



Nama: 1. Marchel Ferry Timoteus S [121140195]
2. Silva Oktaria Putri [122140085]
2. Irma Amelia Novianti [122140128]
Tugas Ke: **Tugas Besar**
Mata Kuliah: **Sistem / Informasi Multimedia (IF4021)**

Tanggal: 31 Mei 2025

1 Deskripsi Projek

Suara Menentukan Nasib adalah sebuah program permainan sederhana yang dibuat menggunakan Python, terinspirasi dari adegan “Red Light, Green Light” di serial *Squid Game*. Dalam permainan ini, karakter Mario bergerak berdasarkan intensitas suara yang dideteksi melalui mikrofon, namun hanya dapat bergerak jika tangan dan kepala pemain terdeteksi oleh kamera secara *real-time*. Saat lampu merah menyala, Mario tidak boleh bergerak dan jika suara terlalu keras maka akan langsung tereliminasi, sedangkan saat lampu hijau menyala, Mario dapat maju selama intensitas suara cukup kuat. Permainan memiliki batas waktu untuk mencapai garis akhir, dan pada akhir sesi, sistem menampilkan notifikasi “Permainan Berakhir” dengan dua opsi yaitu **Play Again** untuk memulai ulang atau **Exit** untuk keluar dari permainan.

2 Teknologi

Berikut adalah teknologi-teknologi yang dimanfaatkan dalam pengerjaan *Final Project* ini:

- Python merupakan bahasa pemrograman yang bersifat interpreted, high-level, dan serbaguna. Bahasa ini dirancang dengan filosofi yang menekankan keterbacaan kode melalui penggunaan indentasi yang konsisten.
- Pygame adalah pustaka Python lintas *platform* yang ditujukan untuk pengembangan gim video. Pustaka ini menyediakan fungsi untuk mengelola grafis dan audio, yang dapat digunakan bersamaan dengan Python.
- MediaPipe adalah framework lintas platform yang digunakan untuk membangun pipeline machine learning dalam aplikasi multimedia. Framework ini memungkinkan deteksi objek, wajah, pelacakan tangan, serta identifikasi pose.
- CV2 digunakan untuk menangkap video melalui *webcam*. OpenCV sendiri merupakan pustaka pemrosesan citra dan video yang sangat powerful, mendukung berbagai fungsi seperti deteksi objek, pengenalan wajah, dan banyak fitur lainnya.

3 Cara Kerja

Cara kerja pada proyek *suara menentukan nasib* adalah sebagai berikut:

- Sistem menggunakan kamera *webcam* untuk menangkap video secara langsung.
- Sistem menggunakan *library* seperti Mediapipe yang mana mendeteksi tangan dan melacak posisi *landmark* seperti jari / telapak tangan.
- Posisi tangan dalam video dikonversi ke koordinat yang dapat digunakan dalam *game*, misalnya untuk memulai game.
- layar akan menampilkan keterangan jika 'S' untuk memulai, dan Spasi untuk Pause.
- game pada permainan akan mendeteksi suara jika terlalu keras maka lampu merah menyala, mario akan langsung tereliminasi.
- sebaliknya saat lampu hijau menyala dan suara cukup kuat, maka mario akan bergerak maju.
- setelah pemain menyelesaikan game, akan muncul pilihan "play Again" atau "Exit".

4 Penjelasan Kode Program

4.1 Menggunakan Webcam dan MediaPipe

```
1 import cv2
2 import mediapipe as mp
3
4 class InputHandler:
5     def __init__(self):
6         self.cap = cv2.VideoCapture(0)
7         self.mp_pose = mp.solutions.pose
8         self.pose = self.mp_pose.Pose()
9
10    def read_frame(self):
11        ret, frame = self.cap.read()
12        if not ret:
13            return None, None
14
15        frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
16        results = self.pose.process(frame_rgb)
17        landmarks = results.pose_landmarks.landmark if results.pose_landmarks else None
18
19        return frame, landmarks
20
21    def release(self):
22        self.cap.release()
```

Penjelasan Fungsi:

- `init_(self)` : Konstruktor kelas, membuka webcam dan menginisialisasi Mediapipe pose detector.
- `read_frame(self)` : Mengambil satu frame dari webcam, memprosesnya untuk mendapatkan landmark pose. Mengembalikan dua nilai: frame video asli (untuk ditampilkan), dan landmark pose.
- `release(self)` : Fungsi pembersihan yang penting agar webcam ditutup dengan benar saat program selesai.

Penjelasan Alur Program :

- Saat game menunggu start, tampilkan teks "Tekan S untuk Memulai".
- Saat game berjalan, tekan spasi dapat menghentikan (pause) game
- Kondisi ini menangani event keyboard dari pygame.

