



---

Nama:

1. Machzaul Harmansyah (122140172)
2. Putri Diana Sari Rambe (122140052)
3. Debora Sihombing (122140032)

Tugas : **Final Project**

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

Tanggal: **Sabtu, 31 Mei 2025**

---

## 1 Deskripsi Proyek

Proyek ini adalah sebuah game interaktif berbasis Python yang memanfaatkan teknologi computer vision dan augmented reality. Pemain menangkap bola virtual berwarna menggunakan keranjang virtual yang posisinya mengikuti area hidung. Bola muncul secara acak dengan warna tertentu, dan pemain harus menggerakkan kepala atau tubuh ke kiri, kanan, atau tengah untuk mengarahkan keranjang agar bisa menangkap bola yang sesuai warna instruksi. Game ini menggunakan webcam sebagai input video real-time untuk melacak gerakan wajah dan tubuh pemain. Sistem juga memakai pengenalan warna agar hanya bola dengan warna yang tepat yang dihitung. Kalau pemain salah menangkap bola warna yang tidak sesuai, permainan langsung selesai. Skor dihitung berdasarkan jumlah bola yang berhasil ditangkap dengan benar. Dengan konsep sederhana ini, game menggabungkan teknologi pengolahan video, interaksi gerakan tubuh, dan pengenalan warna, sehingga menciptakan pengalaman bermain yang seru dan menantang.

## 2 Fitur Utama

- Face Tracking Real-time - Menggunakan MediaPipe untuk deteksi wajah
- Multi-color Ball System - 4 warna bola berbeda (Merah, Biru, Hijau, Kuning)
- Dynamic Difficulty - Kecepatan bola meningkat seiring waktu
- Score System - Sistem poin berdasarkan bola yang berhasil ditangkap
- Rich Graphics - Antarmuka visual yang menarik dengan gambar PNG
- Interactive Menu - Menu lobby untuk pemilihan warna bola

## 3 Metode dan Teknologi

Teknologi yang digunakan dalam mengerjakan Final Project ini adalah sebagai berikut:

- Bahasa Pemrograman Python. Python digunakan sebagai bahasa pemrograman utama dalam proyek ini karena kemudahan dalam penulisan dan keterbacaan kode. Bahasa ini dikenal memiliki sintaks yang sederhana dan terstruktur dengan indentasi yang jelas, sehingga memudahkan pengembangan serta pemeliharaan program. Selain itu, Python memiliki banyak pustaka yang mendukung berbagai kebutuhan pemrograman, termasuk pengolahan citra dan pembelajaran mesin, yang relevan dengan proyek ini.

- OpenCV (cv2). OpenCV merupakan pustaka open-source yang sangat populer untuk pengolahan gambar dan video. Dalam proyek ini, OpenCV digunakan untuk mengakses webcam secara real-time, menangkap dan menampilkan frame video secara berkelanjutan. Selain itu, OpenCV memudahkan manipulasi gambar seperti mengubah ukuran, membalik, dan mengonversi warna. OpenCV juga digunakan untuk menambahkan berbagai elemen grafis seperti teks, bentuk, dan gambar dengan transparansi, sehingga tampilan video menjadi lebih interaktif dan menarik.
- MediaPipe. MediaPipe adalah framework lintas platform yang fokus pada pengembangan aplikasi pembelajaran mesin multimodal. Pada proyek ini, MediaPipe Face Mesh digunakan untuk mendeteksi serta melacak landmark wajah secara real-time, khususnya ujung hidung. Informasi posisi ini digunakan sebagai kontrol untuk menggerakkan objek virtual dalam aplikasi. MediaPipe juga memiliki mekanisme untuk memastikan tingkat kepercayaan deteksi wajah tetap tinggi, sehingga hasil pelacakan dapat diandalkan selama interaksi berlangsung.
- Random. Modul random pada Python berfungsi untuk menghasilkan nilai acak yang memberikan variasi pada permainan. Posisi kemunculan bola dan warna bola diacak secara dinamis, sehingga menciptakan tantangan yang beragam dan menghindari pola yang mudah ditebak. Selain itu, modul ini juga mengatur interval waktu dan kecepatan kemunculan bola, menjaga permainan tetap menarik dan tidak monoton.
- Time. Modul time berperan dalam pengelolaan waktu selama permainan berlangsung. Modul ini digunakan untuk mengatur timer permainan selama 60 detik sebagai batas waktu sesi. Selain itu, modul time mengatur jeda waktu kemunculan bola secara berkala dan mengukur durasi berbagai aktivitas dalam permainan. Data waktu ini juga digunakan untuk menyesuaikan tingkat kesulitan dengan mengatur kecepatan permainan secara bertahap.

## 4 Cara Kerja

Cara kerja pada Simulasi Penangkapan Objek Virtual Tracking Pada Bagian Tubuh:

- Sistem menggunakan kamera webcam untuk menangkap video secara langsung.
- MediaPipe digunakan untuk mendeteksi pose tubuh, khususnya bagian hidung, sebagai titik kontrol utama.
- Posisi hidung dalam video dikonversi menjadi koordinat yang dapat digunakan dalam permainan sebagai posisi keranjang.
- Sebelum permainan dimulai pemain akan diarahkan untuk memilih warna bola menggunakan pendeteksi kepala.
- Permainan akan dimulai secara acak dari atas layar dan turun dengan kecepatan tertentu dan waktu permainan akan dimulai.
- Pemain harus menggerakkan tubuh ke kiri, tengah, atau kanan untuk mengarahkan keranjang ke jalur bola.
- Semakin lama pemain bertahan dalam permainan maka semakin cepat bola akan turun dari atas layar
- Hanya bola dengan warna target yang boleh ditangkap. Jika bola dengan warna lain tertangkap, permainan berakhir.
- Setiap bola benar ditangkap akan menambah skor, dan jika skor mencapai batas tertentu, pemain dinyatakan menang.

