

PORTOFOLIO TP-PBO 2025

A. Identitas Proyek

Nama Mahasiswa:

Muhammad Akbar Riyan Hartoto

Muhammad Miko Ardiansyah

Bintang Alfathir Daud

NIM:

2400018052

2400018056

2400018054

Program Studi / Kelas: Informatika / A

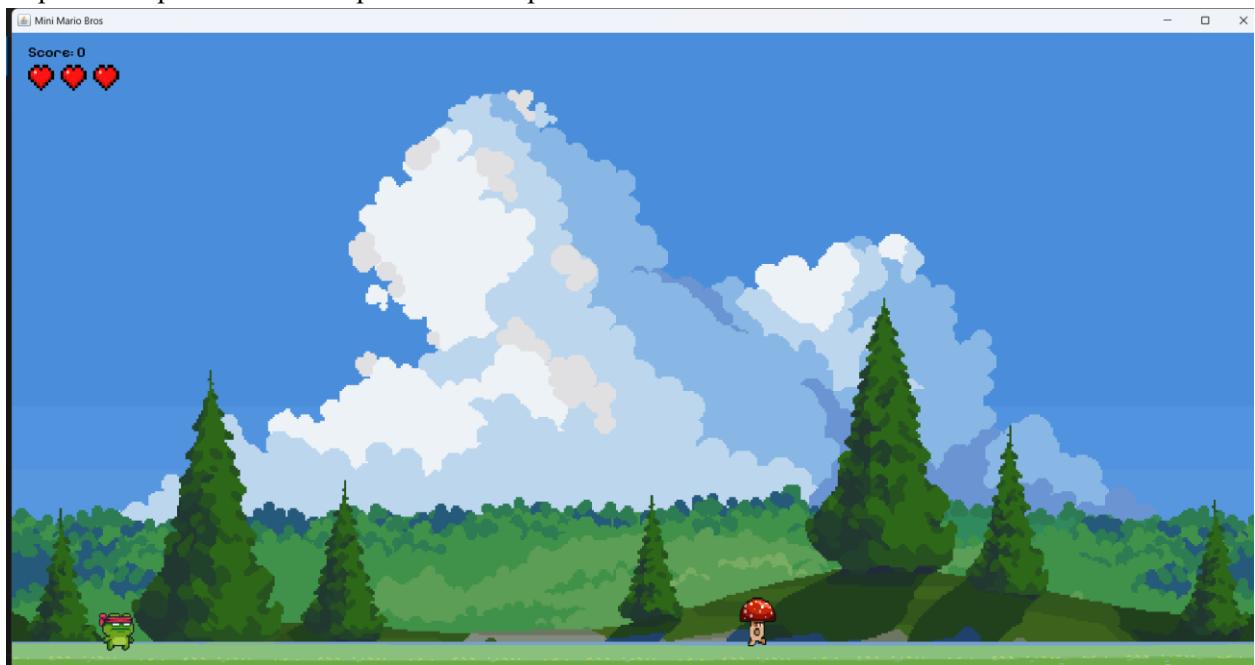
Mata Kuliah: Pemrograman Berorientasi Objek

Tahun: 2025

Judul Proyek: Game platformers 2 dimensi berbasis Java mengungkapkan konsep pemrograman berorientasi objek

Link Repository (GitHub/GitLab): [Miko-Ard/Project_PBO](#)

Capture Tampilan Awal / Tampilan Utama Aplikasi:



B. Persoalan Bisnis dan Deskripsi Proyek

Persoalan Bisnis :

Perkembangan industri hiburan digital, khususnya game 2D sederhana, semakin pesat dan menjadi salah satu media hiburan yang banyak diminati oleh berbagai kalangan, termasuk pelajar dan mahasiswa. Namun, terdapat beberapa persoalan bisnis yang sering muncul, antara lain:

1. Minimnya Game Edukatif yang Sederhana dan Ringan

Banyak game yang tersedia saat ini membutuhkan spesifikasi perangkat yang tinggi, sehingga tidak semua pengguna dapat mengaksesnya dengan nyaman. Hal ini menjadi kendala bagi pengguna dengan perangkat terbatas.