4.2 Menampilkan Keterangan 'S' untuk Start dan Spasi untuk Pause

```
1 import pygame
2
3 def handle_events():
4     for event in pygame.event.get():
5         if event.type == pygame.QUIT:
6             return 'exit'
7         if event.type == pygame.KEYDOWN:
8             if event.key == pygame.K_s:
9                 return 'start'
10            elif event.key == pygame.K_SPACE:
11                return 'pause'
12            elif event.key == pygame.K_ESCAPE:
13                return 'exit'
14        return 'none'
15
16 # Dalam game loop:
17 status = handle_events()
18 if status == 'start':
19     is_running = True
20 elif status == 'pause':
21     is_paused = not is_paused
22 elif status == 'exit':
23     running = False
```

Penjelasan:

- `handle_events()` : menggunakan loop untuk mengambil semua event input di pygame.
- `pygame.QUIT` : mendeteksi klik tombol close window, menginstruksikan keluar dari game.
- `pygame.KEYDOWN` menangani event tekan tombol keyboard, merespon tombol S, SPACE, dan ESCAPE
- Saat game menunggu start, tampilkan teks "Tekan S untuk Memulai".
- Saat game berjalan, tekan spasi dapat menghentikan (pause) game.
- Mengembalikan string status yang digunakan oleh game loop untuk mengubah kondisi game.

4.3 Deteksi Level Suara Mikrofon dan Kontrol Lampu

```
1 import sounddevice as sd
2 import numpy as np
3
4 def get_audio_level(duration=0.1, fs=44100):
5     # Rekam suara selama durasi singkat (misal 0.1 detik)
6     recording = sd.rec(int(duration * fs), samplerate=fs, channels=1)
7     sd.wait() # Tunggu hingga rekaman selesai
8
9     # Hitung Root Mean Square (RMS) sebagai indikator volume suara
10    rms = np.sqrt(np.mean(recording**2))
11    return rms
```

Penjelasan Fungsi :

- `get_audio_level()` : digunakan untuk merekam audio selama 0.1 detik (bisa diubah)
- `sounddevice` : digunakan untuk mengakses microphone
- `rms` digunakan sebagai ukuran intensitas suara. `rms` yang lebih besar artinya suara lebih keras
- `THRESHOLD` : nilai ini dibandingkan dengan nilai `rms` untuk menentukan apakah suara cukup keras.

Penjelasan Alur Program :

- Audio diambil secara real-time dari mikrofon
- jika suara melebihi threshold saat lampu hijau, Mario bergerak maju.
- Jika suara melebihi threshold saat lampu merah, Mario langsung kalah dan lampu merah menyala.
- Jika suara di bawah threshold saat lampu hijau, Mario diam.

4.4 Mario Bergerak Maju Jika Lampu Hijau dan Suara Cukup Kuat

```
1 class Player:
2     def __init__(self):
3         self.x = 0
4         self.speed = 5
5         self.is_moving = False
6
7     def move_forward(self):
8         self.is_moving = True
9         self.x += self.speed
10        if self.x > 580:
11            self.x = 580 # batas finish
12
13    def stop(self):
14        self.is_moving = False
```

Penjelasan Fungsi :

- `move_forward()` : Memajukan posisi Mario di sumbu X sebanyak speed piksel
- Membatasi posisi maksimal supaya Mario tidak keluar dari layar.
- `stop()` : mengubah status `is_moving` menjadi False supaya Mario diam.

Penjelasan Alur Program :

- Saat lampu hijau menyala, dan suara yang masuk lebih dari threshold, Mario maju dengan kecepatan tertentu.
- Jika suara tidak cukup keras, Mario diam (tidak bergerak).

4.5 Mario Tereliminasi Jika Lampu Merah dan Suara Terlalu Keras

```
1 if lampu_merah and audio_level > THRESHOLD:
2     sound_manager.play_sound('red_light')
3     sound_manager.play_sound('lose')
4     game_over = True
5     result = 'lose'
```

Penjelasan Fungsi :

- Kondisi ini dicek terus dalam game loop.
- Jika kondisi terpenuhi, suara lampu merah dan efek kalah dimainkan.
- **game_over** mengindikasikan loop utama game harus berhenti dan masuk ke tampilan akhir.
- **result** : menyimpan apakah pemain menang atau kalah

Penjelasan Alur Program :

- Saat lampu merah menyala, dan suara melebihi threshold, Mario langsung kalah.
- Game masuk status game over dan menampilkan suara lampu merah serta suara kalah.

