakan keyboard atau kontroler tradisional, pemain mengendalikan game hanya dengan membentuk gestur tangan yang berbeda di depan kamera. Game ini menantang pemain untuk merespons rintangan yang bergerak dengan cepat dengan gestur tangan yang tepat. Implementasi proyek ini akan dijelaskan secara detail, termasuk arsitektur kode, komponen utama, dan mekanisme game yang dibangun.

3 2. Deskripsi Proyek

3.1 Latar Belakang

Permainan digital terus mencari inovasi dalam metode interaksi untuk meningkatkan keterlibatan pemain. Kontrol fisik tradisional terkadang membatasi pengalaman yang lebih dinamis. Proyek ini hadir sebagai solusi yang memanfaatkan teknologi visi komputer untuk memungkinkan interaksi non-kontak. Dengan fokus pada deteksi gestur tangan menggunakan MediaPipe, game ini bertujuan untuk memberikan pengalaman bermain yang unik dan imersif, di mana gerakan alami pemain menjadi inti dari *gameplay*.

3.2 Tujuan

Tujuan utama dari proyek "*Gesture Diagonal Obstacle Game*" adalah:

1. Mengembangkan sistem deteksi gestur tangan *real-time* menggunakan *MediaPipe Hand Landmarker*.
2. Membuat *game* berbasis rintangan diagonal di mana pemain harus mencocokkan gestur tangan yang diminta.
3. Mengimplementasikan logika game yang mencakup sistem skor, penghitung kegagalan, dan mode koreksi untuk gestur yang salah.
4. Merancang antarmuka pengguna yang intuitif, termasuk layar menu, instruksi, dan *game over*.
5. Menunjukkan kemampuan integrasi berbagai pustaka *Python* (*OpenCV*, *MediaPipe*, *NumPy*, *Pygame*) untuk aplikasi interaktif.

3.3 Manfaat

1. **Pengalaman Bermain Inovatif:** Menyediakan cara bermain game yang baru dan menarik tanpa perangkat kontrol fisik, meningkatkan imersif.
2. **Edukasi dan Demonstrasi Teknis:** Berfungsi sebagai contoh praktis bagaimana visi komputer dan pengenalan gestur dapat diterapkan dalam aplikasi *real-time*.
3. **Aksesibilitas Potensial:** Dapat menjadi dasar untuk pengembangan game yang lebih mudah diakses oleh individu dengan kebutuhan khusus dalam interaksi.

4 Implementasi Teknis

Proyek ini diimplementasikan dalam Python, dengan arsitektur yang terbagi menjadi beberapa bagian fungsional.

4.1 Pustaka yang Digunakan

Program ini mengintegrasikan beberapa pustaka Python kunci untuk fungsionalitasnya:

- **cv2 (OpenCV):** Digunakan untuk manipulasi gambar dan video, termasuk pembacaan *frame* dari *webcam*, penggambaran objek (*overlay*, teks), dan pengelolaan jendela tampilan game.
- **mediapipe as mp:** Pustaka utama untuk deteksi dan pelacakan landmark tangan secara *real-time*. Modul **mp.solutions.hands** menyediakan model *machine learning* yang ringan untuk mengidentifikasi 21 titik kunci pada setiap tangan.
- **numpy as np:** Digunakan untuk operasi numerik efisien, terutama dalam pemrosesan data koordinat landmark dan perhitungan lainnya.
- **pygame:** Mengelola fungsionalitas audio game, termasuk memutar musik latar dan efek suara untuk skor, peringatan, dan *game over*.
- **os:** Digunakan untuk operasi sistem file, seperti mengelola jalur *resource* (gambar, suara) agar portabel di berbagai sistem operasi.
- **time:** Untuk fungsi terkait waktu, seperti mengukur *frame rate* atau mengelola durasi efek.
- **random:** Digunakan untuk menghasilkan angka acak, seperti memilih jenis rintangan secara acak atau lelucon pada layar *game over*.