2. Kurangnya Media Pembelajaran Pemrograman Berbasis Proyek Nyata

Dalam dunia pendidikan, pembelajaran pemrograman sering kali hanya berfokus pada teori dan contoh kecil, sehingga mahasiswa kesulitan memahami penerapan konsep Object-Oriented Programming (OOP), event handling, dan game loop dalam proyek nyata.

3. Rendahnya Interaktivitas dan Motivasi Pengguna

Aplikasi atau game yang tidak memiliki sistem skor, nyawa, animasi, dan efek suara cenderung kurang menarik, sehingga pengguna cepat merasa bosan.

4. Keterbatasan Game Lokal Berbasis Desktop Java

Saat ini, game desktop berbasis Java masih relatif sedikit dibandingkan platform lain, padahal Java memiliki potensi besar untuk pengembangan game 2D sederhana yang stabil dan cross-platform.

Deskripsi Proyek :

Proyek ini merupakan pengembangan sebuah game 2D platformer sederhana berbasis Java Swing yang berjudul “Mini Mario Bros”. Game ini dirancang sebagai media hiburan ringan sekaligus sarana pembelajaran pemrograman berbasis objek.

Dalam game ini, pemain mengendalikan karakter utama untuk bergerak ke kiri, kanan, dan melompat guna menghindari atau mengalahkan musuh. Sistem permainan dilengkapi dengan mekanisme skor, nyawa (lives), animasi karakter, collision detection, efek suara, serta tampilan game over dan respawn.

Proyek ini mengimplementasikan berbagai konsep penting dalam pengembangan perangkat lunak, antara lain:

1. Object-Oriented Programming (OOP)
2. Game loop menggunakan Timer
3. Event handling (keyboard dan mouse)
4. Manajemen state (player, enemy, game over)

5. Animasi sprite berbasis frame
6. Collision detection menggunakan bounding box

Tujuan Proyek

Tujuan dari proyek ini adalah:

1. Mengembangkan game desktop 2D yang ringan dan mudah dimainkan.
2. Menerapkan konsep OOP dalam pengembangan aplikasi nyata.
3. Menjadi media pembelajaran untuk memahami alur kerja game (update, render, collision).
4. Meningkatkan pengalaman pengguna melalui animasi, skor, dan efek suara.

Manfaat Proyek

Manfaat yang diharapkan dari proyek ini:

1. Sebagai media hiburan sederhana bagi pengguna.
2. Sebagai contoh implementasi game 2D berbasis Java.
3. Membantu mahasiswa memahami penerapan konsep pemrograman secara praktis.
4. Menjadi dasar pengembangan game yang lebih kompleks di masa depan.

C. Daftar Seluruh Spesifikasi Aplikasi

1. Spesifikasi Umum Aplikasi

- Nama Aplikasi: Mini Mario Bros
- Jenis Aplikasi: Game Desktop 2D
- Platform: Desktop (Windows / Linux / macOS)
- Bahasa Pemrograman: Java
- Framework / Library: Java Swing, Java AWT
- Tipe Game: Platfromer 2D
- Target Pengguna: Pelajar, Mahasiswa, dan Pengguna Umum

2. Spesifikasi Fungsional

2.1 Kontrol Pemain

- Pemain dapat bergerak ke kiri menggunakan tombol A atau ←
- Pemain dapat bergerak ke kanan menggunakan tombol D atau →
- Pemain dapat melompat menggunakan tombol W, ↑, atau Space
- Pemain tidak dapat melompat saat berada di udara

2.2 Mekanisme Permainan

- Pemain memiliki 3 nyawa (lives)
- Pemain akan kehilangan nyawa jika menyentuh musuh dari samping
- Pemain dapat mengalahkan musuh dengan melompat dari atas
- Musuh akan respawn setelah dikalahkan
- Game berakhir ketika nyawa pemain habis (Game Over)

2.3 Sistem Skor

- Skor bertambah setiap kali pemain mengalahkan musuh
- Sistem menghitung jumlah musuh yang dikalahkan (kill count)
- Skor dan kill count ditampilkan secara real-time

