

PEMROGRAMAN BERORIENTASI OBJEK

LAPORAN GAME ESCAPE THE ENEMY



Dosen Pengampu :

I Gde Agung Sri Sidhimantra, S.Kom., M.Kom.

Dimas Novian Aditia Syahputra, S.Tr.T., M.Tr.T.

Moch Deny Pratama, S.Tr.T., M.Kom.

Binti Kholifah, S.Kom., M.Tr.Kom.

Disusun Oleh :

Dhito Nugroho 23091397004

Kharis Hendroh Prasetyo O 23091397019

Muhamad Rifky Ramdhani 23091397024

Program Studi D4 Manajemen Informatika

Fakultas Vokasi

Universitas Negeri Surabaya

2024

I. PENDAHULUAN

"Escape the Enemy" adalah sebuah permainan yang bertemakan pixel berbasis Python yang dikembangkan menggunakan library Pygame, sebuah pustaka populer untuk pembuatan game 2D. Game ini dirancang sebagai bagian dari proyek akhir mata kuliah Pemrograman Berorientasi Objek (Object-Oriented Programming, OOP) di tingkat perguruan tinggi. Dengan menggunakan konsep-konsep inti OOP, game ini tidak hanya bertujuan untuk menghibur pemain, tetapi juga menjadi media pembelajaran yang efektif dalam mengimplementasikan teori pemrograman secara nyata.

Dalam game ini, pemain mengendalikan karakter utama yang harus menghindari berbagai tipe musuh dengan pola gerakan unik, mengumpulkan power-up yang muncul secara acak, dan menyelesaikan setiap level dengan mencapai objektif tertentu. Game ini terdiri dari lima level progresif dengan tingkat kesulitan yang terus meningkat. Pada level terakhir, pemain harus mengambil koin emas sebagai tujuan akhir untuk memenangkan permainan. Jika berhasil, layar kemenangan akan menampilkan pesan penghargaan, tetapi jika gagal, layar game over akan muncul.

Proses pengembangan game ini sangat menonjolkan penerapan prinsip-prinsip inti OOP, seperti encapsulation, inheritance, polymorphism, dan abstraction. Dengan encapsulation, setiap objek dalam game (seperti pemain, musuh, dan power-up) diimplementasikan dalam kelas terpisah, sehingga kode menjadi modular dan mudah dikelola. Melalui inheritance, kelas-kelas tertentu mewarisi atribut dan metode dari kelas induk untuk mengurangi duplikasi kode. Konsep polymorphism memungkinkan setiap objek memiliki metode yang sama namun dengan implementasi yang berbeda, seperti perilaku unik musuh saat bergerak. Sementara itu, abstraction menyembunyikan detail implementasi, sehingga pengguna hanya perlu memanggil fungsi atau metode tertentu tanpa harus memahami cara kerjanya di balik layar.

Selain aspek teknis, "Escape the Enemy" juga dirancang dengan mempertimbangkan elemen hiburan dan tantangan. Grafis sederhana namun estetis dipadukan dengan alur permainan yang semakin sulit seiring bertambahnya level. Pola gerakan musuh yang dinamis dan munculnya power-up secara acak menambah tingkat kompleksitas permainan, membuat pemain harus berpikir cepat dan strategis untuk bertahan hidup.

Secara keseluruhan, "Escape the Enemy" merupakan contoh aplikasi praktis dari teori OOP dalam konteks pengembangan game. Dengan menyeimbangkan aspek teknis dan hiburan, game ini diharapkan tidak hanya memberikan pengalaman bermain yang menyenangkan bagi pengguna, tetapi juga menjadi referensi yang berharga bagi mahasiswa atau pengembang pemula yang ingin memahami penerapan konsep OOP dalam pengembangan perangkat lunak berbasis Python.

II. DASAR TEORI

Object-Oriented Programming (OOP) adalah salah satu paradigma pemrograman yang paling banyak digunakan dalam pengembangan perangkat lunak modern. Paradigma ini berfokus pada penciptaan objek sebagai entitas yang mewakili data dan perilaku dalam suatu sistem. Dalam OOP, program dirancang dengan memecah sistem menjadi bagian-bagian kecil yang dikenal sebagai objek. Setiap objek memiliki atribut (data) dan metode (fungsi) yang merepresentasikan karakteristik dan perilaku yang unik. Pendekatan ini membantu pengembang menciptakan aplikasi yang lebih modular, terorganisir, dan mudah dikelola, terutama pada proyek berskala besar.

Berikut adalah penjelasan mendalam tentang konsep utama OOP yang diterapkan dalam pengembangan "Escape the Enemy":

- Encapsulation

Encapsulation adalah proses menyembunyikan detail implementasi dari suatu objek dan hanya menampilkan antarmuka yang diperlukan untuk

berinteraksi dengan objek tersebut. Dalam game "Escape the Enemy", setiap entitas seperti pemain, musuh, power-up, dan koin diimplementasikan sebagai kelas terpisah.

Contoh dalam Game:

Dalam game ini, kelas Player memiliki atribut seperti health, speed, dan shield_active. Semua atribut ini hanya dapat dimodifikasi melalui metode tertentu seperti update() atau saat pemain berinteraksi dengan power-up. Dengan cara ini, atribut-atribut tersebut dilindungi dari manipulasi langsung.

● **Inheritance**

Inheritance memungkinkan sebuah kelas untuk mewarisi atribut dan metode dari kelas lain yang lebih umum. Dalam "Escape the Enemy", kelas GameObject bertindak sebagai kelas dasar (parent class) yang menjadi fondasi untuk kelas-kelas lain seperti Player, Enemy, dan PowerUp.

Contoh dalam Game:

Kelas Enemy dan PowerUp mewarisi kelas dasar seperti GameObject, yang mencakup properti umum seperti posisi (x dan y) serta metode seperti draw() dan update().

● **Polymorphism**

Polymorphism memungkinkan penggunaan metode dengan nama yang sama pada objek dari kelas yang berbeda, tetapi dengan implementasi yang unik sesuai dengan kebutuhan objek tersebut. Dalam game ini, konsep polymorphism diterapkan pada metode seperti update() dan draw().

Contoh dalam Game:

Kelas Enemy memiliki beberapa tipe musuh seperti rock, fire, dan bomb, di mana setiap jenis musuh memiliki properti dan pola pergerakan yang berbeda. Meskipun begitu, semua musuh dipanggil dengan metode update() dan draw() yang sama.

- Abstraction

Abstraction adalah proses menyembunyikan detail implementasi dari pengguna dan hanya menampilkan antarmuka yang diperlukan. Dalam game ini, abstraction diterapkan untuk menyederhanakan interaksi antara berbagai komponen sistem.

Contoh dalam Game:

Kelas PowerUp memiliki tipe seperti health, shield, dan slow, tetapi logika penerapan efek masing-masing power-up disembunyikan di dalam kelas. Pemain hanya perlu mengetahui bahwa mengambil power-up akan memberikan efek tertentu tanpa mengetahui bagaimana efek tersebut diterapkan.

III. KONSEP GAME

“Escape the Enemy” adalah sebuah game bertema survival yang menantang pemain untuk bertahan hidup di tengah ancaman musuh yang terus berdatangan. Dalam permainan ini, pemain harus memanfaatkan keterampilan, strategi, dan berbagai power-up yang tersedia untuk memperpanjang waktu bertahan hidup dan menghadapi tantangan yang semakin sulit di setiap level.

Game ini terdiri dari 9 level, di mana setiap level menghadirkan musuh yang lebih kuat dan rintangan yang lebih kompleks. Tujuan utama dari permainan ini adalah untuk mencapai level terakhir (level 9) dan mengumpulkan koin yang muncul sebagai hadiah atas ketahanan dan keberanian pemain. Dengan kombinasi gameplay yang intens, elemen strategi, dan hadiah menarik, “Escape the Enemy” menawarkan pengalaman bermain yang seru dan penuh adrenalin.

Berikut adalah fitur-fitur utama dalam game ini:

- Karakter Pemain

Sebelum memulai permainan, pemain dapat memilih karakter yang sesuai dengan preferensi mereka. Setiap karakter memiliki desain unik yang menambah daya tarik visual dalam permainan.

- Musuh

Game ini menghadirkan berbagai tipe musuh dengan pola gerakan berbeda, seperti zigzag, linear, dan random. Setiap tipe musuh dirancang untuk memberikan tantangan unik, sehingga pemain harus terus beradaptasi dengan situasi yang berubah-ubah.

- Power-Up

Di dalam game ini terdapat item seperti health, shield, dan slow (debuff) yang dapat membantu pemain bertahan lebih lama.

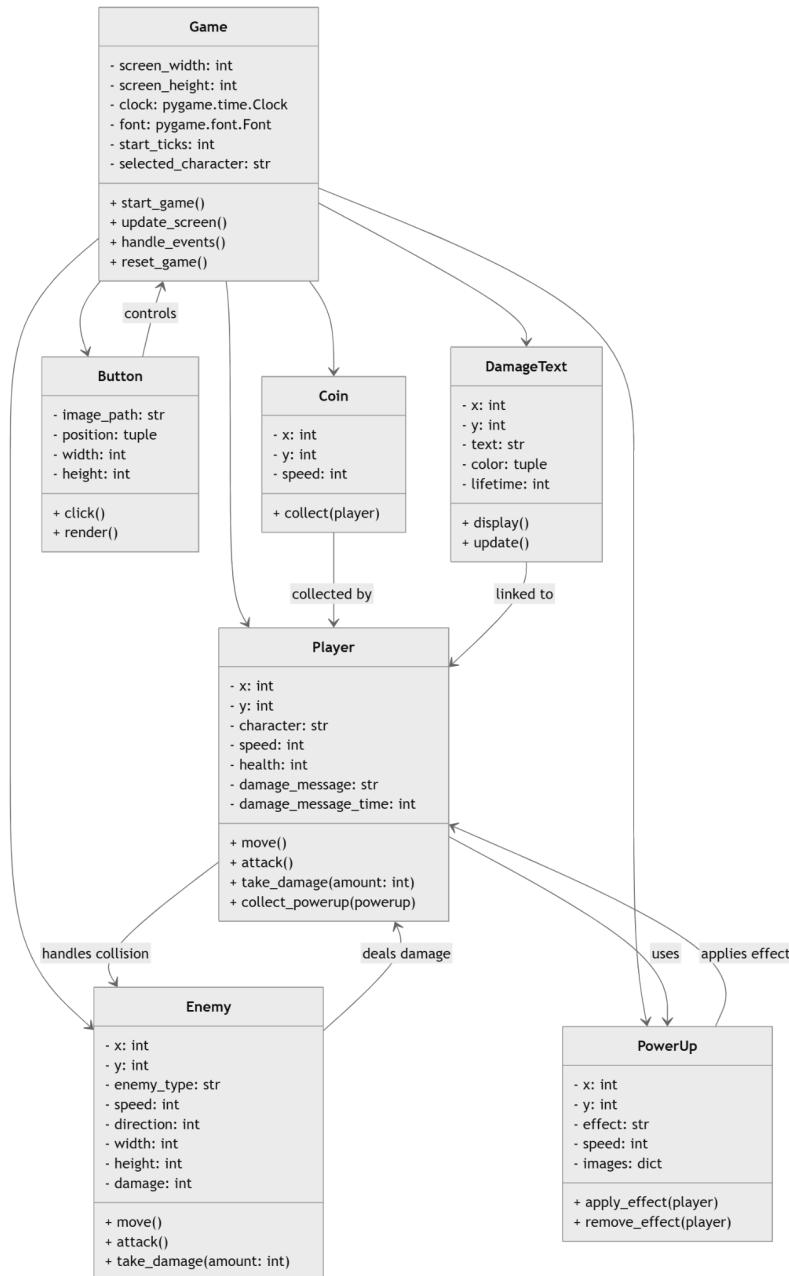
- Level

Permainan terdiri dari sembilan level, dengan tingkat kesulitan yang terus meningkat di setiap level. Setiap level memiliki desain arena, jumlah musuh, dan pola serangan yang lebih kompleks, sehingga menuntut pemain untuk terus mengasah kemampuan bertahan mereka.