## 5 Cara Menjalankan Game

### 1. Quick Start

```
1 <div style="background-color: #d4edda; padding: 16px; border-radius: 6px; border-left
: 4px solid #28a745;">
2 <strong> Jalankan Game:</strong>
3 </div>
4
```

Kode 1: Quick Start

```
1 # Pastikan berada di direktori proyek
2 cd Simulasi-Interaktif-Penangkapan-Objek-Virtual-dengan-tracking-bagian-tubuh
3 # Jalankan game
4 python main.py
5
```

Kode 2: Quick Start

### 2. Troubleshooting Launch Issues

```
1 <div style="background-color: #f8d7da; padding: 16px; border-radius: 6px; border-left
: 4px solid #dc3545;">
2 <strong> Jika mengalami error:</strong>
3 </div>
4
```

Kode 3: Troubleshooting Launch Issues

- Kamera tidak terdeteksi:

```
1 # Cek apakah kamera tersedia
2 python -c "import cv2; print('Kamera OK' if cv2.VideoCapture(0).isOpened() else '
Kamera Error')"
```

Kode 4: Kamera tidak terdeteksi

- File gambar tidak ditemukan:

Pastikan semua file .png ada di direktori yang sama dengan main.py Download assets yang hilang dari repository

- ModuleNotFoundError:

```
1 # Install ulang dependencies
2 pip install --upgrade opencv-python mediapipe numpy
```

Kode 5: ModuleNotFoundError

## 6 Cara Bermain

Langkah-langkah bermain :

### 1. Mulai Game

- Jalankan python main.py

- Tekan Enter di layar home untuk lanjut

## 2. Pilih Warna Bola

- Di layar lobby, arahkan kepala ke salah satu bola warna
- Tahan posisi selama 2 detik untuk memilih

## 3. Gameplay

- Gerakkan kepala untuk mengontrol keranjang
- Tangkap hanya bola dengan warna yang dipilih
- Hindari bola dengan warna lain (akan game over)
- Bertahan selama 60 detik untuk menang

## 7 Kontrol Game

- Tombol Enter berfungsi untuk memulai game dari home screen
- Tombol ESC berfungsi untuk keluar dari game
- Tombol R berfungsi untuk restart game (di layar game over)
- Gerakan Kepala berfungsi untuk mengontrol keranjang penangkap

## 8 Penjelasan Kode Program

Kode program yang digunakan dalam pembuatan Penangkapan Objek Virtual Tracking Pada Bagian Tubuh menggunakan main.py. Berikut ini merupakan penjelasan dari masing-masing bagian code:

### 1. main.py

```
1  #Instalasi Modul
2  import cv2
3  import mediapipe as mp
4  import random
5  import time
```

Kode 6: Instalansi Modul

Peran yang digunakan pada kode tersebut adalah cv2 yang berfungsi sebagai Penangkap video dari kamera, menampilkan UI dan manipulasi gambar, import mediapipe as mp mendeteksi landmark wajah menggunakan Face Mesh, import random sebagai Menghasilkan posisi, warna, dan delay bola secara acak dan import time sebagai timer permainan dan durasi seleksi bola.

```
1  #Inisialisasi MediaPipe Face Mesh
2  mp_face_mesh = mp.solutions.face_mesh
3  face_mesh = mp_face_mesh.FaceMesh(min_detection_confidence=0.5, min_tracking_confidence=0.5)
```

Kode 7: Mendeteksi posisi wajah dan kontrol permainan

Kode diatas digunakan sebagai menginisialisasi dan mempersiapkan model Face Mesh dari MediaPipe , yaitu library machine learning yang dikembangkan oleh Google. Fungsinya adalah untuk mendeteksi titik-titik wajah (landmark) pada dari webcam.

```

1  #Akses Webcam
2  cap = cv2.VideoCapture(0)

```

Kode 8: Akses Webcam

Baris yang digunakan sebagai akses kamera yang dapat membuka perangkat webcam yang dapat digunakan sebagai pendeteksi wajah dan menampilkan UI pada permainan

```

1  #Muat gambar bola berdasarkan warna
2  keranjang_img = cv2.imread('keranjang.png', cv2.IMREAD_UNCHANGED)
3  keranjang_img = cv2.resize(keranjang_img, (150, 150))
4
5  ball_images = {
6      (255, 0, 0): cv2.imread('bolabiru.png', cv2.IMREAD_UNCHANGED),
7      (0, 255, 255): cv2.imread('bolakuning.png', cv2.IMREAD_UNCHANGED),
8      (0, 255, 0): cv2.imread('bolahijau.png', cv2.IMREAD_UNCHANGED),
9      (0, 0, 255): cv2.imread('bolamerah.png', cv2.IMREAD_UNCHANGED)
10 }

```

Kode 9: Membuat Gambar

Kode ini digunakan sebagai elemen visual yang digunakan sebagai keranjang dan bola untuk menjalankan permainan, keranjang sebagai "cursor" yang dikendalikan oleh pengguna dan menghindari bola-bola yang ditangkap sesuai dengan warna yang di pilih

```

1  #Ubah Ukuran Gambar Bola
2  for color in ball_images:
3      ball_images[color] = cv2.resize(ball_images[color], (60, 60))
4
5  ball_radius = 30
6  colors = list(ball_images.keys())
7