- Skor akan di-reset saat game dimulai ulang

2.4 Animasi

- Animasi player:
 1. Idle
 2. Run
 3. Jump
 4. Fall
 5. Hit
- Animasi musuh:
 1. Idle
 2. Run
 3. Die
- Animasi berbasis sprite sheet dan frame

2.5 Efek Visual

- Background statis
- Tanah (ground) dengan sprite khusus
- Floating text saat pemain mengalahkan musuh
- Tampilan Game Over dengan tombol restart
- Indikator nyawa menggunakan ikon hati

2.6 Efek Suara

- Efek suara saat pemain melompat
- Efek suara saat musuh dikalahkan
- Manajemen audio menggunakan Java Sound API

2.7 Input dan Event Handling

- Input keyboard menggunakan KeyListener
- Input mouse untuk tombol restart menggunakan MouseListener
- Game loop menggunakan Timer

3. Spesifikasi Non-Fungsional

3.1 Performa

- Game berjalan pada kecepatan tetap menggunakan interval Timer 20 ms
- Aplikasi ringan dan tidak membutuhkan spesifikasi tinggi
- Tidak memerlukan koneksi internet

3.2 Keamanan dan Stabilitas

- Validasi input keyboard
- Penanganan error saat load asset
- Fallback font jika font utama gagal dimuat

3.3 Usability

- Kontrol sederhana dan mudah dipahami
- Tampilan visual jelas dan informatif
- Feedback visual dan audio saat interaksi

4. Spesifikasi Teknis

4.1 Resolusi dan Tampilan

- Resolusi default: 1920×1020
- Mendukung resize window
- Rendering menggunakan paintComponent (Graphics g)

4.2 Struktur Kelas

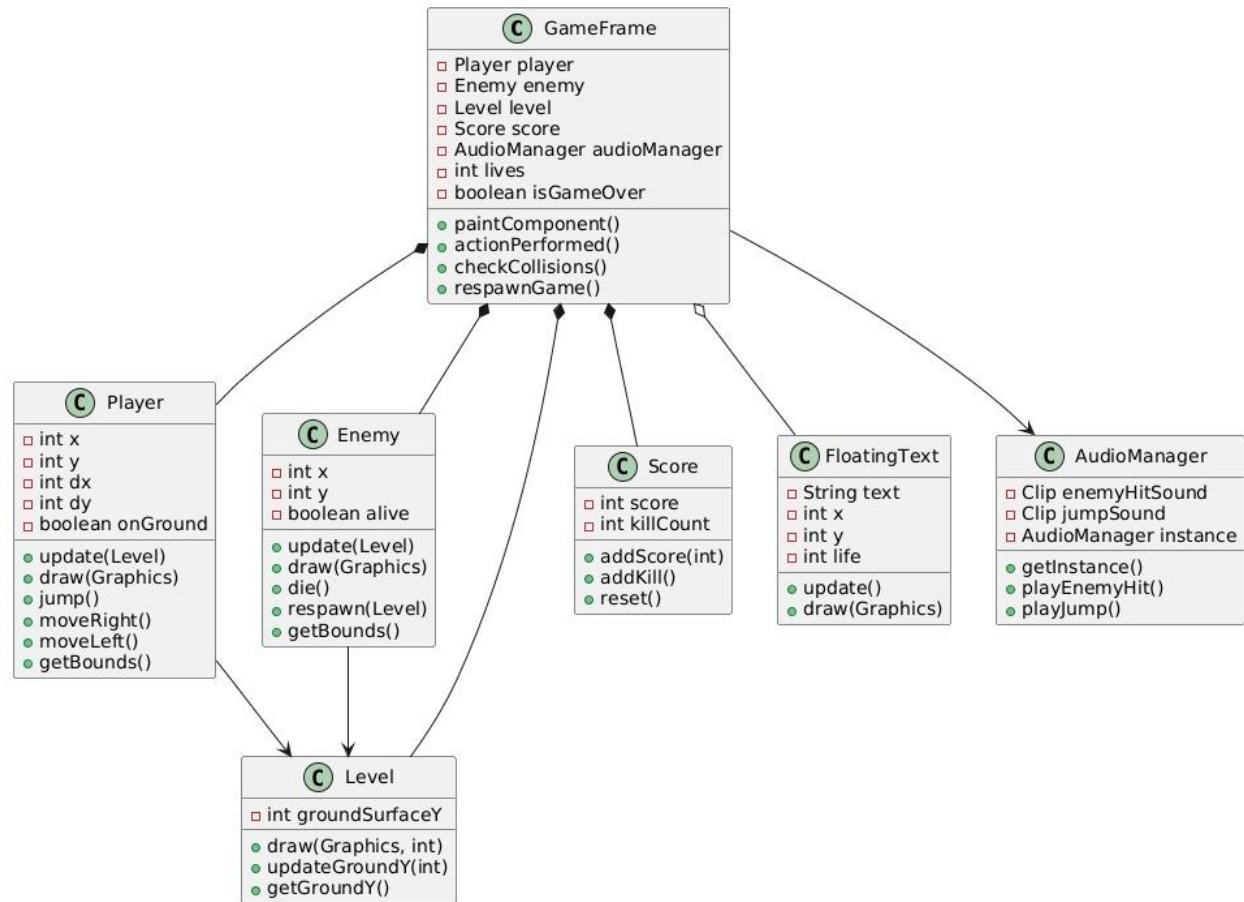
- GameFrame → Mengatur game loop, input, dan rendering
- Player → Mengatur pergerakan dan animasi pemain
- Enemy → Mengatur musuh dan animasinya
- Level → Mengatur posisi dan tampilan tanah
- Score → Mengatur skor dan kill count
- FloatingText → Menampilkan teks animasi
- AudioManager → Mengatur efek suara