- Objektif Akhir

Pada level terakhir (level 9), tujuan utama adalah mengumpulkan koin yang muncul di arena. Pemain yang berhasil mengambil koin akan memenangkan permainan, dan layar kemenangan akan ditampilkan sebagai penghargaan atas keberhasilan mereka.

IV. CLASS DIAGRAM



V. ALUR GAME

1. Menu Utama

Pemain memulai dari menu utama, di mana mereka dapat memilih salah satu opsi berikut:

- Start: Memulai permainan dari awal.
- Hero: Memilih karakter untuk permainan
- Quit: Menutup permainan.

2. Pemilihan Karakter

Sebelum masuk ke dalam gameplay, pemain diberikan kesempatan untuk memilih karakter yang ingin dimainkan. Setiap karakter memiliki tampilan unik yang memberikan variasi dalam pengalaman bermain.

3. Gameplay

- Pemain menghadapi tantangan dengan menghindari musuh, mengumpulkan power-up, dan bertahan hidup melalui lima level.
- Durasi setiap level berlangsung selama 10 detik sebelum level berikutnya dimulai.
- Musuh dan power-up muncul secara acak dengan tingkat kesulitan yang terus meningkat, memberikan tekanan kepada pemain untuk berpikir cepat dan bereaksi tepat waktu.

4. Objektif Akhir

Pada level 9, sekitar detik ke-80 hingga ke-100, koin akan muncul sebagai tujuan utama. Pemain harus mengambil koin untuk memenangkan permainan.

5. Kemenangan

Jika pemain berhasil mengambil koin, layar kemenangan akan muncul, menampilkan pesan selamat dan gambar mahkota sebagai simbol keberhasilan pemain dalam menyelesaikan tantangan.

6. Kekalahan

Jika health(darah) pemain habis akibat serangan musuh sebelum berhasil mengambil koin, layar game over akan ditampilkan. Pemain dapat memilih untuk kembali ke menu utama atau keluar dari permainan.

VI. TAMPILAN GAME

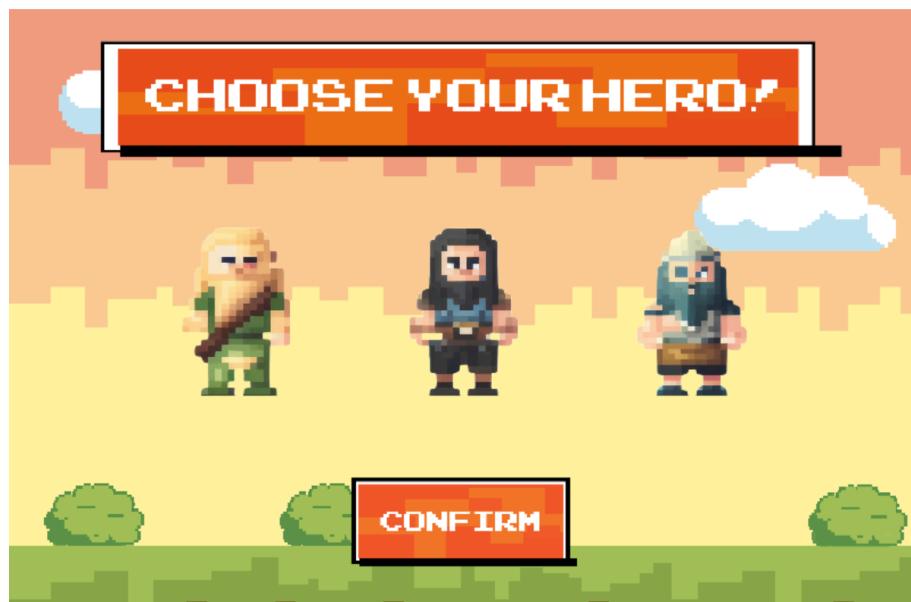
a. Tampilan Awal Game

Ketika player berada di main menu, maka tampilannya akan seperti ini.



b. Tampilan Hero

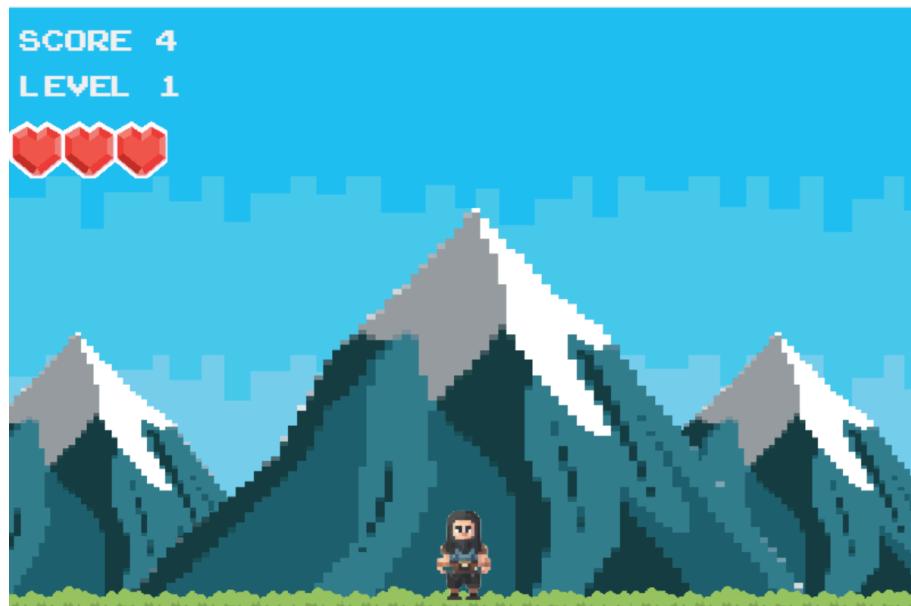
Ketika player menekan tombol "Hero" pada main menu, maka akan diarahkan ke bagian pemilihan karakter. Pemain dapat memilih 1 dari 3 karakter yang tersedia. Dan dapat menekan tombol confirm jika sudah memilih karakter.



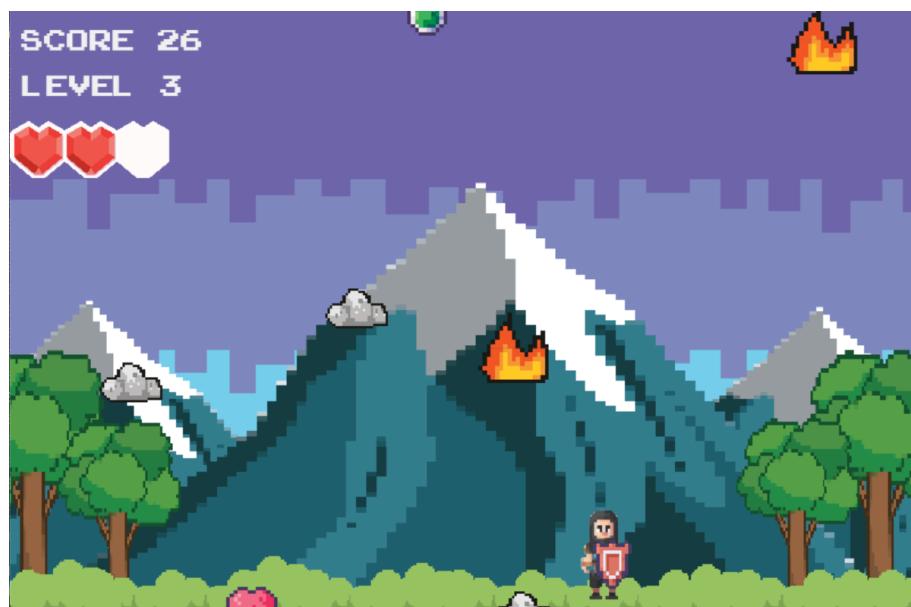
c. Tampilan Ketika Game Dimainkan

Ketika player sudah menentukan karakter mana yang digunakan (dalam contoh memilih karakter 2) dan menekan tombol start maka game akan segera dimulai.