4.6 Play Again atau Exit Setelah Game Selesai

```
1 def display_game_over(screen, result):
2     font = pygame.font.SysFont(None, 48)
3     if result == 'win':
4         text = font.render("Anda Menang! (Y) Play Again / (N) Exit", True, (0, 255, 0))
5     else:
6         text = font.render("Anda Kalah! (Y) Play Again / (N) Exit", True, (255, 0, 0))
7     screen.blit(text, (50, 200))
8     pygame.display.flip()
9
10    # Loop event untuk game over
11    while game_over:
12        for event in pygame.event.get():
13            if event.type == pygame.KEYDOWN:
14                if event.key == pygame.K_y:
15                    # Restart game
16                    game_over = False
17                    main() # atau panggil ulang loop utama game
18                elif event.key == pygame.K_n or event.key == pygame.K_ESCAPE:
19                    pygame.quit()
20                    exit()
```

Penjelasan Fungsi :

- **display_game** : menggambarkan teks pesan hasil akhir.
- **Y** : mengulang permainan di awal
- **N** atau **ESC** : Keluar program
- **pygame.display.flip()** : memperbarui layar dengan pesan hasil.

Penjelasan Alur Program :

- Setelah pemain menang atau kalah, game menampilkan pesan akhir dan pilihan.
- : Jika pemain tekan tombol tertentu (misal Y untuk main lagi, N untuk keluar), maka game restart atau keluar.

5 Hasil Analisis

5.1 Program Utama

Bagian program utama ini bertanggung jawab untuk menjalankan permainan. Berikut adalah implementasi dari bagian tersebut:

```
1 if __name__ == "__main__":
2     play_again = True
3     while play_again:
4         game = Game() # Buat instance dari class Game
5         if game.is_running:
6             play_again = game.run()
7         else:
8             play_again = False # Don't try to play again
9     pygame.quit()
```

Kode 1: Program Utama Permainan

5.2 Deskripsi Kode

Kode di atas melakukan beberapa langkah penting sebagai berikut:

1. Memeriksa Nama Modul:

Kode dimulai dengan pernyataan `if __name__ == "__main__":` yang memeriksa apakah skrip ini dijalankan sebagai program utama. Hal ini memastikan bahwa bagian kode ini hanya dieksekusi ketika file ini dijalankan secara langsung, bukan ketika diimpor sebagai modul dalam skrip lain.

2. Loop Permainan:

Variabel `play_again` digunakan sebagai flag untuk menentukan apakah permainan akan diulang. Loop `while play_again:` akan terus berjalan selama nilai `play_again` bernilai `True`, memungkinkan pemain untuk memilih opsi "Play Again" atau "Exit" setelah sesi permainan selesai.

3. Membuat Instance Game:

Dalam setiap iterasi loop, sebuah instance dari kelas `Game` dibuat menggunakan `game = Game()`. Ini memastikan bahwa setiap kali permainan dimulai ulang, objek `Game` baru dibuat dengan kondisi awal yang bersih.

4. Menjalankan Permainan:

Metode `game.run()` dipanggil untuk menjalankan logika utama permainan. Metode ini mengatur semua aspek permainan, termasuk deteksi input suara, gerakan karakter Mario, pengendalian waktu, serta pengecekan kondisi akhir permainan. Setelah permainan selesai, metode ini mengembalikan nilai boolean (`True` jika pemain ingin memainkan lagi, `False` jika ingin keluar), yang digunakan untuk memperbarui variabel `play_again`.

5. Keluar dari Pygame:

Setelah loop selesai (yaitu ketika `play_again` menjadi `False`), fungsi `pygame.quit()` dipanggil untuk menghentikan Pygame dan membersihkan sumber daya yang digunakan oleh modul Pygame.

Langkah-Langkah Spesifik

Pengecekan `__main__`: `if __name__ == "__main__":`

Pernyataan ini memastikan bahwa blok kode di bawahnya hanya dieksekusi ketika file ini dijalankan secara langsung, bukan saat diimpor sebagai modul.

Inisialisasi Loop Permainan: `play_again = True`

```
2 while play_again:
```

Variabel `play_again` digunakan sebagai kontrol ulang permainan. Loop ini akan terus berjalan selama `play_again` bernilai `True`.

Membuat Instance Game: `game = Game()`

Sebuah instance dari kelas `Game` dibuat untuk setiap sesi permainan. Kelas `Game` mungkin mengandung logika inti seperti inisialisasi karakter, deteksi suara, dan kontrol waktu.

Jalankan Logika Permainan: `if game.is_running:`

```
2     play_again = game.run()
```

Metode `game.run()` menjalankan logika utama permainan, termasuk deteksi suara, gerakan karakter, dan pengecekan kondisi akhir permainan. Metode ini mengembalikan nilai boolean yang menentukan apakah pemain ingin memainkan lagi (`True`) atau keluar dari permainan (`False`).

