



Nama: **Bayu Prameswara Haris, Muhammad Riveldo Hermawan Putra, Joshia Fernandes Sectio Purba, (122140219, 122140037, 122140170)** Tugas Ke: **Tugas Besar**
Mata Kuliah: **Sistem / Teknologi Multimedia(IF25-40305)** Tanggal: 12 Desember 2025

1 Pendahuluan

1.1 Latar Belakang

Perkembangan teknologi multimedia telah menjadikan game mobile sebagai bagian tak terpisahkan dari hiburan kita sehari-hari. Di Indonesia sendiri, salah satu game yang sangat populer dan menawarkan keragaman konten, mulai dari karakter hingga suara unik, adalah Mobile Legends. Namun, di tengah semua kemajuan ini, satu hal yang seringkali terasa stagnan adalah cara kami berinteraksi dengan game itu sendiri—semuanya masih terbatas pada menekan tombol virtual atau menyentuh layar. Keterbatasan pada kontrol tradisional ini, terutama pada genre game edukasi atau kuis, menciptakan sebuah kesenjangan sehingga butuh cara yang lebih alami dan imersif agar pemain tidak hanya melihat, tapi juga merasakan keterlibatan penuh dengan permainannya.

Untuk menjembatani kesenjangan interaksi tersebut, proyek ini mengambil pendekatan melalui gamifikasi dan pemrosesan visual. Penerapan gamification telah terbukti menjadi solusi yang inovatif dan efektif untuk mendorong motivasi serta ketertarikan pemain/siswa dalam belajar [1]. Hal ini menjadi landasan mengapa perlunya menggabungkan unsur edukasi dengan interaksi fisik yang unik. Solusi teknisnya adalah dengan memanfaatkan teknologi Computer Vision. Daripada menekan tombol, sistem kami akan mendeteksi gerakan kepala pemain. Pendekatan ini dipilih karena penelitian menunjukkan bahwa penggunaan pustaka Computer Vision seperti MediaPipe dan OpenCV sangat efektif untuk menciptakan mekanisme input non-tradisional, memungkinkan pemain berinteraksi tanpa menyentuh perangkat keras [2]. Inilah yang memungkinkan terciptanya interaksi real-time yang dicari.

Atas dasar kebutuhan akan interaksi baru dan dukungan teknologi yang sudah ada, proyek Tugas Besar ini menghasilkan sebuah game edukasi interaktif multimedia berjudul "Guess The Voices of The ML Hero". Konsepnya sederhana namun menantang: pemain diminta menebak hero hanya dari suaranya, dan jawaban diberikan melalui gerakan kepala—miring ke kiri atau ke kanan. Secara teknis, sistem ini mengintegrasikan deteksi landmark wajah dari MediaPipe FaceMesh untuk membaca orientasi kepala, sementara pemutaran suara hero diurus secara asynchronous (non-blocking). Dengan adanya integrasi ini, "Guess The Hero" tidak hanya berfungsi sebagai alat uji pengetahuan tentang hero Mobile Legends, tetapi juga menjadi bukti nyata implementasi sistem Computer Vision yang fungsional dan real-time sebagai kontrol input dalam sebuah aplikasi hiburan edukatif.

1.2 Rumusan Masalah

Permasalahan yang ingin diselesaikan melalui pengembangan game ini dapat dirumuskan menjadi pertanyaan-pertanyaan teknis dan implementatif berikut:

1. Bagaimana cara membuat game ini yang menggabungkan pengenalan suara (asynchronous) dan deteksi gerakan kepala (Head Tilt) dalam sebuah sistem multimedia interaktif?

2. Bagaimana cara menerapkan pemutaran audio, deteksi wajah (MediaPipe FaceMesh), dan pengolahan orientasi kepala untuk menciptakan interaksi yang responsif dalam game ini?

1.3 Tujuan dan Manfaat

Berikut merupakan tujuan dan manfaat dari proyek game ini :

1. Tujuan:

- Menerapkan pemutaran audio yang terintegrasi dengan gerakan wajah (kepala) pemain, serta menggunakan MediaPipe untuk mendeteksi wajah dan orientasi kemiringan, menciptakan respons game yang dinamis dan menarik.
- Merancang dan mengembangkan game interaktif yang menggabungkan teknologi multimedia seperti pengenalan suara, deteksi wajah, dan pengolahan gerakan kepala untuk menciptakan pengalaman bermain yang lebih imersif.

2. Manfaat:

- Meningkatkan keterlibatan pemain dengan menggabungkan input fisik (gerakan kepala) dan audio.
- Memperkenalkan teknologi multimedia yang meningkatkan interaksi dalam game edukasi.
- Memberikan pengalaman bermain yang interaktif dan mendidik melalui media suara dan visual.