4.2 Komponen Utama Kode Program

4.2.1 Inisialisasi Audio (**initialize_pygame_audio**)

Fungsi ini bertanggung jawab untuk memulai sistem audio Pygame dan memuat semua aset suara yang diperlukan (musik latar, efek suara skor, peringatan, *game over*). Jika ada kesalahan dalam memuat file suara, pesan akan dicetak ke konsol.

Struktur Audio

- `pygame.mixer.init()`: Menginisialisasi modul pencampur suara Pygame.
- `pygame.mixer.music.load()`: Memuat file musik latar (`stecu.wav`).
- `pygame.mixer.Sound()`: Memuat efek suara terpisah (`score.mp3`, `beep-warning-6387.mp3`, `game-over-arcade-6435.mp3`).
- `pygame.mixer.music.set_volume(0.3)`: Mengatur volume musik latar.

4.2.2 Konstanta dan Variabel Global

Beberapa konstanta didefinisikan untuk mengonfigurasi aspek-aspek game:

- `OBSTACLE_SIZE`, `INITIAL_OBSTACLE_SPEED_X/Y`, `SPEED_INCREASE_FACTOR_X/Y`
- `SCORE_INCREASE_INTERVAL`, `MAX_FAILS`
- `DETECTION_ZONE_X/Y_START/END_RATIO`
- `RETRY_RETREAT/ADVANCE_SPEED_X/Y`
- `OBSTACLE_TYPES`, `JOKES`
- `STATE_MENU`, `STATE_INSTRUCTIONS`, `STATE_PLAYING`, `STATE_GAMEOVER`

4.2.3 Fungsi Bantu Umum

- `play_sound(sound)`, `draw_text_with_outline(...)`, `draw_button(...)`, `is_click_on_button(...)`
- `draw_pose_obstacle(...)`, `create_obstacle(...)`

4.2.4 Logika Pendeteksi Gestur Tangan (**detect_gesture**)

Fungsi utama kontrol gestur:

- Deteksi berdasarkan 21 landmark dari MediaPipe.
- Fungsi bantu: `is_finger_extended`, `is_finger_bent`, `is_thumb_open_general`
- Gestur dikenali: `Pointing`, `Peace`, `Metal`, `Open Hand`, `Fist`

4.2.5 Event Handler Mouse (**mouse_callback**)

Menangani klik kiri mouse untuk deteksi tombol UI.

4.2.6 Antarmuka Pengguna (UI)

- Fungsi UI: `draw_detection_zone`, `draw_hud_panel`, `render_menu_screen`, `render_instructions_screen`, `render_game_info`, `render_gameover_screen`

4.2.7 Fungsi Gameplay Utama (**run_gameplay_loop**)

Mengelola logika:

- Pergerakan rintangan
- Mode koreksi (*retry*)
- Deteksi gestur dan zona
- Skor, kegagalan, dan pembuatan rintangan baru

Fungsi Reset Game (**reset_game**)

Mengatur ulang semua status game ke kondisi awal.

Alur Eksekusi Utama (**main function**)

1. Inisialisasi:

- (a) Membuka *webcam* dan mengatur resolusi.
- (b) Membuat jendela **OpenCV** fullscreen, mendaftarkan *callback*.
- (c) Memuat audio dan menginisialisasi detektor tangan.
- (d) Mengatur ulang status game dan memulai musik latar.

2. Game Loop:

- (a) Membaca dan membalik *frame*.
- (b) Memproses *frame* dengan **hands_detector**.
- (c) Jika tangan terdeteksi dan status game aktif:
 - Gambar landmark dan deteksi gestur
 - Efek visual di pergelangan tangan
- (d) **Manajemen Status Game:**
 - **STATE_MENU**: Menu utama
 - **STATE_INSTRUCTIONS**: Petunjuk permainan
 - **STATE_PLAYING**: Gameplay aktif
 - **STATE_GAMEOVER**: Tampilkan skor dan lelucon
- (e) Tampilkan *frame* akhir ke jendela.
- (f) Menunggu input keyboard untuk keluar.