```

Kode 10: Akses Webcam

Kode diatas merupakan konfigurasi awal dan variabel-variabel permainan yang digunakan sebelum game dimulai. Pertama, semua gambar bola yang telah dimuat sebelumnya diubah ukurannya menjadi 60x60 piksel agar sesuai dengan tampilan layar dan kebutuhan visual dalam game.

```

1  # Game variables
2  score = 0
3  game_over = False
4  game_started = False
5  show_lobby = False
6  show_menu = False
7  selected_ball_color = None
8  balls = []
9  total_game_time = 60
10 start_time = 0
11 lobby_selection_time = 0
12 selection_duration = 2.0 # 2 detik untuk memilih
13

```

Kode 11: Game Variabel

Kode diatas merupakan pendeklarasi variabel-variabel utama dalam permainan seperti skor, status permainan, warna bola yang dipilih, serta list bola dan mengatur waktu total permainan selama 60 detik untuk menentukan permainan akan dimulai

```

1  # Konfigurasi untuk spacing bola
2  MIN_BALL_DISTANCE = 120 # Jarak minimum antar bola
3  MAX_SPAWN_ATTEMPTS = 50 # Maksimum percobaan spawn untuk menghindari infinite loop
4

```

Kode 12: Konfigurasi spacing bola

Kode rini berfungsi sebagai pencegah bola menunjukkan terlalu dekat antara satu dengan yang lain yang diatur sebesar 120 piksel anatara jarak bola sebanyak 50 kali percobaan untuk memuat bola baru

```

1  # Konfigurasi kecepatan
2  BASE_SPEED_MIN = 3
3  BASE_SPEED_MAX = 6
4  SPEED_INCREASE_RATE = 0.5 # Peningkatan kecepatan per 10 detik
5  MAX_SPEED_TIME = 40 # Waktu maksimum untuk peningkatan kecepatan (detik)
6

```

Kode 13: Konfigurasi kecepatan

Kode diatas berfungsi sebagai konfigurasi kecepatan bola dalam rentang kecepaqtan awal yang terus meningkat seiring berjalannya waktu permainan, kecepatan dibatasi hingga 40 detik agar permainan tidak terlalu sulit

```

1  # Mapping warna untuk tampilan
2  color_names = {
3      (255, 0, 0): "BIRU",
4      (0, 255, 255): "KUNING",
5      (0, 255, 0): "HIJAU",
6      (0, 0, 255): "MERAH"
7  }
8

```

Kode 14: Tampilan Warna

Kode diatas berfungsi sebagai pendefinisi data dan fungsi yang digunakan dalam pengaturan awal permainan serta mengatur logika peningkatan kesulitan yang membantu menampilkan informasi warna bola dengan benar pada pemain agar mudah digunakan pemain.

```

1  #Pengaturan Skor
2  def reset_game():
3      global score, game_over, game_started, show_menu, show_lobby, selected_ball_color, balls,
4          start_time
5          score = 0
6          game_over = False
7          game_started = False
8          show_menu = False
9          show_lobby = False
10         selected_ball_color = None
11         balls = []
12

```

Kode 15: Skor

Kode diatas digunakan untuk mereset variabel permainan ke tampilan awal saat game baru dimulai ulang, serta mengosongkan daftar bola dari awal tanpa memperlihatkan data sebelumnya

```

1  def get_current_speed_range():
2      #Menghitung range kecepatan berdasarkan waktu yang telah berlalu
3      if not game_started:
4          return BASE_SPEED_MIN, BASE_SPEED_MAX
5
6      elapsed_time = time.time() - start_time
7
8      # Batasi peningkatan kecepatan sampai 40 detik
9      capped_time = min(elapsed_time, MAX_SPEED_TIME)
10
11     # Hitung multiplier berdasarkan waktu (setiap 10 detik kecepatan naik)
12     speed_multiplier = 1 + (capped_time // 10) * SPEED_INCREASE_RATE
13
14     min_speed = int(BASE_SPEED_MIN * speed_multiplier)

```

```

15     max_speed = int(BASE_SPEED_MAX * speed_multiplier)
16
17     return min_speed, max_speed
18

```

Kode 16: Kecepatan Bola Permainan

Kode digunakan sebagai rentang kecepatan bola berdasarkan waktu permainan, semakin lama permainan berlangsung hingga mencapai batas maksimum waktu, fungsi kode ini bertujuan pemain menjadi semakin menantang seiring waktu.

```

1 def check_ball_collision(new_x, new_y, existing_balls):
2     for ball in existing_balls:
3         distance = ((new_x - ball["x"])**2 + (new_y - ball["y"])**2)**0.5
4         if distance < MIN_BALL_DISTANCE:
5             return True
6     return False
7
8 def create_ball():
9     for attempt in range(MAX_SPAWN_ATTEMPTS):
10        x = random.randint(...)
11        if not check_ball_collision(x, y, balls):
12            return {"x": x, "y": y, ...}
13

```

Kode 17: Check Ball Create Ball

Kode diatas berfungsi untuk memastikan bola tidak muncul terlalu dekat antara yang lain dan membuat bola baru dengan posisi acak menggunakan beberapa percobaan untuk menghindari pada infinite loop

```

1 # Fungsi layar awal, pemilihan bola dan UI
2 def draw_start_screen(frame, iw, ih): ...
3 def draw_lobby_screen(frame, iw, ih, nose_tip_x, nose_tip_y): ...
4 def draw_game_ui(frame, iw, remaining_time): ...
5 def draw_end_screen(frame, iw, ih, win): ...
6