1.4 Ruang Lingkup

Pengembangan game ini terbatas pada platform Dekstop, dengan fokus pada genre game edukasi interaktif. Game ini akan menggabungkan fitur pengenalan suara dan deteksi gerakan kepala menggunakan webcam atau sensor lain yang relevan. Batasan fitur yang akan dibangun mencakup pemilihan hero, pengenalan suara berdasarkan arah gerakan kepala, serta interaksi berbasis gerakan yang disesuaikan dengan suara hero yang dihasilkan. Fitur tambahan yang melibatkan integrasi dengan media sosial atau pengolahan data besar tidak akan dijadikan fokus utama dalam pengembangan game ini.

1.5 Metodologi Pengembangan

Metodologi pengembangan yang digunakan dalam pembuatan game ini adalah metode **Multimedia Development Life Cycle (MDLC)**. Metode ini dipilih karena pendekatannya yang sistematis dalam pengembangan aplikasi multimedia, mencakup tahap perencanaan, desain, implementasi, pengujian, dan pemeliharaan. Proses pengembangan dimulai dengan analisis kebutuhan dan desain game, diikuti dengan implementasi menggunakan bahasa pemrograman Python, yang memungkinkan integrasi suara dan gerakan kepala. Selanjutnya, game diuji untuk memastikan fungsionalitas dan kelancaran pengalaman pengguna.

2 Tinjauan Pustaka

2.1 Studi Literatur

Studi literatur ini mengumpulkan dan menganalisis penelitian-penelitian terdahulu yang relevan sebagai landasan konseptual dan teknis proyek, khususnya pada dua pilar teknologi utama: Computer Vision untuk kontrol gerakan dan pemrosesan multimedia real-time.

2.1.1 Computer Vision

Inovasi dalam interaksi manusia-komputer telah mengarahkan pada penggunaan gerakan tubuh sebagai mekanisme kontrol, menggantikan perangkat input fisik. Dalam konteks aplikasi interaktif seperti game, deteksi orientasi wajah (kemiringan kepala) dianggap sebagai solusi yang imersif dan ergonomis. Penelitian mengenai sistem kendali berbasis orientasi wajah, seperti yang dikembangkan oleh Haikal et al. [3], telah membuktikan efektivitas pustaka MediaPipe Face Mesh.

Sistem deteksi orientasi wajah memanfaatkan beberapa titik landmark penting pada wajah, seperti titik pada hidung, dan titik terluar mata kiri dan kanan, untuk menghitung parameter sudut kemiringan. Secara teknis, perhitungan sudut kemiringan horizontal (roll angle) didasarkan pada dua titik utama pada garis mata (eye-baseline), yaitu landmark ID 33 (mata kiri) dan ID 263 (mata kanan). Sudut kemiringan yang dihasilkan dari garis hubung kedua titik tersebut kemudian dibandingkan dengan nilai ambang (threshold).

Penelitian tersebut menggunakan pendekatan Rule-Based Reasoning (RBR) atau klasifikasi berbasis aturan, yang menetapkan nilai ambang (threshold) sudut untuk menerjemahkan kemiringan menjadi perintah yang spesifik. Temuan ini sangat relevan sebagai landasan teoritis proyek ini, menegaskan bahwa MediaPipe adalah tools yang tepat dan fungsional untuk mengidentifikasi kemiringan kepala secara akurat dan dijadikan sebagai sinyal kontrol dalam game kuis.

2.1.2 Parallel Processing

Aplikasi multimedia yang melibatkan live stream dari webcam dan pemutaran audio memiliki tantangan utama dalam menjaga performa real-time tanpa mengalami delay (jeda) atau freeze. Pemrosesan video merupakan tugas komputasi yang berat dan berpotensi memblokir alur utama program.

Untuk mengatasi masalah ini, teknik Parallel Processing atau Multithreading menjadi solusi yang krusial. Penelitian yang dilakukan oleh Widjanarko et al. [4] telah menganalisis efektivitas teknik parallel processing pada pemrosesan video menggunakan pustaka OpenCV dan bahasa Python. Hasilnya menunjukkan bahwa penerapan multithreading dapat menghasilkan perbedaan kinerja yang signifikan dibandingkan tanpa pemrosesan paralel, terutama ketika sumber data berasal dari webcam. Prinsip ini menjadi landasan arsitektur sistem proyek ini, di mana pemrosesan frame utama dan tugas I/O (seperti pemutaran file audio) harus berjalan secara asynchronous untuk memastikan game tetap responsif dan lancar (Non-Blocking).

2.2 Metode Pengembangan Game

Bagian ini akan membahas prinsip-prinsip non-teknis yang memandu perancangan game agar pengalaman pengguna (user experience) menjadi optimal dan imersif. Teori desain game dan kaidah multimedia yang relevan diterapkan untuk memastikan keseimbangan antara fungsionalitas teknologi dan kenyamanan bermain.