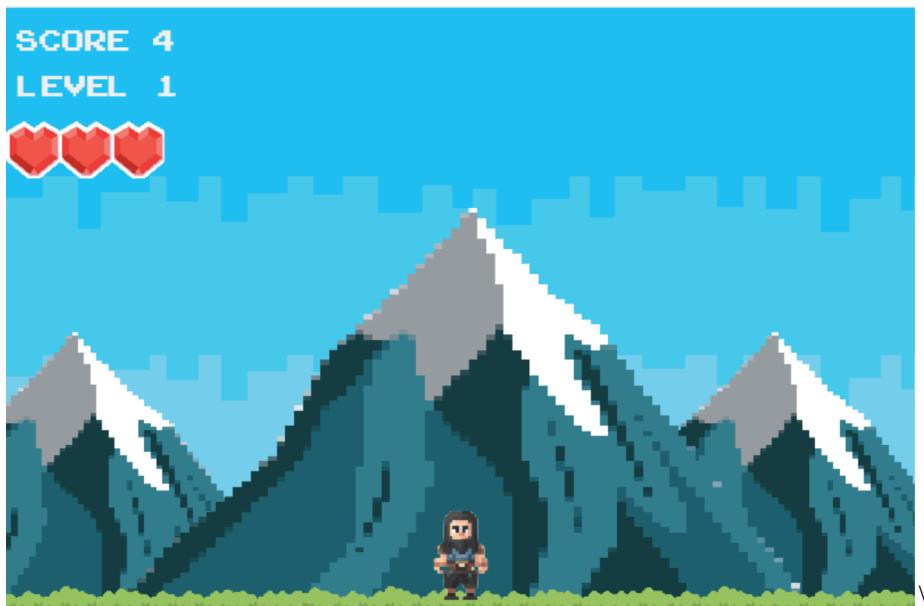
- Ketika awal permainan



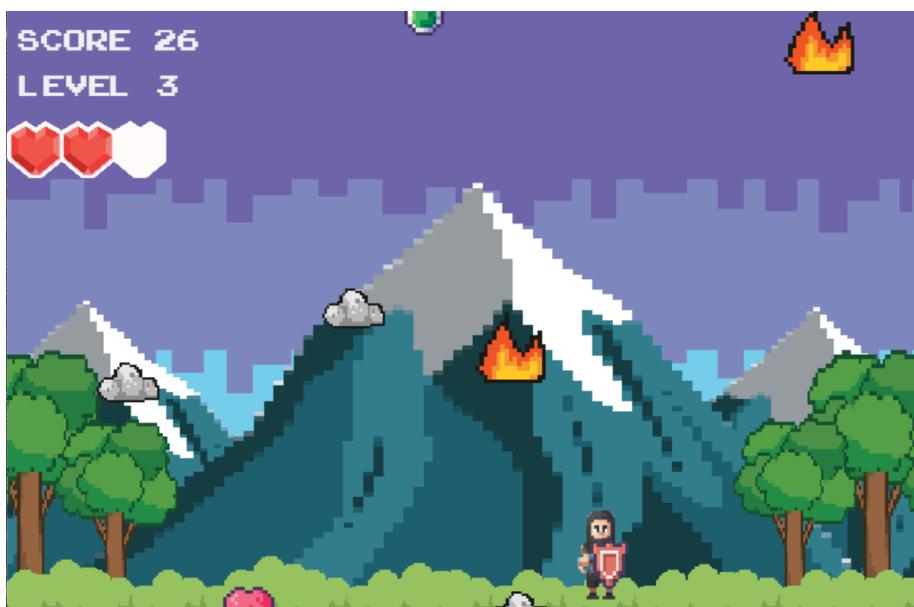
- Ketika player mendapatkan powerup (Shield)



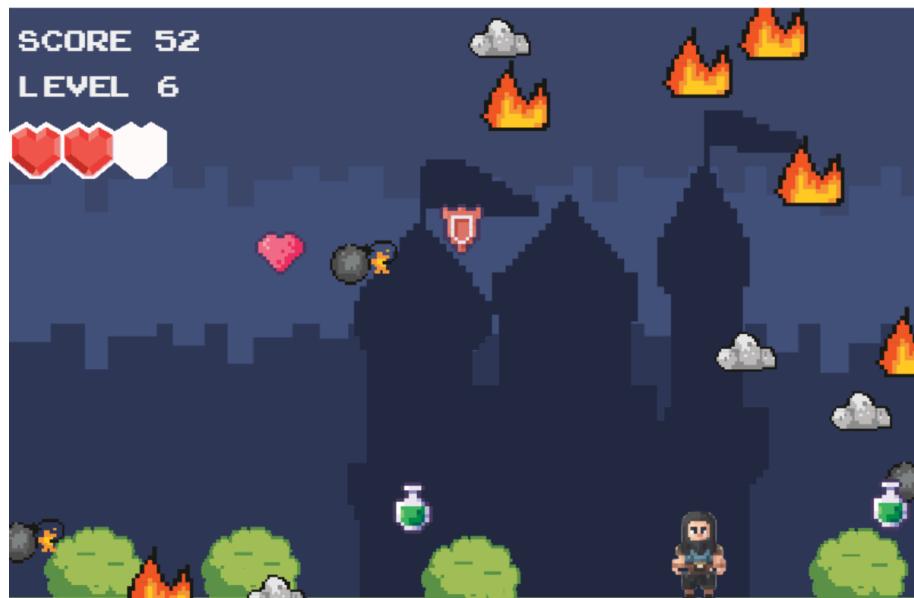
- Pergantian Latar
 - Latar 1 (Level 1-3)



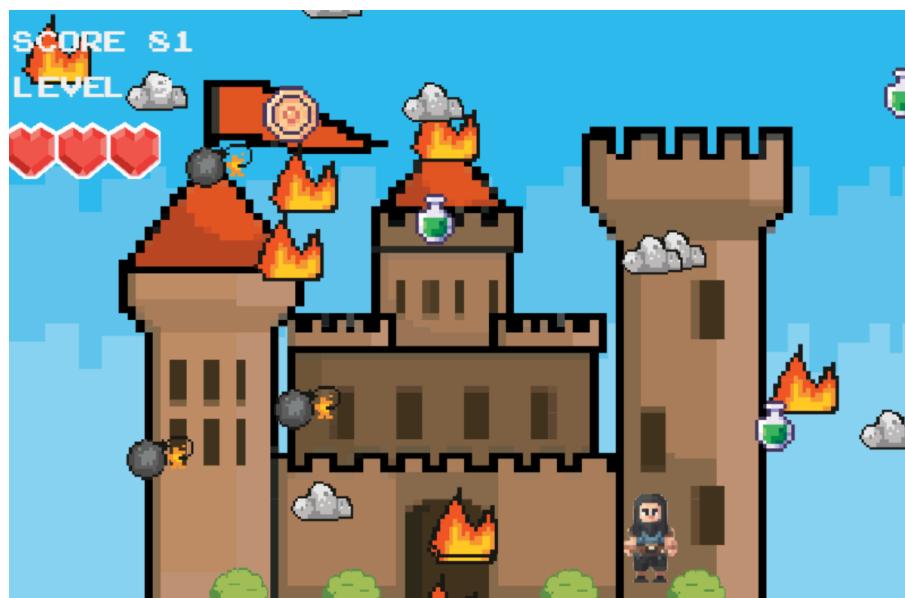
- Latar 2 (Level 3-6)



o Latar 3 (Level 6-8)



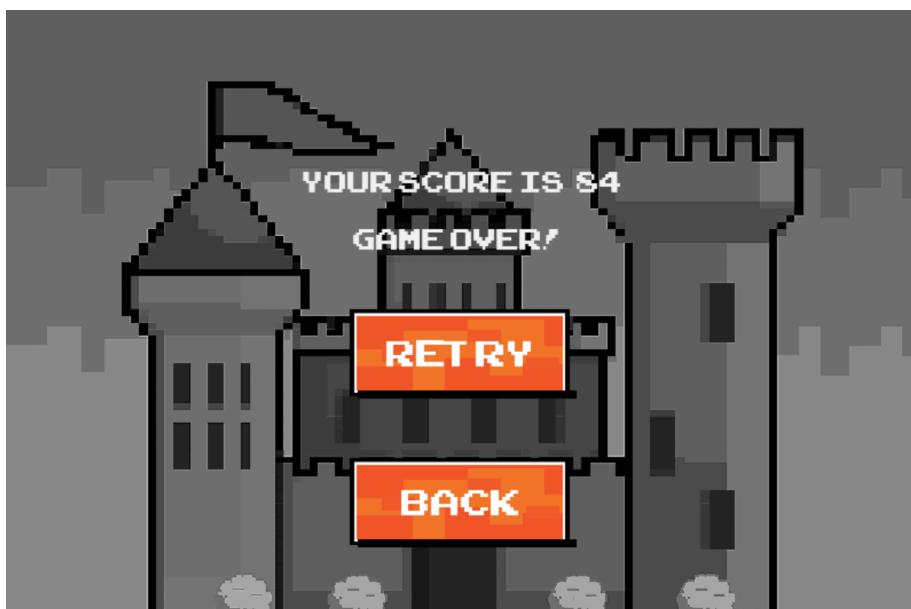
o Latar 4 (Level 9)



- Ketika Player Berhasil Memenangkan Game



- Ketika Player Gagal



VII. SOURCE CODE

Main.py

1. Import Library dan Inisialisasi

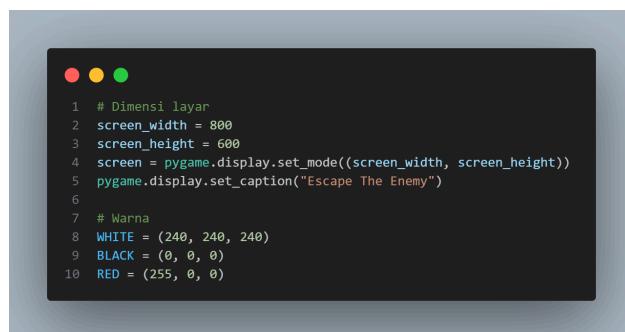


```
1 import pygame
2 import sys
3 from player import Player
4 from enemy import Enemy
5 from powerup import PowerUp
6 import random
7
8 pygame.init()
```

Bagian ini mengimpor semua library dan modul yang diperlukan untuk membuat permainan:

- pygame: Digunakan untuk membuat game 2D. Library ini menyediakan fungsi untuk grafis, suara, input, dan lainnya.
- sys: Mengontrol fungsi sistem, seperti keluar dari program.
- Kelas Kustom:
 - Player: Mengatur logika pemain.
 - Enemy: Mengatur logika musuh.
 - PowerUp: Mengatur logika item power-up.
- random: Digunakan untuk membuat nilai acak, seperti menentukan kapan power-up muncul.
- pygame.init(): Menginisialisasi semua modul pygame sebelum digunakan.