5. Spesifikasi Asset

- Format gambar: PNG

- Format audio: WAV
 - Font: TrueType Font (TTF)
 - Asset disimpan dalam folder /assets
6. Spesifikasi Pengembangan
- Paradigma pemrograman: Object-Oriented Programming (OOP)
 - Menggunakan konsep:
 1. Encapsulation
 2. Abstraction
 3. Enum
 4. Event-driven programming
 - Mudah dikembangkan untuk fitur lanjutan (level baru, boss, item)
7. Spesifikasi Keterbatasan
- Belum mendukung multiplayer
 - Belum menggunakan physics engine khusus
 - Belum ada penyimpanan skor permanen (database / file)

D. Rancangan Model Diagram UML



E. Rancangan Antar Muka Berbasis GUI

1. Gambaran Umum Antarmuka

Aplikasi Mini Mario Bros menggunakan antarmuka berbasis Graphical User Interface (GUI) yang dikembangkan dengan Java Swing. Antarmuka dirancang secara sederhana namun interaktif, dengan fokus pada kemudahan penggunaan dan pengalaman bermain yang nyaman.

Seluruh elemen antarmuka ditampilkan dalam satu jendela utama (JFrame) yang berisi panel permainan (JPanel) sebagai area utama interaksi pengguna.

2. Struktur Antarmuka Utama

2.1 Jendela Utama (JFrame)

- Judul jendela: Mini Mario Bros
- Ukuran awal: 1920×1020
- Dapat di-resize
- Menjadi container utama untuk seluruh elemen GUI

2.2 Panel Permainan (Game Panel / JPanel)

Panel permainan merupakan area utama tempat seluruh komponen game dirender, meliputi:

- Background permainan
- Karakter pemain
- Musuh
- Tanah (ground)
- Floating text
- Skor dan nyawa
- Tampilan Game Over

Panel ini menggunakan metode `paintComponent(Graphics g)` untuk melakukan rendering.