2.2.1 Prinsip Desain Interaksi dan Teori Flow

Untuk menjamin engagement dan retensi pemain yang maksimal, game ini mengadopsi prinsip desain dari Teori Flow. Teori ini menjelaskan kondisi mental di mana seseorang sepenuhnya tenggelam dalam suatu aktivitas, tercapai ketika terdapat keseimbangan ideal antara tingkat tantangan yang disajikan oleh game dan keterampilan yang dimiliki pemain.

Dalam konteks game interaktif ini, penerapan teori flow berfokus pada:

1. Mengatur Tingkat Kesulitan: Dengan menyajikan tantangan melalui pengenalan suara hero yang terkadang mirip atau ambigu.

2. Memastikan Kontrol yang Mudah Dikuasai: Menggunakan input gerakan kepala (kemiringan) yang natural dan intuitif, sehingga pemain dapat fokus pada pemecahan masalah (menebak hero) dan bukan pada kesulitan mengoperasikan kontrol.

Keseimbangan antara tantangan suara dan kemudahan kontrol ini esensial untuk menjaga pemain tetap berada dalam zona flow (keterlibatan mendalam) tanpa merasa frustrasi [5].

2.2.2 Desain Antarmuka Pengguna dan Pengalaman (UI/UX)

Kualitas Antarmuka Pengguna (User Interface) dan Pengalaman Pengguna (User Experience) adalah faktor krusial dalam keberhasilan game interaktif. Perancangan UI/UX yang efektif memastikan game tidak hanya fungsional secara teknis tetapi juga memberikan kenyamanan dan kejelasan informasi bagi pemain serta pengalaman pengguna yang tak terlupakan berpusat pada pemahaman mendalam tentang pengguna, perancangan yang kreatif, dan pentingnya responsivitas.

Prinsip-prinsip UI/UX yang diterapkan dalam game ini meliputi:

1. Desain Imersif: Antarmuka visual (kotak pertanyaan, pilihan jawaban) dirancang secara minimalis dan bersifat dinamis. Elemen-elemen ini diposisikan sedemikian rupa agar selalu berada di dekat fokus visual utama pemain, yaitu area wajah yang dideteksi oleh webcam. Pendekatan ini secara aktif menghindari antarmuka statis dan meminimalkan gangguan visual, sehingga meningkatkan aspek immersion dan mendorong keterlibatan fisik pemain.
2. Umpan Balik Instan dan Jelas: Dalam game berkecepatan tinggi, kejelasan dan kecepatan umpan balik (feedback) adalah prioritas. Oleh karena itu, sistem harus mampu menampilkan indikator visual (ikon correct/wrong dan gambar hero yang benar) segera setelah input gerakan kepala terdeteksi, mempertahankan siklus respons yang cepat dan memuaskan [6].

3 Metodologi Pengembangan

3.1 Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk menentukan spesifikasi fungsional dan non-fungsional yang harus dipenuhi oleh game "Guess The Voices of The ML Hero", didasarkan pada tujuan proyek dan studi literatur.

1. Kebutuhan Fungsional :
 - Sistem harus mampu mendeteksi orientasi kepala (Kiri/Kanan/Tengah) menggunakan webcam dan Computer Vision (MediaPipe).
 - Sistem harus mampu memutar suara hero secara asynchronous (non-blocking).
 - Sistem harus mampu memilih pertanyaan secara acak dan mengevaluasi jawaban dari input gerakan kepala.
 - Sistem harus menampilkan feedback visual (skor dan indikator benar/salah) secara instan.
2. Kebutuhan Non-Fungsional :
 - Performa deteksi harus real-time (latensi rendah).
 - Antarmuka harus sederhana dan mudah digunakan (usable).

3.2 Multimedia Development Life Cycle

Metode pengembangan yang digunakan dalam proyek "Guess The Hero" adalah Multimedia Development Life Cycle (MDLC). Metode ini dipilih karena pendekatannya yang sistematis, fleksibel, dan sangat sesuai untuk pengembangan aplikasi multimedia interaktif yang membutuhkan integrasi aset (audio, visual) dan teknologi real-time (video processing).

Tahapan MDLC diterapkan sebagai kerangka kerja proyek, meliputi enam fase utama berikut:

1. Tahap Konsep (Concept) : Tujuan dari konsep ini ialah mendefinisikan ide, tujuan, dan sasaran pengguna aplikasi. Tahap ini menghasilkan ide dasar game kuis tebak suara hero Mobile Legends menggunakan input gerakan kepala. Output formal dari tahap ini adalah rumusan masalah, tujuan proyek, dan penetapan sasaran pengguna.
2. Tahap Perancangan (Design) : Tujuan tahap ini untuk membuat spesifikasi rinci mengenai tampilan, arsitektur, dan alur interaksi sistem. Tahap ini mencakup perancangan antarmuka pengguna (UI/UX) sesuai prinsip flow, perancangan Game Design Document, dan perancangan arsitektur diagram blok sistem, yaitu
 - Perancangan modul Deteksi Wajah (input webcam ke MediaPipe).
 - Perancangan modul Manajemen Audio (threading) agar tidak memblokir game loop.
 - Perancangan Logika Kuis (pemilihan hero acak dan mekanisme evaluasi jawaban).
3. Tahap Pengumpulan Material (Material Collecting) : Tujuan tahap ini untuk mengumpulkan dan memproses semua aset digital yang diperlukan sebelum implementasi kode. Terdapat 3 kategori pengumpulan material, yaitu file suara hero Mobile Legends (.mp3), file gambar hero (.jpg) dan aset UI (latar belakang, ikon feedback), dan penyusunan basis data hero statis (nama, suara, gambar) yang dimuat ke dalam sistem.
4. Tahap Pembuatan (Assembly) : Tujuan tahap ini untuk melakukan implementasi kode program dan mengintegrasikan seluruh aset sesuai perancangan yang terbagi menjadi beberapa macam :
 - Penyusunan kode utama (game loop) yang mengatur siklus pertanyaan dan skor.
 - Implementasi fungsionalitas Computer Vision (OpenCV dan MediaPipe) untuk deteksi orientasi wajah.
 - Pengembangan modul manajemen audio yang menggunakan teknik threading untuk playback non-blocking.
 - Integrasi user interface (UI Overlay) ke dalam frame video yang diproses.
5. Tahap Pengujian (Testing) : Tujuan tahap ini untuk memastikan semua fungsionalitas sistem berjalan sesuai spesifikasi dan memenuhi tujuan proyek. Melakukan dua jenis pengujian, yaitu pengujian fungsionalitas teknis (akurasi deteksi gerakan kepala dan fungsi threading audio) dan pengujian oleh pengguna untuk mengevaluasi usability, kenyamanan, dan kepuasan bermain (user acceptance test).
6. Tahap Distribusi (Distribution) : Tujuan dari konsep ini untuk menyiapkan aplikasi agar dapat diakses dan digunakan oleh pengguna akhir dan embuat paket aplikasi menjadi executable standalone yang siap dipasang dan dijalankan pada sistem operasi desktop yang memenuhi persyaratan minimum (webcam).

3.3 Perancangan Sistem dan GDD

Perancangan ini meliputi aspek konseptual permainan (Konsep Game dan Sasaran Pengguna) serta spesifikasi fungsionalitas dan desain antarmuka yang lengkap, yang tertuang dalam Game Design Document (GDD).

3.3.1 Konsep Game

Game yang akan dikembangkan memiliki konsep edukasi interaktif yang memanfaatkan teknologi pengenalan suara dan gerakan kepala. Pemain akan dihadapkan pada dua gambar hero Mobile Legends dan diminta untuk menebak hero yang benar berdasarkan suara yang keluar. Suara tersebut akan diputar secara acak, dan pemain harus menggerakkan kepala ke arah gambar hero yang benar (kiri atau kanan). Dengan menggunakan teknologi deteksi gerakan kepala, sistem akan mencocokkan arah gerakan dengan posisi gambar hero yang benar.

3.3.2 Sasaran Pengguna

Sasaran pengguna dari game ini adalah pemain Mobile Legends yang tertarik untuk meningkatkan pengetahuan mereka mengenai hero-hero dalam permainan. Game ini ditujukan untuk pemain berusia 15 tahun ke atas dengan minat pada game mobile dan teknologi interaktif. Pemain yang memiliki keterampilan dasar dalam bermain game akan lebih mudah beradaptasi dengan permainan ini, meskipun game ini juga dapat dinikmati oleh pemain pemula yang ingin mempelajari lebih banyak tentang Mobile Legends.

3.3.3 Game Design Document (GDD)

GDD ini mencakup beberapa aspek penting dalam perancangan game, antara lain:

1. **Alur Cerita:** Tidak ada cerita yang mendalam dalam game ini, karena fokus utamanya adalah pada pengenalan suara hero dan interaksi fisik
2. **Desain Level:** Setiap level akan menampilkan dua hero secara acak dengan suara yang akan diputar.
3. **Antarmuka Pengguna (UI/UX):** Desain antarmuka yang sederhana dengan dua gambar hero dan indikator suara untuk memberikan pengalaman yang intuitif.
4. **Aset Game:** Gambar hero, suara, dan latar belakang yang relevan dengan tema Mobile Legends.

4 Implementasi dan Pengujian

4.1 Proses Pembuatan

Proses pembuatan game dimulai dengan desain antarmuka dan pengembangan fitur-fitur utama, meliputi pemutaran suara hero secara asynchronus dan deteksi gerakan kepala sebagai input jawaban. Bahasa pemrograman Python digunakan bersama dengan pustaka OpenCV untuk deteksi gerakan kepala, Mediapipe Facemesh untuk pengelolaan gambar dengan mendeteksi landmark wajah secara real-time, dan Pygame untuk pengelolaan audio dan efek suara. Pembuatan aset game meliputi pengambilan gambar hero dan rekaman suara yang akan digunakan dalam game.