2. Pengaturan Layar dan Warna

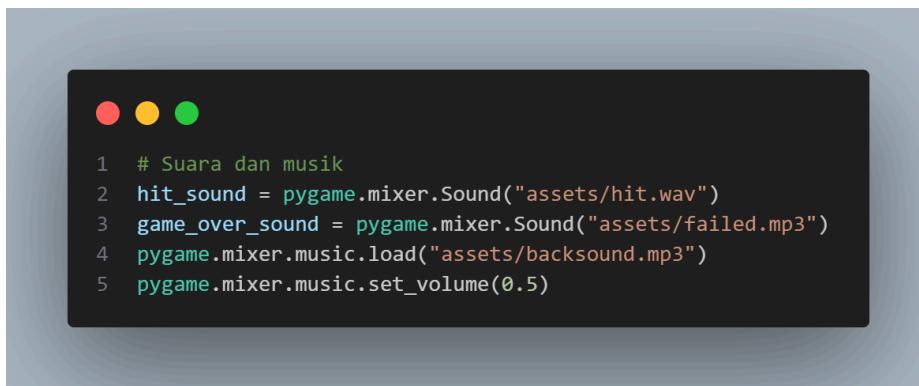


```
1 # Dimensi layar
2 screen_width = 800
3 screen_height = 600
4 screen = pygame.display.set_mode((screen_width, screen_height))
5 pygame.display.set_caption("Escape The Enemy")
6
7 # Warna
8 WHITE = (240, 240, 240)
9 BLACK = (0, 0, 0)
10 RED = (255, 0, 0)
```

Bagian ini mengatur tampilan layar dan warna dasar yang akan digunakan:

- Dimensi Layar: screen_width dan screen_height menentukan ukuran layar game (800x600 piksel).
- Pembuatan Layar: pygame.display.set_mode() menciptakan jendela dengan dimensi yang telah ditentukan.
- Judul Layar: pygame.display.set_caption() menetapkan nama jendela menjadi "Escape The Enemy".
- Warna:
 - WHITE: Warna putih dengan nilai RGB (240, 240, 240).
 - BLACK: Warna hitam dengan nilai RGB (0, 0, 0).
 - RED: Warna merah dengan nilai RGB (255, 0, 0).

3. Suara dan Musik



```
1 # Suara dan musik
2 hit_sound = pygame.mixer.Sound("assets/hit.wav")
3 game_over_sound = pygame.mixer.Sound("assets/failed.mp3")
4 pygame.mixer.music.load("assets/backsound.mp3")
5 pygame.mixer.music.set_volume(0.5)
```

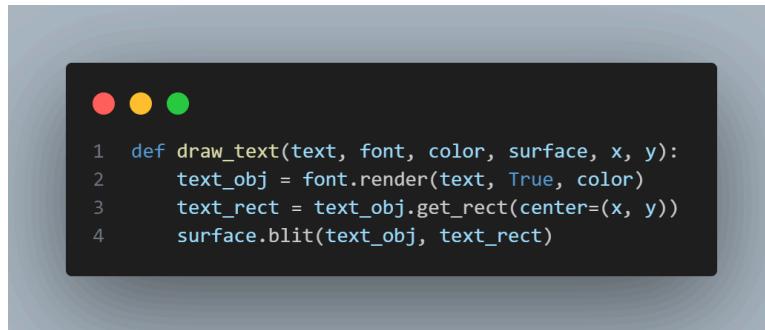
Bagian ini memuat efek suara dan musik untuk permainan:

- Efek Suara:
 - hit_sound: Efek suara yang dimainkan saat pemain bertabrakan dengan musuh.
 - game_over_sound: Efek suara untuk layar akhir permainan.
- Musik Latar:
 - pygame.mixer.music.load() memuat file musik untuk diputar selama permainan berlangsung.

- `pygame.mixer.music.set_volume(0.5)` mengatur volume musik menjadi setengah.

4. Fungsi Utilitas

a. Fungsi draw_text



```

1 def draw_text(text, font, color, surface, x, y):
2     text_obj = font.render(text, True, color)
3     text_rect = text_obj.get_rect(center=(x, y))
4     surface.blit(text_obj, text_rect)

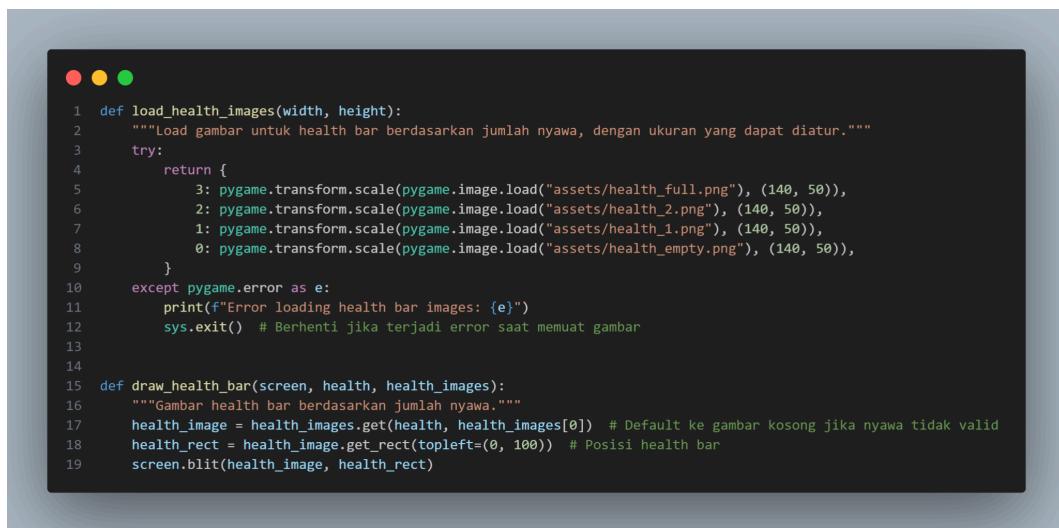
```

Fungsi ini digunakan untuk menampilkan teks di layar.

Parameter:

- text: Isi teks yang akan ditampilkan.
- font: Jenis dan ukuran font yang digunakan.
- color: Warna teks (contoh: putih atau merah).
- surface: Layar tempat teks akan ditampilkan.
- x, y: Koordinat posisi teks.

b. Fungsi health Bar



```

1 def load_health_images(width, height):
2     """Load gambar untuk health bar berdasarkan jumlah nyawa, dengan ukuran yang dapat diatur."""
3     try:
4         return [
5             pygame.transform.scale(pygame.image.load("assets/health_full.png"), (140, 50)),
6             pygame.transform.scale(pygame.image.load("assets/health_2.png"), (140, 50)),
7             pygame.transform.scale(pygame.image.load("assets/health_1.png"), (140, 50)),
8             pygame.transform.scale(pygame.image.load("assets/health_empty.png"), (140, 50)),
9         ]
10    except pygame.error as e:
11        print(f"Error loading health bar images: {e}")
12        sys.exit() # Berhenti jika terjadi error saat memuat gambar
13
14
15 def draw_health_bar(screen, health, health_images):
16     """Gambar health bar berdasarkan jumlah nyawa."""
17     health_image = health_images.get(health, health_images[0]) # Default ke gambar kosong jika nyawa tidak valid
18     health_rect = health_image.get_rect(topleft=(0, 100)) # Posisi health bar
19     screen.blit(health_image, health_rect)

```

Fungsi ini bertanggung jawab untuk menampilkan indikator kesehatan pemain:

- **Memuat Gambar Health:** load_health_images() memuat gambar untuk berbagai level kesehatan (contoh: penuh, setengah, kosong).
- **Menggambar Health Bar:** draw_health_bar() menampilkan indikator kesehatan pemain berdasarkan nilai health.

c. Generator Musuh



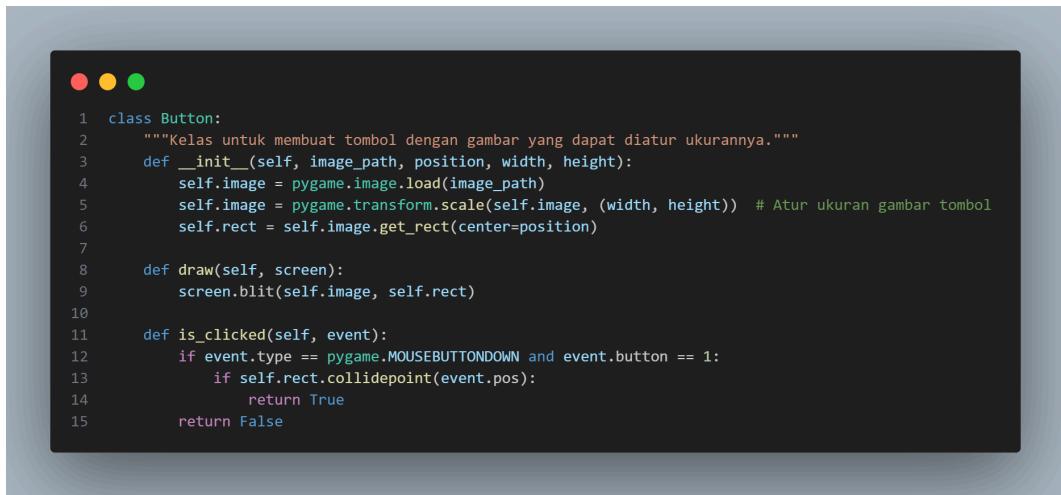
```
1 def get_enemies_for_level(level):
2     """Menghasilkan musuh dengan properti acak berdasarkan level."""
3     enemies = []
4     max_enemies = min(level + 2, 10) # Batasi jumlah musuh berdasarkan level
5     for _ in range(max_enemies):
6         enemy_type = random.choice(["rock", "fire", "bomb"])
7         new_enemy = Enemy(enemy_type)
8         new_enemy.speed = random.randint(2 + level, 5 + level) # Kecepatan musuh bertambah
9         new_enemy.movement_pattern = random.choice(["zigzag", "linear", "random"])
10        enemies.append(new_enemy)
11    return enemies
```

Fungsi ini membuat daftar musuh berdasarkan level permainan:

- Level yang lebih tinggi akan menghasilkan lebih banyak musuh atau musuh yang lebih kuat.
- Ini meningkatkan tantangan secara bertahap.

5. Class Pendukung

a. Kelas button



```
1 class Button:
2     """Kelas untuk membuat tombol dengan gambar yang dapat diatur ukurannya."""
3     def __init__(self, image_path, position, width, height):
4         self.image = pygame.image.load(image_path)
5         self.image = pygame.transform.scale(self.image, (width, height)) # Atur ukuran gambar tombol
6         self.rect = self.image.get_rect(center=position)
7
8     def draw(self, screen):
9         screen.blit(self.image, self.rect)
10
11    def is_clicked(self, event):
12        if event.type == pygame.MOUSEBUTTONDOWN and event.button == 1:
13            if self.rect.collidepoint(event.pos):
14                return True
15        return False
```

- Kelas ini membuat tombol yang dapat diklik, misalnya untuk menu atau pilihan karakter:
 - Mengatur teks, ukuran, posisi, dan warna tombol.
 - Menangani event klik pada tombol.