3. Komponen GUI Utama

3.1 Background

- Background ditampilkan memenuhi seluruh area panel
- Berfungsi sebagai latar visual permainan
- Menggunakan gambar statis format PNG

3.2 Karakter Pemain (Player)

- Ditampilkan sebagai sprite animasi
- Memiliki beberapa state visual:
 1. Idle
 2. Run
 3. Jump
 4. Fall
 5. Hit
- Posisi berubah berdasarkan input keyboard

3.3 Musuh (Enemy)

- Ditampilkan sebagai sprite animasi

- Bergerak secara otomatis ke kiri dan kanan
- Memiliki animasi mati (die)

3.4 Tanah (Ground)

- Ditampilkan di bagian bawah layar
- Menjadi batas jatuh karakter
- Digambar menggunakan sprite khusus

3.5 Informasi Skor

- Ditampilkan di pojok kiri atas layar
- Menampilkan:
 1. Total skor
 2. Jumlah musuh yang dikalahkan (kill count)
- Menggunakan font khusus (Pixel Font)

3.6 Indikator Nyawa

- Ditampilkan menggunakan ikon hati
- Setiap ikon mewakili satu nyawa
- Berkurang saat pemain terkena musuh

3.7 Floating Text

- Teks animasi yang muncul saat musuh dikalahkan
- Bergerak ke atas dan memudar
- Digunakan sebagai feedback visual kepada pemain

4. Antarmuka Game Over

4.1 Tampilan Game Over

- Muncul saat nyawa pemain habis
- Ditampilkan di tengah layar
- Menggunakan frame visual khusus

4.2 Elemen Game Over

- Teks GAME OVER
- Informasi skor akhir
- Tombol Restart

4.3 Tombol Restart

- Dapat diklik menggunakan mouse
- Mengulang permainan dari awal
- Mengatur ulang skor, nyawa, dan posisi karakter

5. Interaksi Pengguna

5.1 Input Keyboard

- Mengontrol pergerakan dan aksi pemain
- Input langsung memengaruhi animasi dan posisi karakter

5.2 Input Mouse

- Digunakan untuk menekan tombol restart saat Game Over

6. Alur Interaksi Antarmuka

Aplikasi dijalankan

JFrame tampil

Panel permainan dirender

Player & Enemy muncul

Skor dan nyawa ditampilkan

Input keyboard aktif

Pemain bermain

GUI update setiap frame

Animasi berjalan

Feedback visual & audio muncul

Nyawa habis

Tampilan Game Over muncul

Pemain klik Restart

Game dimulai ulang

7. Prinsip Desain Antarmuka

- Sederhana dan intuitif
- Responsif terhadap input
- Feedback visual yang jelas
- Konsisten dalam penggunaan warna dan font
- Mudah dikembangkan untuk fitur tambahan

8. Keterbatasan Antarmuka

- Belum menggunakan menu utama terpisah
- Belum mendukung pengaturan (settings)
- Belum mendukung fullscreen otomatis