4.2 Analisis dan Arsitektur Sistem

Terdapat beberapa sub-bab yang menyajikan analisis berupa arsitektur dan alur kerja proyek 'Guess The Voices of The ML Hero'.

4.2.1 Arsitektur Sistem dan Alur Kerja Utama

Dalam proyek ini, dibangun dengan arsitektur modular yang mana program dipecah menjadi beberapa file Python, di mana setiap file punya tanggung jawabnya masing-masing sehingga membuat kode menjadi lebih rapi dan gampang di-maintenance. Seluruh modul diatas dikendalikan oleh main.py yang bertindak sebagai entry point aplikasi. Alur kerja utama dalam proyek ini adalah :

1. **Inisialisasi** : Saat game dimulai, `main.py` akan membuka webcam dengan OpenCV dan menginisiasi model MediaPipe untuk siap mendeteksi wajah pemain
2. **Game Looping** : Terdapat beberapa siklus saat terjadinya game looping :
 - **Pengambilan Soal**: `question_manager.py` dipanggil untuk mengambil set data soal yang unik.
 - **Pemutaran Audio**: `audio_manager.py` memutar suara hero secara asynchronous (non-blocking).
 - **Deteksi Wajah**: Frame dikirim ke MediaPipe FaceMesh untuk deteksi 468 titik landmark pada wajah.
 - **Deteksi Input**: Hasil landmark diteruskan ke `tilt_detector.py` untuk menentukan jawaban pemain berdasarkan kemiringan kepala.
 - **Render UI**: `ui_overlay.py` menggambar antarmuka (pertanyaan dan pilihan) yang posisinya dinamis, selalu mengikuti wajah pemain.
 - **Umpan Balik**: `main.py` dan `video_processor.py` bekerja sama untuk memperbarui skor dan menampilkan gambar hero setelah pemain menjawab.
3. **Layar Akhir** : Setelah selesai, `main.py` menampilkan score dan musik

```

1 import cv2
2 import mediapipe as mp
3 from question_manager import get_question
4 from audio_manager import play_audio, stop_audio
5 from tilt_detector import get_head_tilt_direction
6 from ui_overlay import draw_tiktok_style_overlay
7 from video_processor import draw_result
8 import time
9
10 # Inisialisasi MediaPipe & OpenCV
11 mp_face_mesh = mp.solutions.face_mesh
12 face_mesh = mp_face_mesh.FaceMesh(min_detection_confidence=0.5, min_tracking_confidence=0.5)
13 cap = cv2.VideoCapture(0)
14 cv2.namedWindow("Guess The Hero!")

```

Kode 1: Kode Inisialisasi dan Integrasi Modul (`main.py`)

4.2.2 Analisis Teknologi Computer Vision dan Input

Modul `tilt_detector.py` adalah inti dari proyek karena mengubah gerakan non-konvensional menjadi input game. Teknologi kuncinya adalah `mediapipe.solutions.face_mesh`, yang mampu mendeteksi 468 titik 3D (landmark) pada wajah secara real-time. Landmark ini punya dua fungsi utama: sebagai penentu orientasi kepala (Input Kontrol) dan sebagai titik acuan (anchor) untuk penempatan UI. Ada beberapa logika yang terjadi pada modul ini :

1. Fungsi `get_head_tilt_direction(face_landmarks)` mengambil koordinat (x, y) dari dua titik penting: mata kiri bagian dalam (`landmark[33]`) dan mata kanan bagian dalam (`landmark[263]`).
2. Sudut garis imajiner yang menghubungkan kedua mata dihitung menggunakan fungsi `math.atan2(dy,dx)` dan kemudian dikonversi ke derajat. `math.atan2` dipilih karena kemampuannya menghitung sudut yang akurat di semua kuadran.
3. Sudut tersebut kemudian dibandingkan dengan `tilt_threshold` yang ditetapkan sebesar 15 derajat. Jika sudut di atas 15 derajat, dianggap miring ke kanan. Jika di bawah 15 derajat, dianggap miring ke kiri.

```

1 import math
2
3 def get_head_tilt_direction(face_landmarks):
4     """Mendeteksi kemiringan kepala (kiri, kanan, atau tengah) dari landmark wajah."""
5     if not face_landmarks:
6         return "CENTER"
7
8     # Titik referensi pada mata kiri dan kanan
9     p1 = face_landmarks.landmark[33] # Mata kiri
10    p2 = face_landmarks.landmark[263] # Mata kanan
11
12    # Hitung sudut garis yang menghubungkan kedua mata
13    angle = math.degrees(math.atan2(p2.y - p1.y, p2.x - p1.x))
14
15    # Threshold untuk menentukan kemiringan
16    tilt_threshold = 15
17
18    if angle > tilt_threshold:
19        return "RIGHT" # Sudut positif: miring ke kanan
20    elif angle < -tilt_threshold:
21        return "LEFT" # Sudut negatif: miring ke kiri
22    else:
23        return "CENTER"