Keluar dari Pygame: `pygame.quit()`

Setelah loop selesai, fungsi `pygame.quit()` dipanggil untuk menghentikan Pygame dan membersihkan sumber daya yang digunakan.

6 Implementasi Program

6.1 Mendownload seluruh isi project dari GitHub ke komputer lokal

- Buka terminal atau command prompt
- Jalankan perintah berikut :

```
1 git clone https://github.com/marsselferrys/Tubes_Mulmet.git
```

- Masuk ke dalam direktori proyek

```
1 cd Tubes_Mulmet
```

Perintah `git clone` akan menyalin semua file dari repositori ke folder lokal, lalu perintah `cd` digunakan untuk masuk ke dalam direktori proyek tersebut.

6.2 Buat dan aktifkan virtual environment (Opsional)

- Untuk Linux/Mac :

```
1 uv venv
2 source venv/bin/activate
```

- Untuk Windows

```
1 uv venv
2 .venv\Scripts\activate
```

Aktifkan virtual environment agar paket python yang digunakkan tidak bercampur dengan sistem global. Virtual environment adalah lingkungan Python terpisah, yang membantu mencegah konflik antar proyek.

6.3 Install Semua Depedency

Setelah environment aktif, instal semua paket yang dibutuhkan dengan menjalankan:

```
1 pip install -r requirements.txt
2
3 uv pip install -r requirements.txt // jika menggunakan uv
```

Perintah ini akan membaca file requirements.txt dan menginstal semua library seperti opencv-python, numpy, mediapipe, dan lainnya yang diperlukan program agar bisa berjalan.

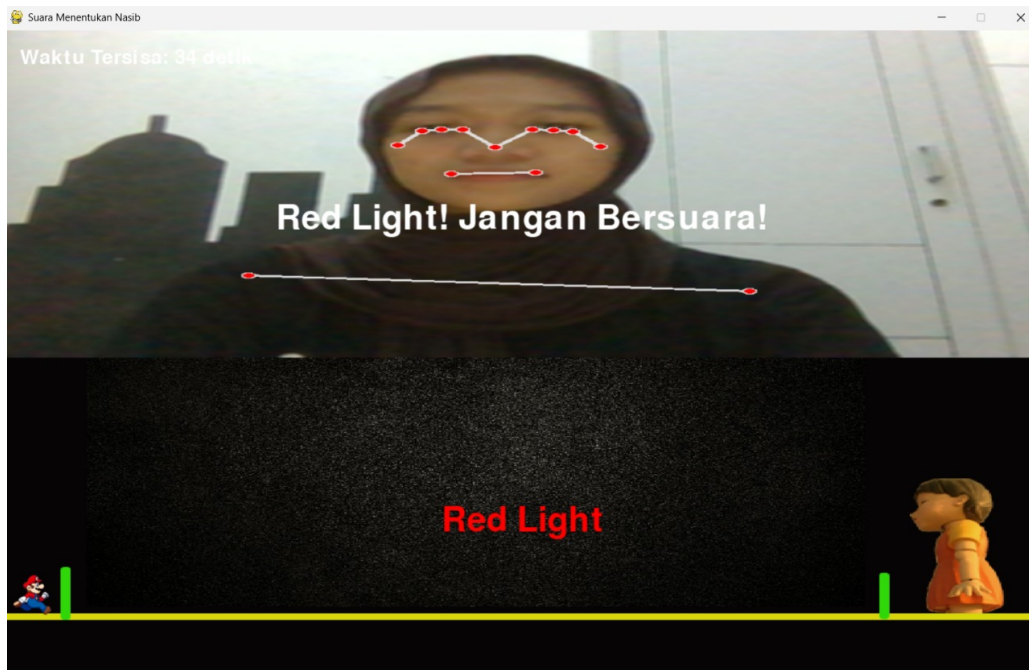
6.4 Jalankan Program

```
1 python main.py
```

File main.py akan menjalankan program secara real-time Rekan tombol "S" untuk memulai program



Gambar 1: Tampilan awal antarmuka



Gambar 2: Pemain Dilarang Bersuara



Gambar 3: Pemain harus bersuara



Gambar 4: Permainan Berakhir

- layar akan menampilkan keterangan jika 'S' untuk memulai, dan Spasi untuk Pause.
- game pada permainan akan mendeteksi suara jika terlalu keras maka lampu merah menyala, mario akan langsung tereliminasi.
- sebaliknya saat lampu hijau menyala dan suara cukup kuat, maka mario akan bergerak maju.
- setelah pemain menyelesaikan game, akan muncul pilihan "play Again" atau "Exit".