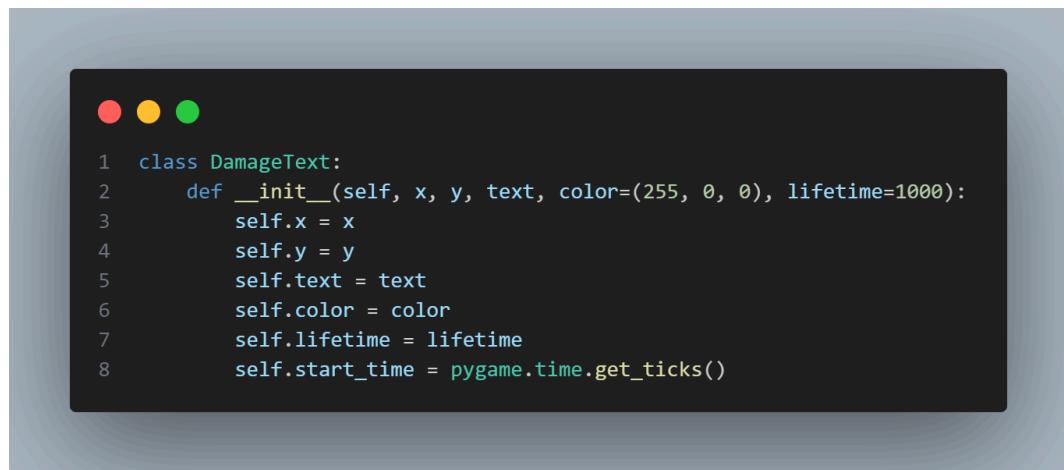
b. Kelas Coin



```
1 class Coin:
2     def __init__(self):
3         self.image = pygame.image.load("assets/coin.png")
4         self.image = pygame.transform.scale(self.image, (50, 50)) # Sesuaikan ukuran
5         self.rect = self.image.get_rect()
6         self.rect.x = random.randint(50, screen_width - 50)
7         self.rect.y = -50 # Mulai di luar layar
8         self.speed = 2
```

- Kelas ini membuat objek koin:
 - Gerakan: Mengatur posisi koin agar bisa bergerak di layar.
 - Penggambaran: Menampilkan koin di layar.

c. Kelas DamageText

A screenshot of a Mac OS X desktop. In the top-left corner, there are three colored window control buttons: red, yellow, and green. Below them is a dark terminal window with white text. The text in the terminal is a Python class definition for 'DamageText'.

```
1 class DamageText:
2     def __init__(self, x, y, text, color=(255, 0, 0), lifetime=1000):
3         self.x = x
4         self.y = y
5         self.text = text
6         self.color = color
7         self.lifetime = lifetime
8         self.start_time = pygame.time.get_ticks()
```

- Kelas ini membuat teks sementara untuk menampilkan angka damage atau efek lainnya di layar:
 - Teks akan muncul sebentar, lalu menghilang.

6. Menu dan Pemilihan Karakter

a. Menu Utama



```
1 def main_menu():
2     """Tampilan menu utama."""
3     global selected_character
4     pygame.init()
5     pygame.mixer.music.play(-1)
6     screen = pygame.display.set_mode((800, 530))
7     pygame.display.set_caption("Escape The Enemy")
8     # Muat gambar background
9     background = pygame.image.load("assets/main_menu.png") # Gambar menu Anda
10    background = pygame.transform.scale(background, (screen_width, screen_height)) # Sesuaikan ukuran dengan layar
11
12    # Tombol dengan gambar dan ukuran yang dapat diatur
13    start_button = Button("assets/start_button.png", (400, 260), 200, 80)
14    hero_button = Button("assets/hero_button.png", (400, 360), 200, 80) # Tombol Hero
15    quit_button = Button("assets/quit_button.png", (400, 460), 200, 80)
16
17    running = True
18    while running:
19        # Tampilkan gambar background
20        screen.blit(background, (0, 0))
21
22        # Gambar teks dan tombol
23        screen.blit(background, (0, 0))
24        start_button.draw(screen)
25        hero_button.draw(screen)
26        quit_button.draw(screen)
27
28
29        pygame.display.flip()
30
31        for event in pygame.event.get():
32            if event.type == pygame.QUIT:
33                pygame.quit()
34                sys.exit()
35            if event.type == pygame.MOUSEBUTTONDOWN and event.button == 1:
36                if start_button.is_clicked(event):
37                    if not selected_character:
38                        draw_text("Choose your hero first!", font, RED, screen, 400, 500)
39                        pygame.display.flip()
40                        pygame.time.delay(2000) # Tampilkan peringatan selama 2 detik
41                    else:
42                        main() # Mulai game
43                        running = False
44
45                if hero_button.is_clicked(event):
46                    character_selection() # Ke layar pemilihan karakter
47
48            if quit_button.is_clicked(event):
49                pygame.quit()
50                sys.exit()
```

- Fungsi ini menampilkan menu awal permainan:
 - Pilihan seperti "Mulai", "Pilih Karakter", atau "Keluar".
 - Menggunakan tombol untuk navigasi.

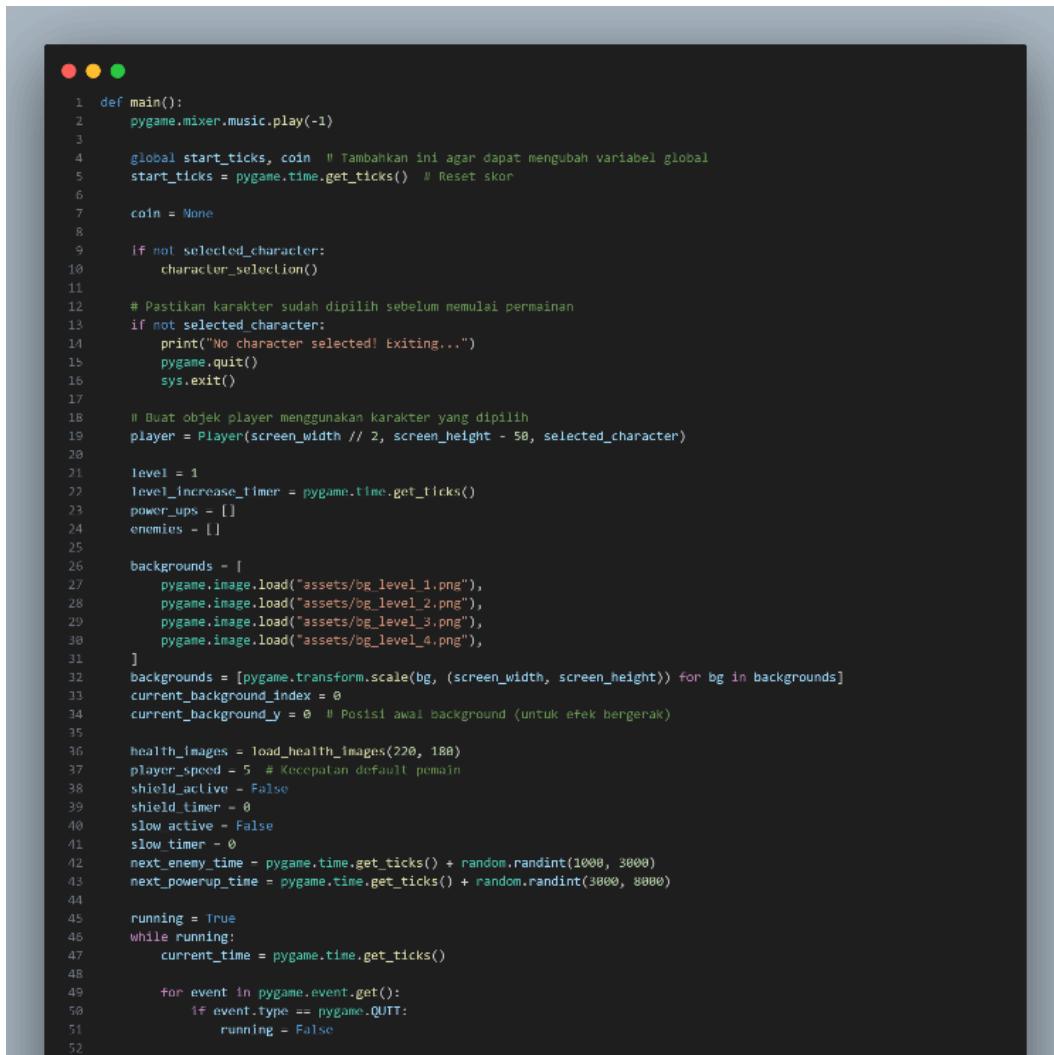
b. Pemilihan Karakter

```
1 def character_selection():
2     global selected_character # Variabel global untuk menyimpan pilihan karakter
3     pygame.init()
4     screen = pygame.display.set_mode((800, 530))
5     # Muat gambar latar belakang
6     background = pygame.image.load("assets/bg_hero.png")
7     background = pygame.transform.scale(background, (800, 530))
8
9     # Gambar karakter
10    chara_1_image = pygame.image.load("assets/chara_1.png")
11    chara_2_image = pygame.image.load("assets/chara_2.png")
12    chara_3_image = pygame.image.load("assets/chara_3.png")
13
14    # Ubah ukuran gambar karakter
15    chara_1_image = pygame.transform.scale(chara_1_image, (100, 150))
16    chara_2_image = pygame.transform.scale(chara_2_image, (100, 150))
17    chara_3_image = pygame.transform.scale(chara_3_image, (100, 150))
18
19    chara_1_rect = chara_1_image.get_rect(center=(200, 265))
20    chara_2_rect = chara_2_image.get_rect(center=(400, 265))
21    chara_3_rect = chara_3_image.get_rect(center=(600, 265))
22
23    confirm_button = Button("assets/confirm_button.png", (400, 450), 200, 80)
24    temp_selected_character = None # Pilihan sementara sebelum konfirmasi
25
26    running = True
27    while running:
28        screen.blit(background, (0, 0))
29
30        # Gambar karakter
31        screen.blit(chara_1_image, chara_1_rect)
32        screen.blit(chara_2_image, chara_2_rect)
33        screen.blit(chara_3_image, chara_3_rect)
34
35        # Gambar border di sekitar karakter yang dipilih sementara
36        if temp_selected_character == "chara_1":
37            pygame.draw.rect(screen, (255, 255, 255), chara_1_rect.inflate(10, 10), 3)
38        elif temp_selected_character == "chara_2":
39            pygame.draw.rect(screen, (255, 255, 255), chara_2_rect.inflate(10, 10), 3)
40        elif temp_selected_character == "chara_3":
41            pygame.draw.rect(screen, (255, 255, 255), chara_3_rect.inflate(10, 10), 3)
42
43        confirm_button.draw(screen)
44
45        pygame.display.flip()
46
47        for event in pygame.event.get():
48            if event.type == pygame.QUIT:
49                pygame.quit()
50                sys.exit()
51
52            if event.type == pygame.MOUSEBUTTONDOWN and event.button == 1:
53                if chara_1_rect.collidepoint(event.pos):
54                    temp_selected_character = "chara_1"
55                elif chara_2_rect.collidepoint(event.pos):
56                    temp_selected_character = "chara_2"
57                elif chara_3_rect.collidepoint(event.pos):
58                    temp_selected_character = "chara_3"
59
60            if confirm_button.is_clicked(event) and temp_selected_character:
61                selected_character = temp_selected_character
62                running = False # Keluar dari loop untuk kembali ke menu utama
```