```

Kode 2: Kode Deteksi Kemiringan Kepala (`tilt_detector.py`)

4.2.3 Desain Antarmuka Pengguna Dinamis

UI dirancang untuk terlihat floating dan posisinya selalu mengikuti wajah pemain. Hal ini bisa terjadi karena menggunakan modul `ui_overlay.py` yang mana terdapat posisi landmark dahi sebagai pusat referensi `landmark[10]` sehingga semua elemen UI digambar pada posisi yang dihitung secara relatif terhadap titik landmark tersebut. Lalu, disini terdapat fungsi `rounded_box()` untuk menggambar kotak dengan sudut tumpul dengan cara menggabungkan empat persegi panjang dan empat lingkaran di sudut-sudutnya.

```

1 def rounded_box(img, text, center, size, box_color, text_color, radius=20, thickness=-1, icon_path=
  None):
2     """Menggambar kotak dengan sudut bulat, teks, dan ikon opsional."""
3     # ... (Penentuan koordinat x1, y1, x2, y2) ...
4
5     # Gambar 4 persegi panjang untuk membentuk dasar kotak bulat
6     cv2.rectangle(img, (x1 + radius, y1), (x2 - radius, y2), box_color, thickness)
7     cv2.rectangle(img, (x1, y1 + radius), (x2, y2 - radius), box_color, thickness)
8
9     # Gambar 4 lingkaran di sudut-sudut
10    cv2.circle(img, (x1 + radius, y1 + radius), radius, box_color, thickness)
11    cv2.circle(img, (x2 - radius, y1 + radius), radius, box_color, thickness)
12    cv2.circle(img, (x1 + radius, y2 - radius), radius, box_color, thickness)
13    cv2.circle(img, (x2 - radius, y2 - radius), radius, box_color, thickness)
14
15    # ... (Penempatan Teks dan Ikon) ...

```

Kode 3: Kode Implementasi Kotak Sudut Tumpul (`ui_overlay.py`)

4.2.4 Manajemen Audio Asynchronous

Salah satu tantangan dalam aplikasi Computer Vision yang membutuhkan responsivitas tinggi adalah mencegah lag atau "macet" (blocking) saat memutar file audio yang berdurasi panjang (suara hero). Solusinya, pemutaran audio ditangani secara asynchronous (non-blocking). Modul `audio_manager.py`

mengimplementasikan mekanisme **Threading**. Saat `play_audio` dipanggil dengan parameter `block=False`, fungsi ini menciptakan thread baru yang didedikasikan sepenuhnya untuk menjalankan pemutaran audio. Hal ini memungkinkan Thread Utama yang bertanggung jawab atas Computer Vision untuk terus berjalan tanpa jeda, memastikan aplikasi tetap responsif dan bebas lag.

```

1 # Di dalam audio_manager.py
2
3 def play_audio(path, block=False, on_finish=None):
4     """
5     Memutar file audio.
6     :param block: Jika True, eksekusi akan berhenti hingga audio selesai.
7     """
8     if block:
9         # ... (Logika Blocking)
10        pass
11    else:
12        # Putar audio di thread terpisah (non-blocking)
13        import threading
14        thread = threading.Thread(target=_play_audio_threaded, args=(path, on_finish))
15        thread.daemon = True # Thread berhenti jika program utama keluar
16        thread.start()

```

Kode 4: Kode Implementasi Threading untuk Audio (`audio_manager.py`)

4.2.5 Logika Permainan dan Manajemen State

Logika permainan diatur oleh dua modul utama: `question_manager.py` yang berfungsi sebagai database dan pengatur soal dan `main.py` yang mengelola state (keadaan) permainan seperti skor dan status jawaban. Modul ini berisi list statis heroes sebagai database soal, yang mencakup nama, path suara, dan path gambar hero. Inti dari modul ini adalah fungsi `get_question()` yang menjamin tidak ada hero yang muncul berulang kali dalam satu sesi permainan. Soal yang sudah muncul dihapus dari daftar `available_heroes` menggunakan metode `pop()`. Jika `available_heroes` kosong, daftar tersebut diisi ulang (reset) dari list heroes asli dan diacak kembali. Ini memastikan semua hero akan muncul sekali sebelum ada pengulangan.

```

1 # Di dalam question_manager.py
2
3 # ... (Definisi heroes dan available_heroes) ...
4
5 def get_question():
6     """Mengambil satu soal acak, memastikan tidak ada pengulangan hingga semua hero muncul."""
7     global available_heroes
8     # Jika hero sudah habis, reset dan acak kembali
9     if not available_heroes:
10        available_heroes = heroes.copy()
11        random.shuffle(available_heroes)
12
13    # Ambil satu hero sebagai jawaban benar dan hapus dari daftar (pop)
14    correct = available_heroes.pop()
15
16    # ... (Logika penentuan pilihan salah) ...
17
18    return question_data