```

Kode 18: Antarmuka Pemain

Kode diatas adalah layar awal sebagai permainan, menampilkan layar pemilihan bola, menampilkan UI saat bermain dan menampilkan layar akhir (menang/kalah) dalam permainan.

```

1 # Logika Deteksi Tabrakan dan Poin
2 dx = ball["x"] - nose_tip_x
3 dy = ball["y"] - nose_tip_y
4 if dx * dx + dy * dy < ball_radius ** 2:
5     if ball["color"] == selected_ball_color:
6         score += 1
7         ...
8     else:
9         game_over = True
10

```

Kode 19: Logika Permainan

Fungsi kode ini merupakan bagian pendeteksi bola tertangkap oleh keranjang pada bagian deteksi hidung serta memberikan poin jika warna bola yang tertangkap keranjang benar serta mengakhiri jika bola yang ditangkap salah

```

1 # Mengelola alur permainan
2 while cap.isOpened():
3     success, frame = cap.read()
4     ...
5     # Update posisi hidung
6     results = face_mesh.process(rgb_frame)

```

```

7  ...
8  # Handle layar awal, lobby, gameplay, end screen
9  ...
10

```

Kode 20: Loop Utama Game

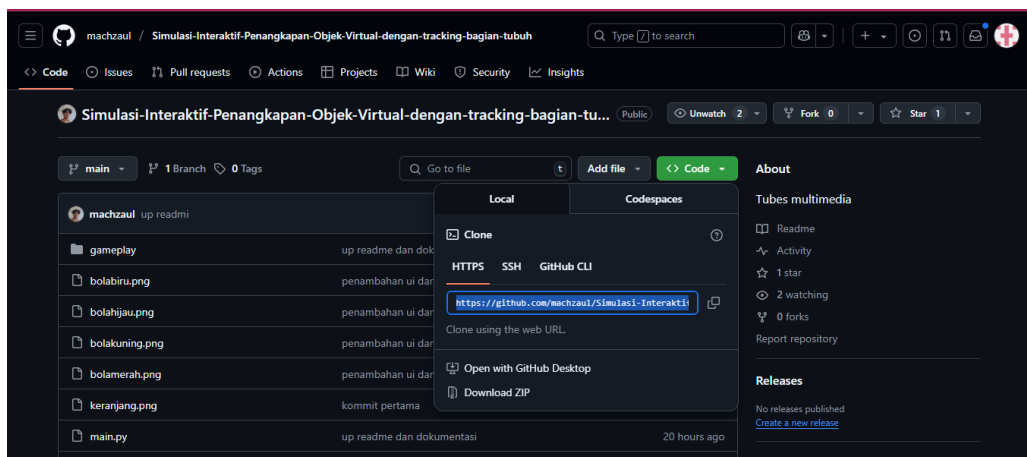
Kode ini adakah inti dari permainan, Loop yang berfungsi sebagai pembaca kamera, mendeteksi wajah, menggambar UI dan mengupdate setatus permainan hingga game berhenti dimainkan.

## 9 Implementasi Program

### 9.1 Langkah 1: Menyalin Link Repository

Langkah pertama dalam mengimplementasikan program ini adalah menyalin URL repository dari GitHub. Berikut adalah langkah-langkahnya:

1. Buka halaman repository GitHub permainan melalui tautan berikut:  
<https://github.com/machzaul/Simulasi-Interaktif-Penangkapan-Objek-Virtual-dengan-tracking-bagian-tubuh>
2. Klik tombol Code berwarna hijau di bagian atas repository.
3. Pilih opsi HTTPS dan salin URL yang ditampilkan, yaitu:



Gambar 1: Salin URL Repository

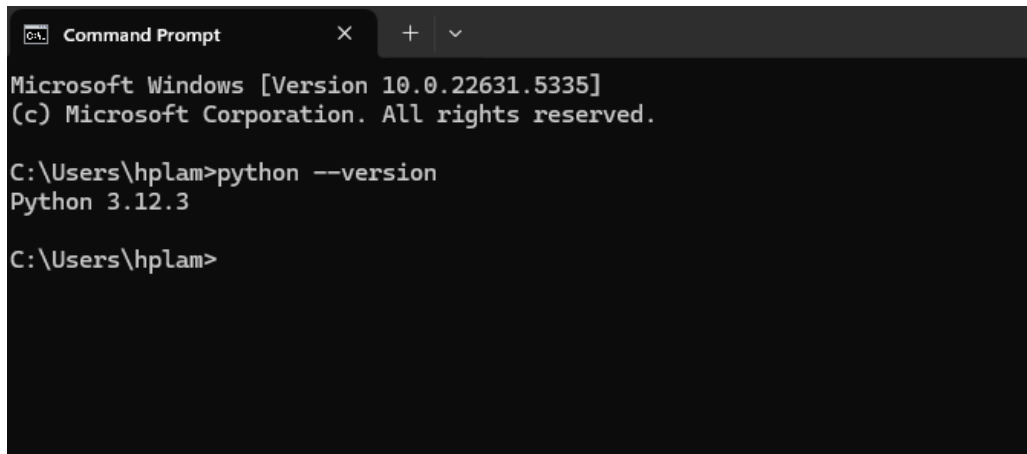
float

### 9.2 Langkah 2: Memeriksa Versi Python

Untuk memastikan program dapat dijalankan dengan benar, diperlukan Python versi 3.10 hingga 3.12. Berikut adalah langkah-langkah untuk mengecek versi Python:

1. Buka terminal atau command prompt
2. Ketik perintah "python-version"
3. Pastikan versi Python yang ditampilkan berada dalam rentang 3.10 hingga 3.12. Jika versi Python tidak sesuai, instal Python versi yang sesuai terlebih dahulu





```

Microsoft Windows [Version 10.0.22631.5335]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hplam>python --version
Python 3.12.3

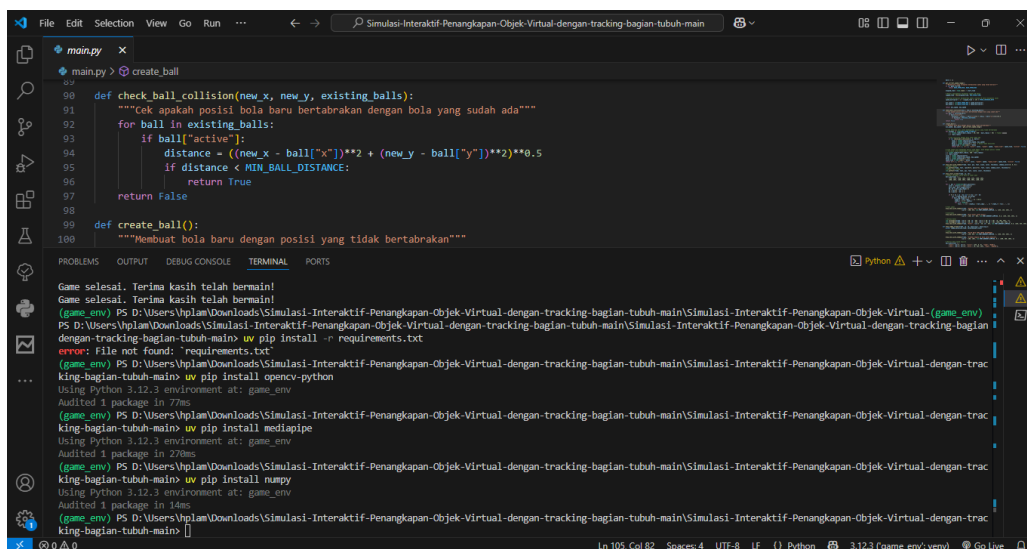
C:\Users\hplam>

```

Gambar 2: Version Python Check

### 9.3 Langkah 3: Menginstall Depenensi

Langkah selanjutnya menginstall depenensi yang disarankan, bisa menggunakan perintah requirements.txt untuk menjalankan program, jika mengalami kesulitan bisa menginstall secara manual seperti gambar dibawah ini:



```

main.py
def check_ball_collision(new_x, new_y, existing_balls):
    """Cek apakah posisi bola baru bertabrakan dengan bola yang sudah ada"""
    for ball in existing_balls:
        if ball["active"]:
            distance = ((new_x - ball["x"])**2 + (new_y - ball["y"])**2)**0.5
            if distance < MIN_BALL_DISTANCE:
                return True
    return False

def create_ball():
    """Membuat bola baru dengan posisi yang tidak bertabrakan"""

```

```

Game selesai. Terima kasih telah bermain!
Game selesai. Terima kasih telah bermain!
(game_env) PS D:\Users\hplam\Downloads\Simulasi-Interaktif-Penangkapan-Objek-Virtual-dengan-tracking-bagian-tubuh-main\Simulasi-Interaktif-Penangkapan-Objek-Virtual-dengan-tracking-bagian-tubuh-main> pip install -r requirements.txt
error: File not found: 'requirements.txt'
(game_env) PS D:\Users\hplam\Downloads\Simulasi-Interaktif-Penangkapan-Objek-Virtual-dengan-tracking-bagian-tubuh-main\Simulasi-Interaktif-Penangkapan-Objek-Virtual-dengan-tracking-bagian-tubuh-main> uv pip install opencv-python
Using Python 3.12.3 environment at: game_env
Audited 1 package in 27ms
(game_env) PS D:\Users\hplam\Downloads\Simulasi-Interaktif-Penangkapan-Objek-Virtual-dengan-tracking-bagian-tubuh-main\Simulasi-Interaktif-Penangkapan-Objek-Virtual-dengan-tracking-bagian-tubuh-main> uv pip install mediapipe
Using Python 3.12.3 environment at: game_env
Audited 1 package in 27ms
(game_env) PS D:\Users\hplam\Downloads\Simulasi-Interaktif-Penangkapan-Objek-Virtual-dengan-tracking-bagian-tubuh-main\Simulasi-Interaktif-Penangkapan-Objek-Virtual-dengan-tracking-bagian-tubuh-main> uv pip install numpy
Using Python 3.12.3 environment at: game_env
Audited 1 package in 14ms
(game_env) PS D:\Users\hplam\Downloads\Simulasi-Interaktif-Penangkapan-Objek-Virtual-dengan-tracking-bagian-tubuh-main\Simulasi-Interaktif-Penangkapan-Objek-Virtual-dengan-tracking-bagian-tubuh-main>

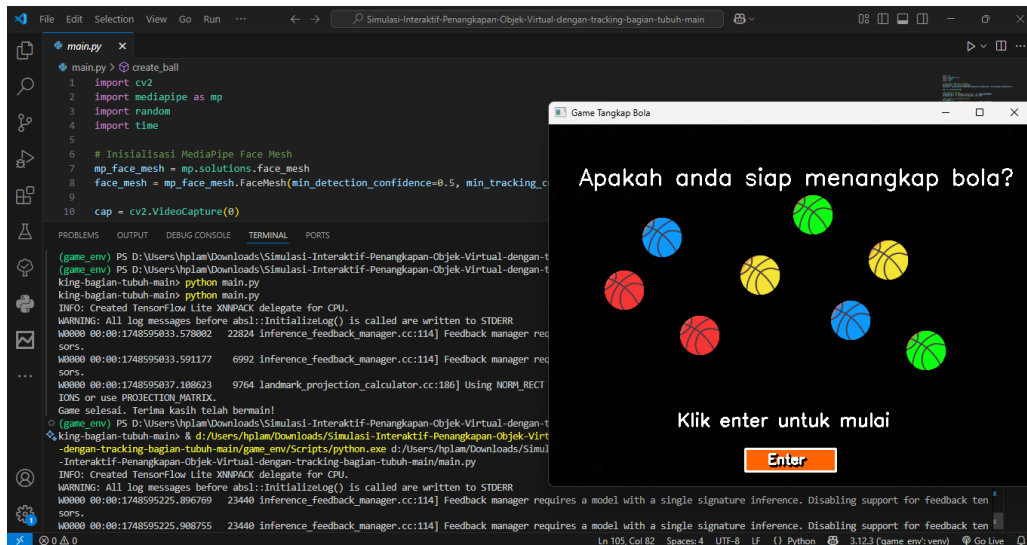
```