- Fungsi ini untuk pemain memilih karakter yang akan digunakan

7. Logika dan Inti Permainan

a. Fungsi Utama (main)



```

1  def main():
2      pygame.mixer.music.play(-1)
3
4      global start_ticks, coin  # Tambahan ini agar dapat mengubah variabel global
5      start_ticks = pygame.time.get_ticks()  # Reset skor
6
7      coin = None
8
9      if not selected_character:
10         character_selection()
11
12     # Pastikan karakter sudah dipilih sebelum memulai permainan
13     if not selected_character:
14         print("No character selected! Exiting...")
15         pygame.quit()
16         sys.exit()
17
18     # Buat objek player menggunakan karakter yang dipilih
19     player = Player(screen_width // 2, screen_height - 50, selected_character)
20
21     level = 1
22     level_increase_timer = pygame.time.get_ticks()
23     power_ups = []
24     enemies = []
25
26     backgrounds = [
27         pygame.image.load("assets/bg_level_1.png"),
28         pygame.image.load("assets/bg_level_2.png"),
29         pygame.image.load("assets/bg_level_3.png"),
30         pygame.image.load("assets/bg_level_4.png"),
31     ]
32     backgrounds = [pygame.transform.scale(bg, (screen_width, screen_height)) for bg in backgrounds]
33     current_background_index = 0
34     current_background_y = 0  # Posisi awal background (untuk efek bergerak)
35
36     health_images = load_health_images(220, 180)
37     player_speed = 5  # Kecepatan default pemain
38     shield_active = False
39     shield_timer = 0
40     slow_active = False
41     slow_timer = 0
42     next_enemy_time = pygame.time.get_ticks() + random.randint(1000, 3000)
43     next_powerup_time = pygame.time.get_ticks() + random.randint(3000, 8000)
44
45     running = True
46     while running:
47         current_time = pygame.time.get_ticks()
48
49         for event in pygame.event.get():
50             if event.type == pygame.QUIT:
51                 running = False

```

```

53     # Tingkatkan level setiap 10 detik
54     if current_time - level_increase_timer > 10000:
55         level += 1
56         level_increase_timer = current_time
57
58         if level % 3 == 0 and current_background_index < len(backgrounds) - 1:
59             current_background_index += 1
60
61     if level == 2 and 20000 <= current_time - start_ticks <= 100000 and not coin:
62         coin = Coin()
63
64     if coin is not None:
65         coin.update()
66         if player.rect.colliderect(coin.rect):
67             pygame.mixer.music.stop()
68             you_win()
69             return
70
71     # Spawn musuh secara acak berdasarkan waktu
72     if current_time >= next_enemy_time:
73         enemy_type = random.choice(["rock", "fire", "bomb"])
74         new_enemy = Enemy(enemy_type)
75         new_enemy.speed = random.randint(2 + level, 5 + level)
76         new_enemy.movement_pattern = random.choice(["zigzag", "linear", "random"])
77         enemies.append(new_enemy)
78
79         next_enemy_time = current_time + random.randint(max(500, 3000 - level * 200), max(1000, 5000 - level * 300))
80
81     # Spawn power-up secara acak berdasarkan waktu
82     if current_time >= next_powerup_time:
83         power_up_type = random.choice(["health", "slow", "shield"])
84         power_ups.append(PowerUp(effect=power_up_type))
85         next_powerup_time = current_time + random.randint(3000, 8000)
86
87     # Efek shield
88     if shield_active and current_time - shield_timer > 3000:
89         shield_active = False
90
91     # Efek slow
92     if slow_active and current_time - slow_timer > 3000:
93         slow_active = False
94         player_speed = 5
95
96     # Perbarui pemain dan objek lain
97     player.update(speed=player_speed if not shield_active else 5)
98     for enemy in enemies[:]:
99         enemy.update()
100        if player.rect.colliderect(enemy.rect):
101            if not shield_active:
102                hit_sound.play()
103                player.health -= 1
104                enemies.remove(enemy)
105            if player.health <= 0:
106                pygame.mixer.music.stop()
107                game_over_sound.play()
108                game_over()
109                return

```

```

110     for power_up in power_ups[:]:
111         power_up.update()
112         if player.rect.colliderect(power_up.rect):
113             if power_up.effect == "health":
114                 player.health = min(player.health + 1, 3)
115             elif power_up.effect == "shield":
116                 shield_active = True
117                 shield_timer = current_time
118             elif power_up.effect == "slow":
119                 slow_active = True
120                 slow_timer = current_time
121                 player_speed = max(player_speed // 2, 1)
122             power_ups.remove(power_up)
123
124     # Gambar elemen ke layar
125     screen.fill(WHITE)
126     screen.blit(backgrounds[current_background_index], (0, current_background_y))
127     screen.blit(backgrounds[current_background_index], (0, current_background_y - screen_height))
128
129     player.draw(screen, shield_active)
130     for enemy in enemies:
131         enemy.draw(screen)
132     for power_up in power_ups:
133         power_up.draw(screen)
134     if coin:
135         coin.draw(screen)
136
137
138     score = (pygame.time.get_ticks() - start_ticks) // 1000
139     score_text = font.render(f"Score {score}", True, WHITE)
140     level_text = font.render(f"Level {level}", True, WHITE)
141     screen.blit(score_text, (10, 10))
142     screen.blit(level_text, (10, 50))
143     draw_health_bar(screen, player.health, health_images)
144
145     pygame.display.flip()
146     clock.tick(60)
147
148     pygame.quit()
149     sys.exit()

```

- Fungsi ini berisi seluruh logika inti permainan, termasuk:
 - Mengatur objek pemain, musuh, dan power-up.
 - Menangani tabrakan antara objek.
 - Menggambar elemen permainan ke layar.
 - Menambah skor, level, dan mengurangi kesehatan pemain.

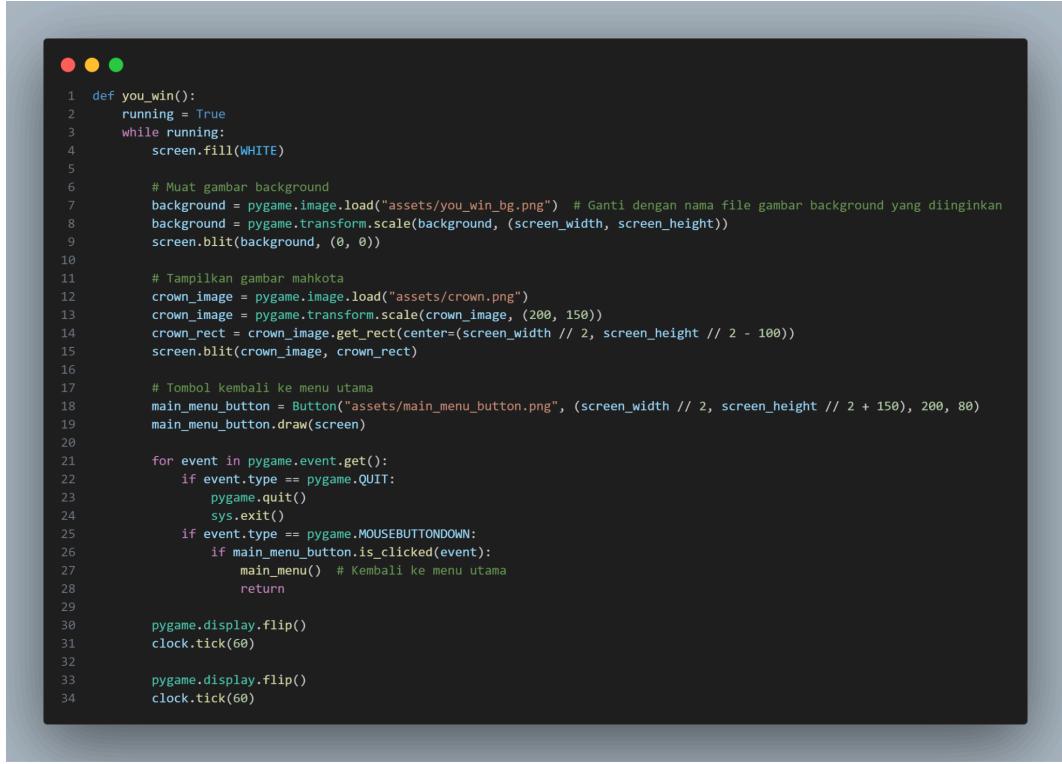
8. Akhir Permainan

a. Game Over

```
● ● ●
1 def game_over():
2     global start_ticks # Agar bisa akses skor terakhir
3     final_score = (pygame.time.get_ticks() - start_ticks) // 1000 # Hitung skor akhir
4
5     pygame.mixer.music.load("assets/game_over_bg.mp3")
6     pygame.mixer.music.play(-1)
7
8     # Muat gambar background untuk game over
9     background = pygame.image.load("assets/bggameover.png")
10    background = pygame.transform.scale(background, (screen_width, screen_height))
11
12    retry_button = Button("assets/retry_button.png", (screen_width // 2, 300), 200, 80)
13    menu_button = Button("assets/menu_button.png", (screen_width // 2, 430), 200, 80)
14
15    while True:
16        screen.blit(background, (0, 0)) # Gambar background
17
18        draw_text("Game Over!", font, WHITE, screen, screen_width // 2, 200)
19        draw_text(f"Your Score {final_score}", font, WHITE, screen, screen_width // 2, 150) # Tampilkan skor terakhir
20
21        retry_button.draw(screen)
22        menu_button.draw(screen)
23
24        pygame.display.flip()
25
26        for event in pygame.event.get():
27            if event.type == pygame.QUIT:
28                pygame.quit()
29                sys.exit()
30            if retry_button.is_clicked(event):
31                pygame.mixer.music.stop()
32                pygame.mixer.music.load("assets/backsound.mp3") # Muat ulang backsound awal
33                pygame.mixer.music.set_volume(0.5)
34                main() # Mulai ulang game
35                return
36            if menu_button.is_clicked(event):
37                pygame.mixer.music.stop()
38                main_menu()
39                return
```

- Fungsi ini menampilkan layar akhir jika pemain kalah:
 - Menampilkan teks seperti "Game Over".
 - Memberikan opsi untuk mengulang atau keluar.