F. Skrip Program dan Penjelasannya

Class audioManager

Laptop file buka di bawah



AudioManager.java

```
src > J AudioManager.java > AudioManager
1  import javax.sound.sampled.*;
2  import java.io.File;
3  import java.net.URL;
4
5  /**
6   * Manages all audio playback for the game.
7   * Uses Java Sound API for audio playback.
8   */
9  public class AudioManager {
10      private Clip enemyHitSound;
11      private Clip jumpSound;
12
13      private static AudioManager instance;
14
15      private AudioManager() {
16          // Private constructor for singleton pattern.
17      }
18
19      public static AudioManager getInstance() {
20          if (instance == null) {
21              instance = new AudioManager();
22          }
23          return instance;
24      }
25
26      public void loadEnemyHitSound(String path) {
27          try {
28              URL soundUrl = getClass().getResource(path);
29              if (soundUrl == null) {
30                  System.out.println("Enemy hit sound file not found: " + path);
31                  return;
32              }
33
34              AudioInputStream audioIn = AudioSystem.getAudioInputStream(soundUrl);
35              enemyHitSound = AudioSystem.getClip();
36              enemyHitSound.open(audioIn);
37              System.out.println("Enemy hit sound loaded successfully.");
38          }
39      }
40
41  }
```

```
src > J AudioManager.java > AudioManager
  9  public class AudioManager {
26      public void loadEnemyHitSound(String path) {
37          System.out.println("Enemy hit sound loaded successfully.");
38      } catch (UnsupportedAudioFileException e) {
39          System.out.println("Audio format not supported for: " + path + " (Java Sound API o");
40      } catch (Exception e) {
41          System.out.println("Error loading enemy hit sound: " + e.getMessage());
42      }
43  }
44
45  public void loadJumpSound(String path) {
46      try {
47          URL soundUrl = getClass().getResource(path);
48          if (soundUrl == null) {
49              System.out.println("Jump sound file not found: " + path);
50              return;
51          }
52
53          AudioInputStream audioIn = AudioSystem.getAudioInputStream(soundUrl);
54          jumpSound = AudioSystem.getClip();
55          jumpSound.open(audioIn);
56          System.out.println("Jump sound loaded successfully.");
57      } catch (UnsupportedAudioFileException e) {
58          System.out.println("Audio format not supported for: " + path + " (Java Sound API o");
59      } catch (Exception e) {
60          System.out.println("Error loading jump sound: " + e.getMessage());
61      }
62  }
63
64  public void playEnemyHit() {
65      if (enemyHitSound == null) return;
66
67      try {
68          if (enemyHitSound.isRunning()) {
69              enemyHitSound.stop();
70          }
71          enemyHitSound.setFramePosition(frames: 0);
```

```
src > J AudioManager.java > AudioManager
  9  public class AudioManager {
 64      public void playEnemyHit() {
 71          enemyHitSound.setFramePosition(frames: 0);
 72          enemyHitSound.start();
 73      } catch (Exception e) {
 74          System.out.println("Error playing enemy hit sound: " + e.getMessage());
 75      }
 76  }
 77
 78  public void playJump() {
 79      if (jumpSound == null) return;
 80
 81      try {
 82          if (jumpSound.isRunning()) {
 83              jumpSound.stop();
 84          }
 85          jumpSound.setFramePosition(frames: 0);
 86          jumpSound.start();
 87      } catch (Exception e) {
 88          System.out.println("Error playing jump sound: " + e.getMessage());
 89      }
 90  }
 91
 92  public void dispose() {
 93      if (enemyHitSound != null) {
 94          enemyHitSound.close();
 95      }
 96      if (jumpSound != null) {
 97          jumpSound.close();
 98      }
 99  }
100 }
101
```

AudioManager.java penjelasan yang ada dalam sintaks file diatas

- Singleton : cuma 1 instance AudioManager di game.
- Clip enemyHitSound & jumpSound : simpan suara efek.
- loadEnemyHitSound(path) / loadJumpSound(path) : muat file audio.
- playEnemyHit() / playJump() : mainkan suara, mulai dari awal.
- dispose() : tutup clip, bersihkan memori.
- Error handling : file tidak ada atau format salah tetap aman.