Gambar 3: Version Python Check

### 9.4 Langkah 4: Menjalankan Program

Setelah depenensi terinstall, jalankan kode program dengan perintah pada terminal vscode sebagai berikut:

- python main.py atau bisa klik run pada bagian atas sebelah kiri vscode



Gambar 4: Run Program

## 9.5 Langkah 5: Antarmuka Program

Tampilan antarmuka terdiri dari beberapa bagian utama yaitu:

1. Home Screen : Menu utama serta memperkenalkan permainan dan memberikan instruksi awal kepada pemain untuk klik enter untuk memulai permainan.
2. Lobby Selection : Ketika pemain untuk memilih warna bola yang ingin ditangkap pada layar dengan mengarahkan kepala selama 2 detik.
3. Gameplay : Inti dari permainan, di mana pemain harus menangkap bola dengan kontrol dan bergerak sesuai dengan posisi hidung pemain (keranjang) serta harus menangkap bola dengan warna yang benar.
4. Victory Screen : Ketika pemain berhasil menangkap semua bola dengan warna yang benar dan tampilan skor akhir setelah menang.
5. Game Over Screen : Ketika pemain menangkap bola dengan warna yang salah atau waktu habis tanpa menyelesaikan petunjuk dengan benar, serta opsi untuk mencoba lagi.

### 9.5.1 Tampilan Antarmuka

Berikut adalah tampilan tangkapan layar antarmuka permainan program:

1. Home Screen : Menu utama permainan  
Pada Home Screen, pemain diperkenalkan pada tampilan awal pada permainan dan diminta untuk memulai. Tampilan ini menampilkan pesan selamat datang seperti "Apakah kamu siap menangkap bola?" serta tombol "Enter" untuk memulai permainan. Gambar bola warna-warni juga ditampilkan sebagai dekorasi tampilan awal untuk meningkatkan kesan visual. Ketika pemain menekan tombol "Enter", layar beralih ke Lobby Selection.



Gambar 5: Home Screen

## 2. Lobby Selection : Memilih warna bola yang ingin ditangkap

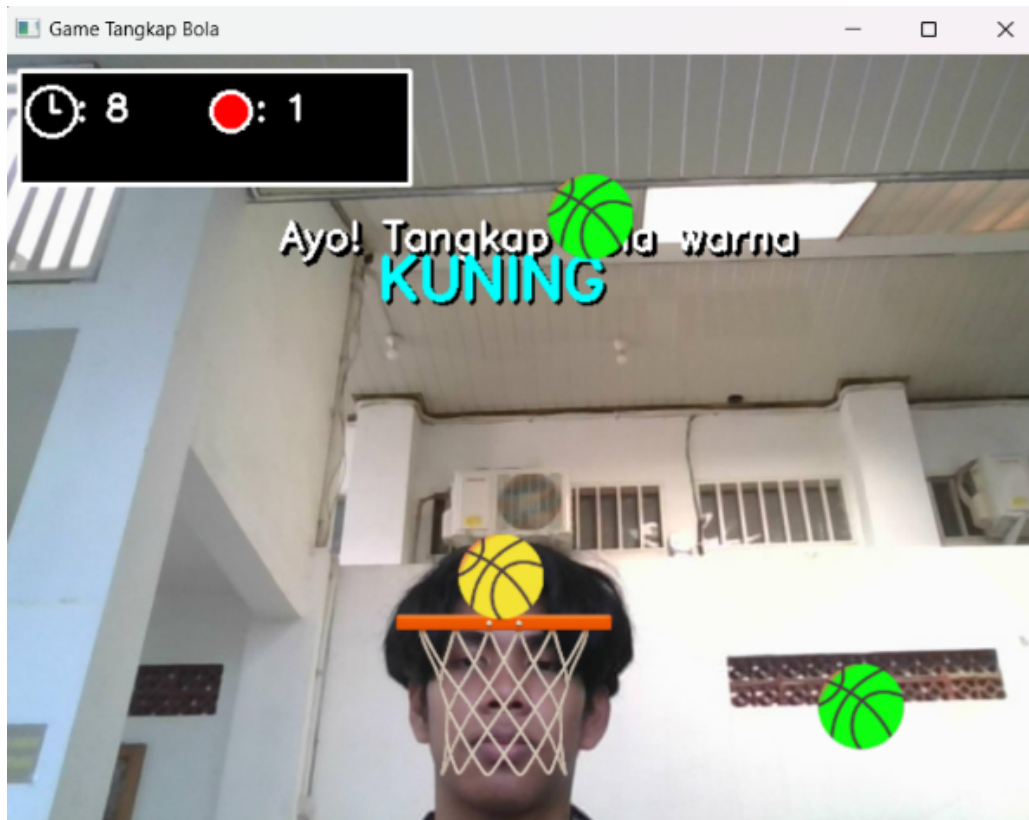
Setelah Home Screen, pemain masuk ke layar pemilihan bola. Di sini, pemain harus memilih warna bola yang ingin ditangkap selama gameplay. Pemain dapat melakukan pemilihan dengan mengarahkan kepala ke salah satu dari empat bola yang tersedia di layar. Sistem menggunakan teknologi deteksi wajah (MediaPipe Face Mesh) untuk mendeteksi posisi hidung pemain. Jika posisi hidung berada dalam radius dari salah satu bola, bola tersebut dipilih. Setelah pemain mengarahkan kepala ke bola yang dipilih selama 2 detik, warna bola tersebut akan terpilih sebagai target permainan, dan layar beralih ke Gameplay .