3. Pembersihan:

- Menghentikan musik
- Melepaskan kamera
- Menutup jendela **OpenCV**

5 Rencana Kerja Proyek

Proyek ini akan dilaksanakan dalam delapan minggu, dengan pembagian tugas yang jelas untuk memastikan progres yang efisien.

5.1 Tahapan Pengembangan

1. Minggu 1-2: Studi Mendalam & Perancangan

- Memahami secara detail dokumentasi MediaPipe Hand Landmarker.
- Mendesain arsitektur game, termasuk definisi gestur dan pemetaan aksinya.
- Mengumpulkan atau membuat aset grafis (emoji rintangan) dan audio (musik, efek suara).

2. Minggu 3-4: Implementasi Deteksi Gestur & Dasar Game

- Mengimplementasikan inisialisasi kamera dan integrasi MediaPipe.
- Mengembangkan fungsi `detect_gesture()` secara akurat untuk gestur yang diperlukan.
- Membuat logika dasar pergerakan rintangan dan sistem deteksi tabrakan.

3. Minggu 5-6: Logika Gameplay & UI

- Mengimplementasikan sistem skor, penghitung kegagalan, dan logika mode koreksi (*retry*).
- Mengembangkan semua fungsi rendering UI (*menu, instructions, game over, HUD*).
- Mengintegrasikan fungsionalitas audio Pygame ke dalam game.

4. Minggu 7-8: Pengujian, Debugging & Finalisasi

- Melakukan pengujian ekstensif terhadap akurasi deteksi gestur dan responsivitas game.
- Memperbaiki *bug* dan melakukan optimasi kinerja.
- Menyempurnakan tampilan UI dan efek visual.
- Menyusun laporan proyek akhir dan mempersiapkan demo.

5.2 Potensi Tantangan

Beberapa tantangan yang mungkin dihadapi selama proyek meliputi:

- **Akurasi Deteksi Gestur:** Variasi dalam ukuran tangan, pencahayaan, dan gaya gerakan pengguna dapat mempengaruhi akurasi MediaPipe. Diperlukan penyesuaian ambang batas deteksi.
- **Kinerja Real-time:** Memastikan bahwa pemrosesan video dan deteksi gestur dapat berjalan dengan lancar tanpa *lag* yang signifikan, terutama pada perangkat keras yang lebih rendah.
- **User Experience:** Merancang gestur yang intuitif dan mudah dilakukan secara konsisten, serta memberikan umpan balik yang jelas kepada pemain.
- **Asset Management:** Memastikan semua file gambar dan suara dimuat dengan benar dan kompatibel dengan format yang diharapkan.

6 Kesimpulan

Proyek "Gesture Diagonal Obstacle Game" ini berhasil mengimplementasikan game interaktif yang dikontrol menggunakan gerakan tangan, memanfaatkan kapabilitas deteksi tangan *real-time* dari MediaPipe. Dengan integrasi yang efektif antara visi komputer, logika game, dan antarmuka pengguna, proyek ini tidak hanya mencapai tujuannya untuk menciptakan pengalaman bermain yang inovatif, tetapi juga berfungsi sebagai bukti konsep yang kuat tentang potensi interaksi non-kontak dalam aplikasi hiburan. Tantangan yang ada akan diatasi melalui pengujian dan optimasi berkelanjutan untuk menghasilkan produk akhir yang robust dan menyenangkan.

7 Referensi dan Daftar Pustaka

1. <https://gemini.google.com/app/ba18a7fedd36d23f?hl=id>
2. <https://g.co/gemini/share/81a8360afe08>
3. https://ai.google.dev/edge/mediapipe/solutions/vision/hand_landmarker
4. <https://vt.tiktok.com/ZSrvavKEX/>
5. <https://chatgpt.com/share/683b1e60-da60-8000-b128-e76f4350b52f>

References