```

Kode 5: Kode Logika Anti-Pengulangan Soal (`question_manager.py`)

4.2.6 Manajemen Output Visual

Modul `video_processor.py` bertanggung jawab menangani output visual spesifik yang tidak terkait dengan UI dinamis, khususnya menampilkan gambar hero yang benar sebagai umpan balik setelah

pemain menjawab. Terdapat fungsi `draw_result(frame, hero_img_path)` dengan logikanya :

1. Memuat gambar hero dari path yang diberikan menggunakan OpenCV (`cv2.imread`).
2. Mengubah ukuran gambar (`resize`) menjadi 160×160 piksel.
3. Menempatkannya pada posisi yang sudah ditentukan—pojok kanan bawah frame video—dengan jarak (`padding`) 20 piksel dari tepi.

```

1 import cv2
2
3 def draw_result(frame, hero_img_path):
4     """Menampilkan gambar hero di pojok kanan bawah setelah menjawab."""
5     h, w, _ = frame.shape
6
7     hero = cv2.imread(hero_img_path)
8     hero = cv2.cvtColor(hero, cv2.COLOR_BGR2RGB) # Konversi ke RGB
9     hero = cv2.resize(hero, (160,160)) # Ubah ukuran gambar
10
11     # Posisi di pojok kanan bawah dengan padding
12     y_offset = h - 160 - 20 # 20px padding dari bawah
13     x_offset = w - 160 - 20 # 20px padding dari kanan
14     # Menempatkan array gambar hero ke dalam array frame
15     frame[y_offset:y_offset+160, x_offset:x_offset+160] = hero
16
17     return frame

```

Kode 6: Kode Penempatan Visual Hasil Jawaban (`video_processor.py`)

4.3 Pengujian

Pengujian dilakukan untuk memvalidasi fungsionalitas sistem (Alpha Testing) dan mengukur pengalaman pengguna (Beta Testing)

4.3.1 Pengujian Fungsionalitas Sistem (Alpha Testing)

Pengujian ini memastikan bahwa modul utama yang menjawab rumusan masalah berjalan dengan benar, yaitu integrasi pengenalan suara (asynchronous) dan deteksi gerakan kepala (Head Tilt).

1. **Pengujian Modul Deteksi Orientasi Kepala** : Tujuan pengujian ini untuk memverifikasi akurasi modul tilt detector dalam mengklasifikasikan arah kemiringan (Kiri/Kanan) pada ambang sudut ± 15 derajat yang ditetapkan. Metodenya berupa percobaan kemiringan kepala sebanyak 100 kali. Hasilnya menunjukkan bahwa akurasi yang didapat diatas 90 persen dan modul berhasil menggunakan Landmark ID 33 dan ID 263 untuk menghitung roll angle dan mengirimkan sinyal kontrol.
2. **Pengujian Modul Threading Audio (Asynchronous)** : Tujuan dari pengujian ini untuk membuktikan bahwa pemutaran audio hero (.mp3) berjalan secara asinkron (non-blocking), sehingga tidak mengganggu frame rate video real-time. Metodenya dilakukan pengujian fungsional dengan mengamati kinerja game loop saat suara hero diputar dan berfokus pada ada/tidaknya jeda (delay) atau bingkai yang terhenti (freeze) pada aliran video. Hasilnya menunjukkan bahwa saat audio diputar, aliran video dari webcam tetap berjalan lancar dan tidak mengalami freeze atau delay yang signifikan.

4.3.2 Pengujian Pengguna (Beta Testing)

Pengujian ini dilakukan untuk mendapatkan umpan balik mengenai kenyamanan (Usability), kepuasan (Enjoyment), dan pengalaman bermain (User Experience) dengan meminta responden untuk mencoba game dan memberikan umpan balik dalam format wawancara terstruktur atau kuesioner terbuka mengenai pengalaman mereka. Hasilnya berupa :

1. **Kenyamanan Kontrol (Usability)** : Untuk respon positif, mayoritas responden menyatakan bahwa kontrol gerakan kepala (kemiringan) sangat intuitif dan cepat dipelajari tanpa memerlukan panduan yang rumit. Hal ini menunjukkan keberhasilan dalam menyediakan input yang natural. Sedangkan untuk respon negatif, beberapa responden mencatat bahwa di awal permainan, mereka masih perlu membiasakan diri untuk membedakan antara gerakan Kanan/Kiri yang disengaja dan yang tidak disengaja.
2. **Kenyamanan Kontrol (Usability)** : Untuk respon positif, responden secara konsisten menyatakan bahwa interaksi menggunakan gerakan kepala memberikan tingkat imersif yang lebih tinggi dan menyenangkan dibandingkan kuis berbasis sentuhan layar konvensional. Kecepatan feedback visual membantu mempertahankan fokus (Flow). Sedangkan untuk respon negatif, konsentrasi yang diperlukan untuk mempertahankan posisi wajah agar tetap berada di area deteksi terkadang terasa melelahkan dalam sesi bermain yang panjang dikarenakan sensitivitas metode RBR terhadap input gambar berkualitas rendah.