Gambar 6: Lobby Selection

### 3. Game Play : Inti Permainan

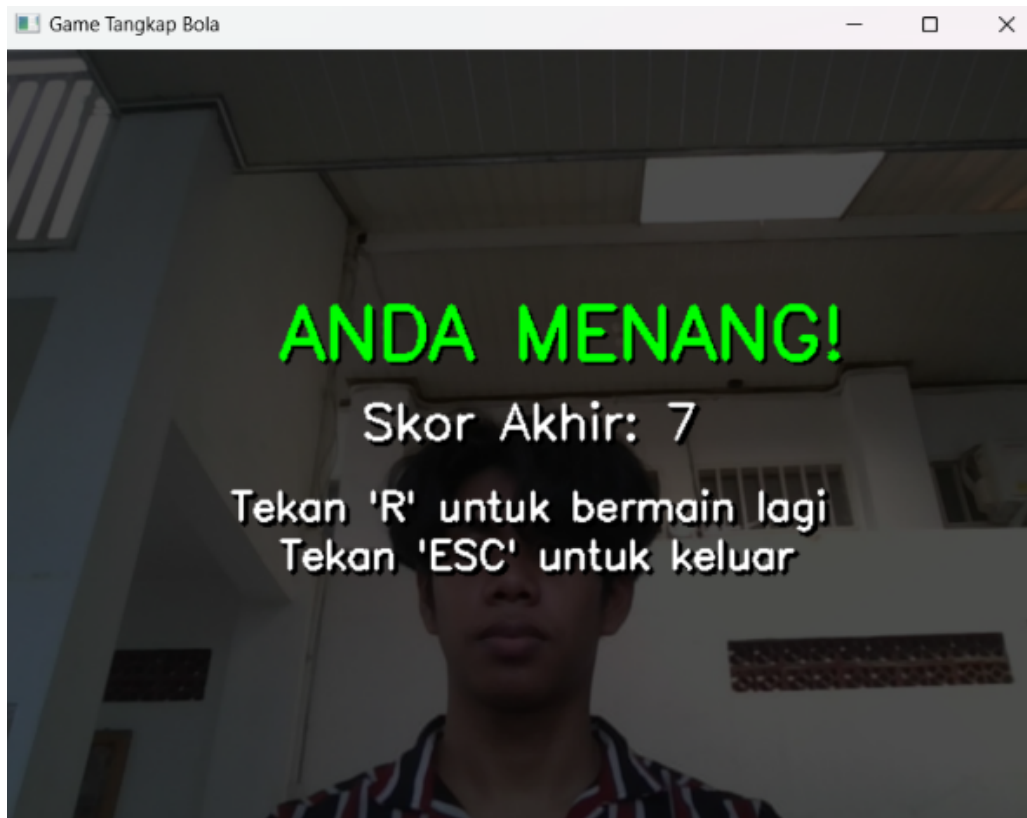
Ini adalah tahap utama permainan di mana pemain harus menangkap bola dengan warna yang telah dipilih sebelumnya. Bola jatuh dari atas layar dengan kecepatan yang meningkat seiring waktu. Pemain menggunakan gerakan kepala untuk menggerakkan keranjang agar sesuai dengan posisi bola. Jika pemain menangkap bola dengan warna yang benar, skor bertambah. Namun, jika pemain menangkap bola dengan warna yang salah atau waktu habis tanpa menyelesaikan misi, permainan berakhir. UI gameplay menampilkan elemen seperti waktu tersisa, skor pemain, dan instruksi untuk menangkap bola dengan warna tertentu.



Gambar 7: Game Play

#### 4. Victory Screen : Permainan Menang

Jika pemain berhasil menangkap semua bola dengan warna yang benar sebelum waktu habis, tampilan victory akan muncul untuk memberikan umpan balik permainan. Teks "ANDA MENANG!" ditampilkan bersama dengan skor akhir. Pemain memiliki opsi untuk bermain ulang dengan menekan tombol "R" atau keluar dengan menekan "ESC". Layar ini memberikan pengalaman puas kepada pemain setelah menyelesaikan permainan dengan sukses.