Class Player

Buka file di laptop



Player.java

```
src > J Player.java > ...
1 import javax.swing.*;
2 import java.awt.*;
3 import java.net.URL;
4
5 /**
6  * Represents the player character, handling its state, movement, and animation.
7  */
8 public class Player {
9
10    private enum AnimationState {
11        IDLE, RUN, JUMP, FALL, HIT
12    }
13
14    private int x, y;
15    private final int width = 60, height = 60;
16    private int dx = 0, dy = 0;
17    private boolean onGround = false;
18    private boolean facingRight = true;
19
20    private AnimationState currentState = AnimationState.IDLE;
21    private int hitTimer = 0;
22
23    private Image idleSheet, runSheet, jumpSheet, fallSheet, hitSheet;
24
25    private final int[] frameCounts = {11, 12, 1, 1, 7}; // Corresponds to AnimationState enum
26    private final int[] animSpeeds = {2, 2, 1, 1, 1};
27
28    private int frameIndex = 0;
29    private int animCounter = 0;
30
31    public Player(int x, int y) {
32        this.x = x;
33        this.y = y;
34        loadSpriteSheets();
35    }
36
37    private void loadSpriteSheets() {
```

```
src > J Player.java > ...
8  public class Player {
37      private void loadSpriteSheets() {
38          try {
39              URL idleUrl = getClass().getResource(name: "/assets/idle.png");
40              if (idleUrl != null) idleSheet = new ImageIcon(idleUrl).getImage();
41
42              URL runUrl = getClass().getResource(name: "/assets/run.png");
43              if (runUrl != null) runSheet = new ImageIcon(runUrl).getImage();
44
45              URL jumpUrl = getClass().getResource(name: "/assets/Jump (32x32).png");
46              if (jumpUrl != null) jumpSheet = new ImageIcon(jumpUrl).getImage();
47
48              URL fallUrl = getClass().getResource(name: "/assets/fall.png");
49              if (fallUrl != null) fallSheet = new ImageIcon(fallUrl).getImage();
50
51              URL hitUrl = getClass().getResource(name: "/assets/hit.png");
52              if (hitUrl != null) hitSheet = new ImageIcon(hitUrl).getImage();
53          } catch (Exception e) {
54              System.err.println(x: "Failed to load player sprite sheets.");
55              e.printStackTrace();
56          }
57      }
58
59      public void update(Level level) {
60          x += dx;
61          dy += 1; // Gravity
62          y += dy;
63
64          if (y >= level.getGroundY() - height) {
65              y = level.getGroundY() - height;
66              dy = 0;
67              onGround = true;
68          } else {
69              onGround = false;
70          }
71
72          if (hitTimer > 0) {
```

```
59     public void update(Level level) {
60         if (hitTimer > 0) {
61             hitTimer--;
62         }
63
64         updateState();
65         updateAnimation();
66     }
67
68     private void updateState() {
69         if (hitTimer > 0) {
70             currentState = AnimationState.HIT;
71         } else if (!onGround) {
72             currentState = (dy < 0) ? AnimationState.JUMP : AnimationState.FALL;
73         } else {
74             currentState = (dx != 0) ? AnimationState.RUN : AnimationState.IDLE;
75         }
76     }
77
78     private void updateAnimation() {
79         animCounter++;
80         int stateIndex = currentState.ordinal();
81         if (animCounter >= animSpeeds[stateIndex]) {
82             animCounter = 0;
83             frameIndex = (frameIndex + 1) % frameCounts[stateIndex];
84         }
85     }
86
87     public void draw(Graphics g) {
88         Image currentSheet = getCurrentImageSheet();
89         if (currentSheet == null) return;
90
91         int frameCount = frameCounts[currentState.ordinal()];
92         int frameW = currentSheet.getWidth(observer: null) / frameCount;
93         int sx1 = frameIndex * frameW;
94         int sy1 = 0;
```

```
src > J Player.java > ...
  8  public class Player {
  99    public void draw(Graphics g) {
106      int sy1 = 0;
107      int sx2 = sx1 + frameW;
108      int sy2 = currentSheet.getHeight(observer: null);
109
110      if (facingRight) {
111        g.drawImage(currentSheet, x, y, x + width, y + height, sx1, sy1, sx2, sy2, observer);
112      } else {
113        g.drawImage(currentSheet, x + width, y, x, y + height, sx1, sy1, sx2, sy2, observer);
114      }
115    }
116
117    private Image getCurrentImageSheet() {
118      switch (currentState) {
119        case IDLE: return idleSheet;
120        case RUN: return runSheet;
121        case JUMP: return jumpSheet;
122        case FALL: return fallSheet;
123        case HIT: return hitSheet;
124        default: return null;
125      }
126    }
127
128    public void moveRight() {
129      dx = 5;
130      facingRight = true;
131    }
132
133    public void moveLeft() {
134      dx = -5;
135      facingRight = false;
136    }
137
138    public void stop() {
139      dx = 0;
140    }
```

```

src > J Player.java > ...
8   public class Player {
138     public void stop() {
140   }
141
142     public void jump() {
143       if (onGround) {
144         dy = -15;
145         onGround = false;
146       }
147     }
148
149     public void bounce() {
150       dy = -10;
151     }
152
153     public void takeHit() {
154       if (hitTimer <= 0) { // Prevent taking hits while already in hit state
155         hitTimer = 30; // Hit animation duration
156       }
157     }
158
159     public boolean isOnGround() { return onGround; }
160     public Rectangle getBounds() { return new Rectangle(x, y, width, height); }
161     public int getDY() { return dy; }
162   }
163