4.4 Hasil Implementasi

Setelah menyelesaikan Tahap Pembuatan (Assembly) dalam MDLC, keseluruhan rancangan sistem telah diimplementasikan dalam bentuk aplikasi game "Guess The Voices of The ML Hero". Setelah tahap implementasi, game dapat berjalan dengan baik, di mana suara hero diputar dengan benar sesuai dengan gambar hero yang ditampilkan. Deteksi gerakan kepala menggunakan webcam berfungsi dengan baik dalam kondisi pencahayaan normal. Namun, perlu adanya perbaikan dalam hal keakuratan deteksi pada kondisi cahaya yang kurang optimal. Selain itu, pengujian sistem menunjukkan bahwa gameplay dapat berjalan dengan lancar pada perangkat dengan spesifikasi sedang.

5 Kesimpulan dan Saran

5.1 Kesimpulan

Dari hasil pengembangan sistem, implementasi, dan pengujian yang dilakukan, dapat disimpulkan bahwa kedua Rumusan Masalah telah terjawab dengan simpulan sebagai berikut:

1. Integrasi Sistem Multimedia Interaktif dan Fungsionalitas : Game edukasi interaktif ini berhasil diwujudkan melalui integrasi teknologi Computer Vision dan Parallel Processing. Hal ini dicapai dengan implementasi pemutaran audio secara asynchronous menggunakan teknik Multithreading yang menjamin game loop utama (video real-time) tetap berjalan lancar (non-blocking). dan penerapan deteksi orientasi kepala (Head Tilt) yang responsif menggunakan MediaPipe Face Mesh dan klasifikasi Rule-Based Reasoning (RBR) pada nilai ambang ± 15 derajat, yang berfungsi sebagai sinyal kontrol. Fungsionalitas gabungan ini telah teruji berhasil di kondisi pencahayaan normal, meskipun terdapat keterbatasan kinerja RBR pada kondisi minim cahaya.
2. Penciptaan Interaksi yang Responsif dan Pengalaman Imersif : Penerapan komponen teknis tersebut berhasil menciptakan interaksi yang responsif dan pengalaman bermain yang imersif. Pengujian Pengguna (Beta Testing) menunjukkan bahwa kontrol gerakan kepala dirasakan intuitif dan menyenangkan oleh mayoritas responden, memvalidasi keberhasilan penerapan prinsip desain Flow dan UI/UX dalam menghasilkan gameplay yang menarik dan terlibat.

5.2 Saran

Untuk pengembangan lebih lanjut, disarankan untuk memperbaiki deteksi gerakan kepala dengan menggunakan teknologi yang lebih canggih dari metode RBR, seperti deep learning, untuk meningkatkan akurasi pada kondisi pencahayaan yang buruk. Selain itu, penambahan fitur-fitur lain, seperti mode permainan berbasis waktu atau integrasi dengan media sosial, dapat meningkatkan daya tarik dan keterlibatan pemain.

Daftar Pustaka

- [1] E. Lilis et al., "Gamification in Education: Innovative Solutions through Learning System Development to Increase Student Motivation and Interest," *Socius: Jurnal Penelitian Ilmu-Ilmu Sosial*, vol. 2, no. 4, pp. 125-131, Nov. 2024.
- [2] R. F. A. Ginting, F. E. Messe, "PENGEMBANGAN GAME EDUKASI PUZZLE BERBASIS COMPUTER VISION HAND TRACKING UNTUK ANAK USIA DINI," *HOAQ: JURNAL TEKNOLOGI INFORMASI*, vol. 16, no. 1, pp. 16-23, Mei 2025.
- [3] M. F. Haikal, F. Utaminigrum, S. Adinugroho, "DETEKSI ORIENTASI WAJAH MINIM PENCAHAYAAN PADA KURSI RODA PINTAR MENGGUNAKAN MEDIAPIPE, CLAHE DAN METODE RBR," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 9, no. 8, Ags. 2025.
- [4] K. W. Widjanarko, K. A. Herlambang, M. A. Karim, "Faster Video Processing Menggunakan Teknik Parallel Processing Dengan Library OpenCV," *Jurnal Komunikasi, Sains dan Teknologi*, vol. 1, no. 1, pp. 10-18, Jun. 2022.
- [5] B. Cowley, D. Charles, M. Black, R. Hickey, "Toward an Understanding of Flow in Video Games," *ACM Computers in Entertainment*, vol. 3, no. 3, 2005.
- [6] R. Mayasari, N. Heryana, *Konsep dan Teori Desain User Experience Perangkat Lunak*. Karawang: PT. Neo Santara Indonesia, 2023.