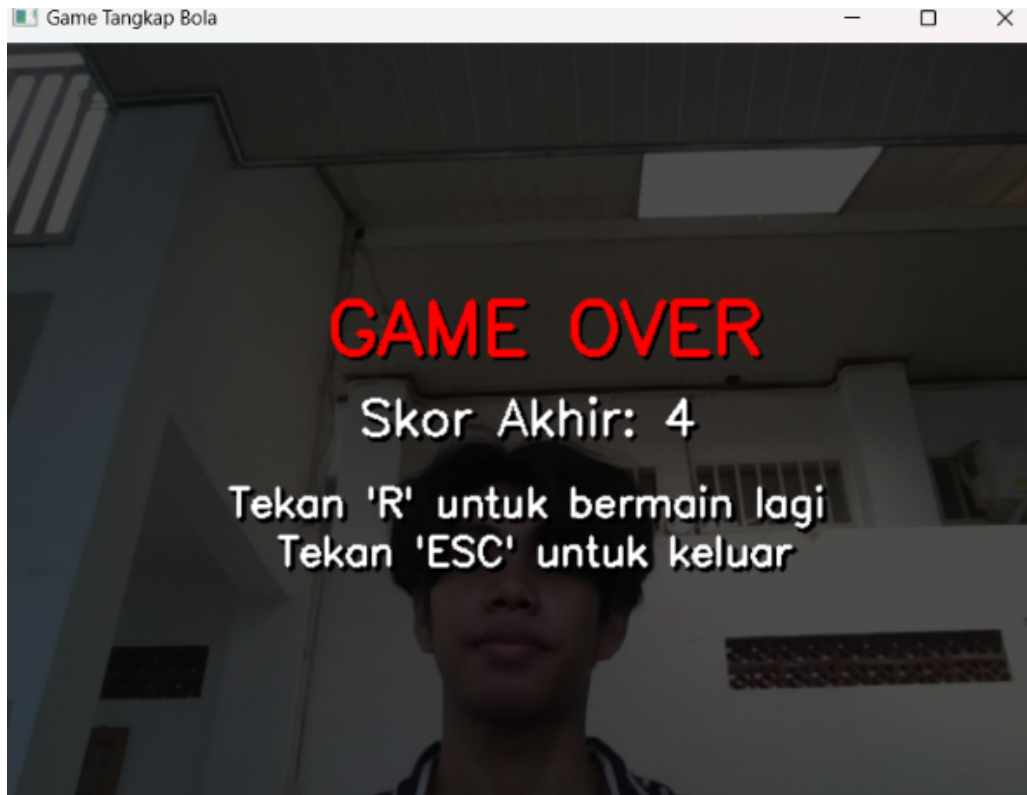


Gambar 8: Victory Screen

##### 5. Game Over : Permainan Kalah

Jika pemain gagal menyelesaikan permainan karena menangkap bola dengan warna yang salah atau waktu habis tanpa menyelesaikan misi, layar ini muncul untuk memberikan umpan balik negatif. Teks "GAME OVER" ditampilkan bersama dengan skor akhir. Pemain juga memiliki opsi untuk mencoba lagi dengan menekan tombol "R" atau keluar dengan menekan "ESC". Layar ini memberikan kesempatan kepada pemain untuk belajar dari kesalahan dan mencoba kembali.





Gambar 9: Game Over Screen

## 10 Instalasi dan Setup

- Clone Repository

```
1 <div style="background-color: #f6f8fa; padding: 16px; border-radius: 6px; border-left: 4px  
2 solid #0366d6;">  
3 <strong> Clone dari GitHub:</strong>  
4 </div>v
```

Kode 21: Clone Repository

```
1 # Clone repository  
2 git clone https://github.com/machzaul/Simulasi-Interaktif-Penangkapan-Objek-Virtual-dengan-  
3 tracking-bagian-tubuh.git  
4 # Masuk ke direktori proyek  
5 cd Simulasi-Interaktif-Penangkapan-Objek-Virtual-dengan-tracking-bagian-tubuh
```

Kode 22: Clone Repository

- Persiapan Environment

```
1 <div style="background-color: #fff3cd; padding: 16px; border-radius: 6px; border-left: 4px  
2 solid #ffc107;">  
3 <strong> Penting:</strong> Disarankan menggunakan virtual environment untuk menghindari konflik  
4 dependencies.  
5 </div>
```

Kode 23: Persiapan Environment

```
1 # Buat virtual environment (opsional tapi disarankan)
2 python -m venv game_env
3 # Aktifkan virtual environment
4 # Windows:
5 game_env\Scripts\activate
6 # macOS/Linux:
7 source game_env/bin/activate
8
```

Kode 24: Persiapan Environment

- Install Dependencies

```
1 <div style="background-color: #d1ecf1; padding: 16px; border-radius: 6px; border-left: 4px
  solid #bee5eb;">
2 <strong> Install semua dependencies yang diperlukan:</strong>
3 </div>
4
```

Kode 25: Install Dependencies

```
1 # Install dependencies utama
2 pip install opencv-python pip install mediapipe
3 pip install numpy
4 # Atau install semua sekaligus (jika ada requirements.txt)
5 pip install -r requirements.txt
6
```

Kode 26: Install Dependencies

Manual Installation (jika ada masalah):

```
1 # OpenCV untuk computer vision
2 pip install opencv-python==4.8.1.78
3 # MediaPipe untuk face detection
4 pip install mediapipe==0.10.7
5 # NumPy untuk array processing
6 pip install numpy==1.24.3
7
```

Kode 27: Manual Installation



## 11 Lampiran

- Kode :[Link Github](#)