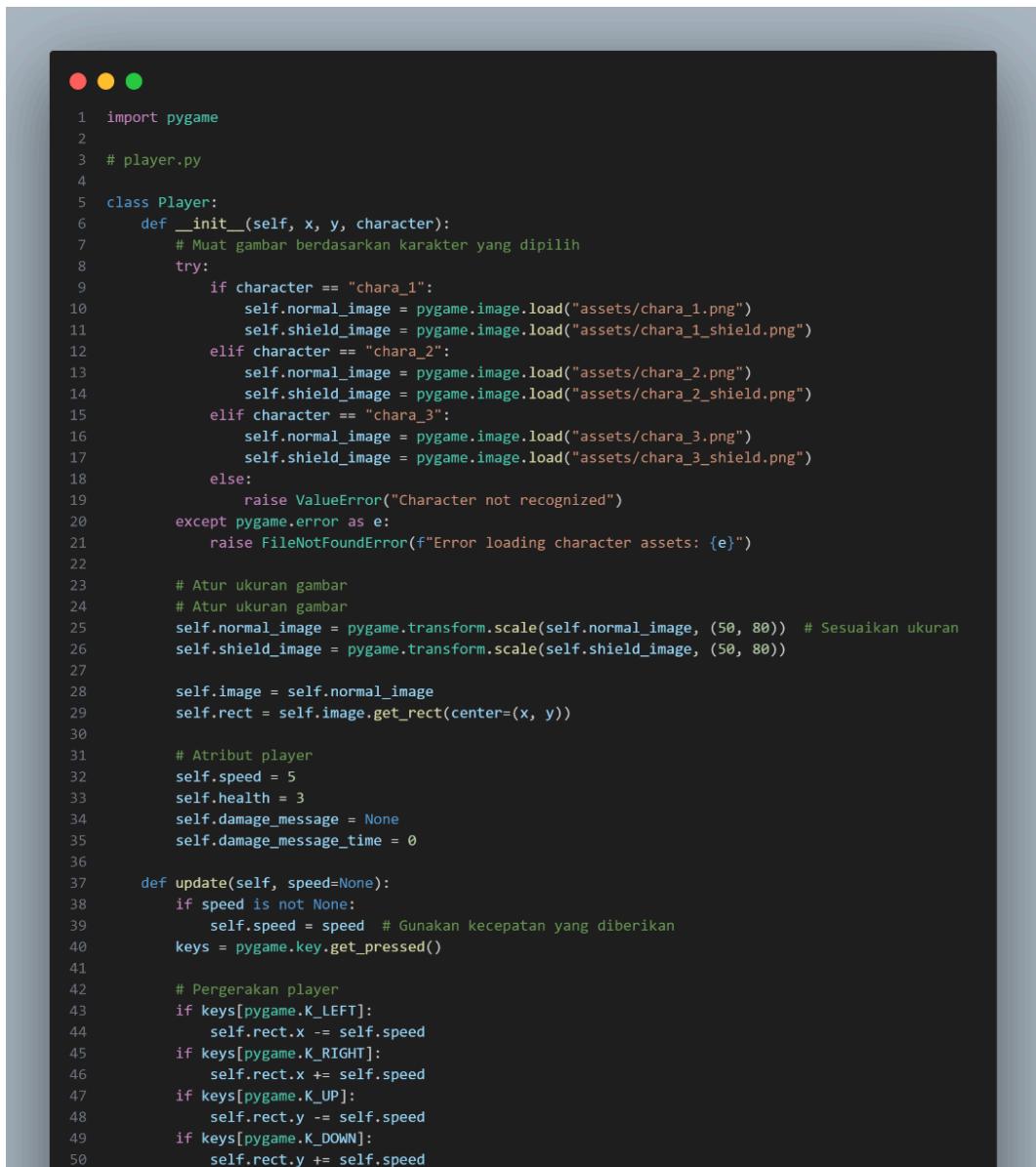
b. You Win



```
 1 def you_win():
 2     running = True
 3     while running:
 4         screen.fill(WHITE)
 5
 6         # Muat gambar background
 7         background = pygame.image.load("assets/you_win_bg.png") # Ganti dengan nama file gambar background yang diinginkan
 8         background = pygame.transform.scale(background, (screen_width, screen_height))
 9         screen.blit(background, (0, 0))
10
11         # Tampilkan gambar mahkota
12         crown_image = pygame.image.load("assets/crown.png")
13         crown_image = pygame.transform.scale(crown_image, (200, 150))
14         crown_rect = crown_image.get_rect(center=(screen_width // 2, screen_height // 2 - 100))
15         screen.blit(crown_image, crown_rect)
16
17         # Tombol kembali ke menu utama
18         main_menu_button = Button("assets/main_menu_button.png", (screen_width // 2, screen_height // 2 + 150), 200, 80)
19         main_menu_button.draw(screen)
20
21         for event in pygame.event.get():
22             if event.type == pygame.QUIT:
23                 pygame.quit()
24                 sys.exit()
25             if event.type == pygame.MOUSEBUTTONDOWN:
26                 if main_menu_button.is_clicked(event):
27                     main_menu() # Kembali ke menu utama
28                     return
29
30             pygame.display.flip()
31             clock.tick(60)
32
33             pygame.display.flip()
34             clock.tick(60)
```

- Fungsi ini menampilkan layar kemenangan jika pemain berhasil menyelesaikan permainan:
 - Menampilkan skor akhir pemain.

Player.py



```
1 import pygame
2
3 # player.py
4
5 class Player:
6     def __init__(self, x, y, character):
7         # Muat gambar berdasarkan karakter yang dipilih
8         try:
9             if character == "chara_1":
10                 self.normal_image = pygame.image.load("assets/chara_1.png")
11                 self.shield_image = pygame.image.load("assets/chara_1_shield.png")
12             elif character == "chara_2":
13                 self.normal_image = pygame.image.load("assets/chara_2.png")
14                 self.shield_image = pygame.image.load("assets/chara_2_shield.png")
15             elif character == "chara_3":
16                 self.normal_image = pygame.image.load("assets/chara_3.png")
17                 self.shield_image = pygame.image.load("assets/chara_3_shield.png")
18             else:
19                 raise ValueError("Character not recognized")
20         except pygame.error as e:
21             raise FileNotFoundError(f"Error loading character assets: {e}")
22
23     # Atur ukuran gambar
24     # Atur ukuran gambar
25     self.normal_image = pygame.transform.scale(self.normal_image, (50, 80)) # Sesuaikan ukuran
26     self.shield_image = pygame.transform.scale(self.shield_image, (50, 80))
27
28     self.image = self.normal_image
29     self.rect = self.image.get_rect(center=(x, y))
30
31     # Atribut player
32     self.speed = 5
33     self.health = 3
34     self.damage_message = None
35     self.damage_message_time = 0
36
37     def update(self, speed=None):
38         if speed is not None:
39             self.speed = speed # Gunakan kecepatan yang diberikan
40         keys = pygame.key.get_pressed()
41
42         # Pergerakan player
43         if keys[pygame.K_LEFT]:
44             self.rect.x -= self.speed
45         if keys[pygame.K_RIGHT]:
46             self.rect.x += self.speed
47         if keys[pygame.K_UP]:
48             self.rect.y -= self.speed
49         if keys[pygame.K_DOWN]:
50             self.rect.y += self.speed
```

```

52         # Batas pergerakan
53         screen_width = 800
54         screen_height = 520
55         self.rect.left = max(self.rect.left, 0)
56         self.rect.right = min(self.rect.right, screen_width)
57         self.rect.top = max(self.rect.top, screen_height // 1.5)
58         self.rect.bottom = min(self.rect.bottom, screen_height)
59
60     def draw(self, screen, shield_active):
61         # Tentukan gambar berdasarkan status shield
62         if shield_active:
63             self.image = self.shield_image
64         else:
65             self.image = self.normal_image
66         screen.blit(self.image, self.rect)
67
68     def handle_collision(self, enemy):
69         """Mengurangi health player ketika terjadi tabrakan dengan musuh."""
70         if hasattr(enemy, 'damage') and isinstance(enemy.damage, (int, float)) and enemy.damage > 0:
71             self.health -= enemy.damage
72             self.damage_message = f"-{enemy.damage}"
73             self.damage_message_time = pygame.time.get_ticks()
74         else:
75             print("Warning: Enemy object must have a 'damage' attribute with a positive value.")
76
77     def draw_damage_message(self, screen, enemy):
78         """Menampilkan pesan damage di atas musuh selama 1 detik."""
79         if self.damage_message and pygame.time.get_ticks() - self.damage_message_time < 1000:
80             font = pygame.font.Font(None, 36)
81             text = font.render(self.damage_message, True, (255, 0, 0))
82             screen.blit(text, (enemy.rect.centerx - text.get_width() // 2, enemy.rect.top - 20))
83         else:
84             self.damage_message = None

```

- Inisialisasi Pemain (`__init__`):
 - Memuat gambar karakter (normal dan dengan shield) berdasarkan karakter yang dipilih.
 - Menskalakan gambar ke ukuran tertentu dan menentukan posisi awal.
 - Menyimpan atribut seperti kecepatan, kesehatan, dan pesan kerusakan.
- Metode update:
 - Mengatur pergerakan pemain berdasarkan input tombol panah (arrow keys).
 - Membatasi pergerakan agar pemain tidak keluar dari area permainan.
- Metode draw:

- Menampilkan gambar pemain di layar, berganti antara gambar normal atau dengan shield berdasarkan status.
- Metode handle_collision:
 - Mengurangi kesehatan pemain jika terjadi tabrakan dengan musuh yang memiliki atribut damage.
 - Menyimpan pesan kerusakan yang akan ditampilkan.
- Metode draw_damage_message:
 - Menampilkan pesan kerusakan di atas musuh selama 1 detik setelah tabrakan.

Enemy.py

1. Class Enemy



```

1  class Enemy:
2      def __init__(self, enemy_type, width=62, height=62):
3          self.x = random.randint(0, 800 - width)
4          self.y = random.randint(-150, -50)
5          self.speed = random.randint(2, 5)
6          self.direction = random.choice([-1, 1]) # Gerakan horizontal (kiri/kanan)
7          self.type = enemy_type
8
9          self.width = width
10         self.height = height
11
12         self.images = {
13             "bomb": "assets/bomb.png",
14             "fire": "assets/fire.png",
15             "rock": "assets/rock.png",
16         }
17
18         self.damage = {"bomb": -2, "fire": -1, "rock": -1}
19
20         self.image = pygame.image.load(self.images.get(self.type, "assets/rock.png"))
21         self.image = pygame.transform.scale(self.image, (self.width, self.height))
22         self.rect = self.image.get_rect(center=(self.x, self.y))
23
24     def update(self):
25         """Update posisi musuh dengan gerakan acak."""
26         self.y += self.speed
27         self.x += self.direction * random.randint(1, 3) # Gerakan horizontal acak
28
29         # Reset posisi jika keluar layar
30         if self.y > 600 or self.x < 0 or self.x > 800:
31             self.y = random.randint(-150, -50)
32             self.x = random.randint(0, 800 - self.width)
33             self.speed = random.randint(2, 5)
34             self.direction = random.choice([-1, 1]) # Arah horizontal baru
35
36         self.rect.center = (self.x, self.y)
37
38     def draw(self, screen):
39         """Gambar musuh ke layar."""
40         screen.blit(self.image, self.rect)

```

‘Class Enemy’ ini merepresentasikan musuh dalam permainan, dengan atribut dan perilaku sebagai berikut:

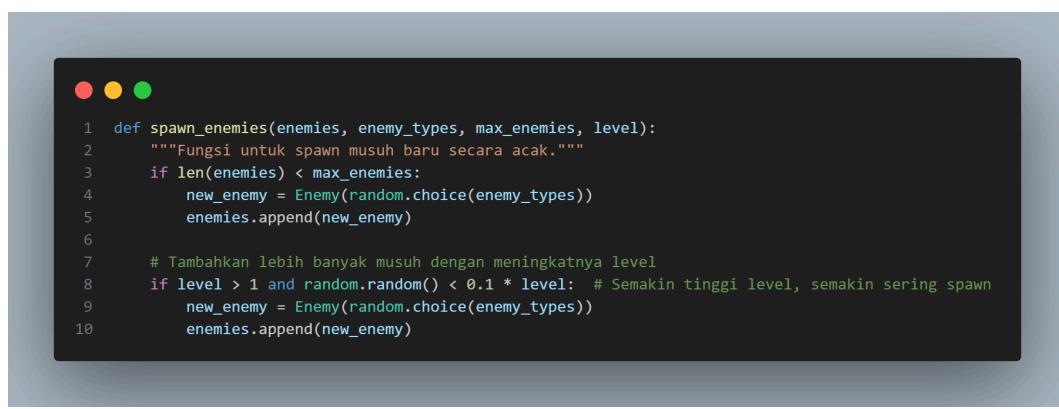
Atribut Utama:

- x, y: Posisi awal musuh di layar, diatur secara acak.
- speed: Kecepatan musuh (acak antara 2-5 piksel per frame).
- direction: Arah gerakan horizontal, acak (-1 untuk kiri, 1 untuk kanan).
- type: Jenis musuh (bomb, fire, atau rock) yang menentukan gambar dan damage.
- rect: Area musuh digunakan untuk mendeteksi tabrakan.

Metode:

- update:
 - Memperbarui posisi musuh dengan menambahkan gerakan vertikal dan horizontal.
 - Jika musuh keluar dari layar, posisinya di-reset ke atas layar dengan nilai acak.
- draw:
 - Menampilkan gambar musuh di layar dengan screen.blit().

2. Fungsi ‘spawn_enemies’

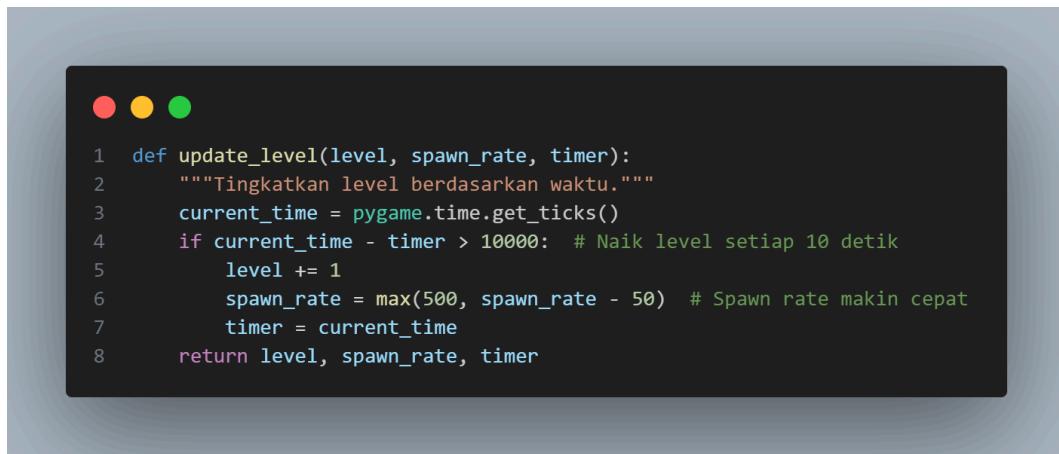


```
 1 def spawn_enemies(enemies, enemy_types, max_enemies, level):
 2     """Fungsi untuk spawn musuh baru secara acak."""
 3     if len(enemies) < max_enemies:
 4         new_enemy = Enemy(random.choice(enemy_types))
 5         enemies.append(new_enemy)
 6
 7     # Tambahkan lebih banyak musuh dengan meningkatnya level
 8     if level > 1 and random.random() < 0.1 * level: # Semakin tinggi level, semakin sering spawn
 9         new_enemy = Enemy(random.choice(enemy_types))
10         enemies.append(new_enemy)
```

Fungsi ini bertugas menambahkan musuh baru ke dalam daftar musuh dengan kondisi:

- Jika jumlah musuh kurang dari batas maksimum (max_enemies), musuh baru akan dibuat secara acak.
- Dengan meningkatnya level, peluang muncul musuh tambahan juga bertambah (10% per level).

3. Fungsi ‘update_level’



```
 1 def update_level(level, spawn_rate, timer):
 2     """Tingkatkan level berdasarkan waktu."""
 3     current_time = pygame.time.get_ticks()
 4     if current_time - timer > 10000: # Naik level setiap 10 detik
 5         level += 1
 6         spawn_rate = max(500, spawn_rate - 50) # Spawn rate makin cepat
 7         timer = current_time
 8     return level, spawn_rate, timer
```

Fungsi ini meningkatkan level permainan secara otomatis setiap 10 detik dan mempercepat spawn musuh:

- Timer: Menghitung waktu berdasarkan ‘pygame.time.get_ticks()’.
- Level: Naik setiap 10 detik.
- Spawn Rate: Berkurang dengan naiknya level hingga mencapai minimum 500 ms.

Powerup.py

```
 1 import pygame
 2 import random
 3
 4 class PowerUp:
 5     def __init__(self, width=50, height=50, effect=None):
 6         self.x = random.randint(0, 800)
 7         self.y = random.randint(-100, 0)
 8         self.speed = 2
 9
10     # Jika efek diberikan, gunakan efek tersebut; jika tidak, pilih secara acak
11     self.effect = effect if effect else random.choice(["health", "slow", "shield"])
12
13     # Gambar untuk setiap jenis efek
14     self.images = {
15         "health": "assets/health.png",
16         "slow": "assets/slow.png",
17         "shield": "assets/shield.png",
18     }
19
20     self.image = pygame.image.load(self.images[self.effect])
21     self.image = pygame.transform.scale(self.image, (width, height))
22     self.rect = self.image.get_rect(center=(self.x, self.y))
23
24     def update(self):
25         self.y += self.speed
26         if self.y > 600:
27             self.y = random.randint(-100, 0)
28             self.x = random.randint(0, 800)
29             self.rect.center = (self.x, self.y)
30
31     def draw(self, screen):
32         screen.blit(self.image, self.rect.topleft)
```

- Inisialisasi (`__init__`):
 - Membuat objek power-up dengan posisi acak di layar.
 - Menentukan efek (misalnya, health, slow, atau shield) yang dipilih secara acak jika tidak diberikan.
 - Memuat gambar sesuai efek yang dipilih dan mengatur ukuran serta posisi objek.
- Metode update:
 - Memperbarui posisi power-up setiap frame dengan gerakan vertikal ke bawah.
 - Jika power-up melewati batas bawah layar, ia akan muncul kembali di atas layar dengan posisi acak.

- Metode draw:
 - Menampilkan gambar power-up di layar pada posisi yang diperbarui.

VIII. KESIMPULAN

Game "Escape the Enemy" adalah bukti nyata bahwa teori Pemrograman Berorientasi Objek dapat digunakan untuk membangun aplikasi modular dan efektif. Dengan menerapkan konsep OOP secara konsisten, pengembang berhasil menciptakan game yang tidak hanya menyenangkan untuk dimainkan tetapi juga dirancang dengan struktur kode yang kuat dan mudah dikelola.

Fitur-fitur seperti power-up, musuh dengan pola gerakan unik, dan level progresif menciptakan gameplay yang menantang dan dinamis. Di balik layar, penerapan prinsip OOP seperti enkapsulasi, inheritance, polimorfisme, dan abstraksi memastikan bahwa game ini tidak hanya dapat dimainkan dengan lancar tetapi juga mudah untuk diperbarui dan dikembangkan di masa depan.

Keberhasilan "Escape the Enemy" dalam mengintegrasikan elemen-elemen ini menunjukkan potensi besar OOP dalam pengembangan perangkat lunak, terutama untuk proyek-proyek berskala besar yang memerlukan struktur kode yang modular dan fleksibel. Game ini dapat menjadi referensi yang berharga bagi mahasiswa, pengembang pemula, atau siapa pun yang ingin memahami bagaimana konsep OOP dapat diterapkan dalam dunia nyata.

Untuk pengembangan lebih lanjut, game ini memiliki potensi untuk ditingkatkan dengan menambahkan fitur-fitur seperti leaderboard, mode permainan tambahan, dan grafis yang lebih kompleks, sehingga semakin memperkaya pengalaman bermain bagi pengguna.

Game ini juga menjadi contoh penerapan teori OOP, meliputi:

- Inheritance: Digunakan untuk mengelompokkan berbagai jenis karakter dan musuh dengan sifat dan perilaku yang berbeda.
- Polymorphism: Memungkinkan perilaku objek berbeda, seperti pola gerakan musuh (zigzag, linear, random).
- Encapsulation: Menjaga integritas data karakter dan musuh dengan membatasi akses langsung ke properti tertentu.
- Abstraction: Menyembunyikan detail implementasi tertentu, seperti logika power-up, sehingga fokus pada antarmuka pengguna.

Link GitHub:

<https://github.com/MuhamadRifkyRamdhani/Escape-The-Enemy.git>