```

Player.java pejelasan yang ada dalam sintaks file diatas

- AnimationState enum : IDLE, RUN, JUMP, FALL, HIT
- Variabel posisi & kecepatan : x, y, dx, dy, ukuran width/height
- Status : onGround, facingRight, currentState, hitTimer
- Gambar sprite : idleSheet, runSheet, jumpSheet, fallSheet, hitSheet
- Frame & animasi : frameCounts, animSpeeds, frameIndex, animCounter
- Constructor : set posisi awal, load sprite sheets
- loadSpriteSheets() : muat gambar sprite dari folder /assets/
- update(Level level) : update posisi, gravitasi, deteksi tanah, hit timer, state, dan animasi
- updateState() : tentukan state berdasarkan dx/dy, onGround, hitTimer
- updateAnimation() : pindah frame animasi sesuai kecepatan animasi
- draw(Graphics g) : gambar frame animasi saat ini, flip jika menghadap kiri
- getCurrentImageSheet() : ambil sprite sheet sesuai state
- Kontrol pemain
- moveRight() / moveLeft() : geser posisi dan arah hadap
- stop() : berhenti horizontal
- jump() : lompat kalau di tanah
- bounce() : lompat kecil (misal setelah musuh kena)
- takeHit() : masuk state HIT selama beberapa frame
- Getter : isOnGround(), getBounds(), getDY()

Classs GameFrame

Buka file di laptop



GameFrame.java

```
src > J GameFrame.java > ...
1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.event.*;
4 import java.io.InputStream;
5 import java.net.URL;
6 import java.util.ArrayList;
7 import java.util.Iterator;
8
9 /**
10  * The main panel for the game, handling the game loop, rendering, and input.
11 */
12 public class GameFrame extends JPanel implements ActionListener, KeyListener {
13
14     private Image background;
15     private Timer timer;
16     private Player player;
17     private Enemy enemy;
18     private Level level;
19     private Score score;
20
21     private Image heartImage;
22     private int lives = 3;
23
24     private Font gameFont;
25     private Font gameFontBold;
26
27     private Image frameImage;
28     private Image buttonImage;
29     private Rectangle buttonBounds;
30
31     private boolean isGameOver = false;
32     private boolean enemyRespawning = false;
33
34     private ArrayList<FloatingText> floatingTexts = new ArrayList<>();
35     private AudioManager audioManager;
36
37     public static final int WIDTH = 1920;
```

```
src > J GameFrame.java > ...
12 public class GameFrame extends JPanel implements ActionListener, KeyListener {
37     public static final int WIDTH = 1920;
38     public static final int HEIGHT = 1020;
39
40     public GameFrame() {
41         setLayout(mgr);
42
43         loadAssets();
44
45         audioManager = AudioManager.getInstance();
46         audioManager.loadEnemyHitSound(path: "/assets/enemy-hit.wav");
47         audioManager.loadJumpSound(path: "/assets/jump-se.wav");
48
49         level = new Level();
50         level.updateGroundY(HEIGHT);
51
52         int groundY = level.getGroundY();
53         player = new Player(x: 100, groundY - 60);
54         enemy = new Enemy(x: 600, groundY - 60);
55         score = new Score();
56
57         timer = new Timer(delay: 20, this);
58         timer.start();
59
60         addMouseListener(new MouseAdapter() {
61             @Override
62             public void mouseClicked(MouseEvent e) {
63                 if (isGameOver && buttonBounds != null && buttonBounds.contains(e.getPoint()))
64                     respawnGame();
65             }
66         });
67
68         setFocusable(focusable: true);
69         addKeyListener(this);
70     }
71
72 }
```

```
src > J GameFrame.java > ...
12  public class GameFrame extends JPanel implements ActionListener, KeyListener {
13
14      private void loadAssets() {
15          try {
16              URL bgUrl = getClass().getResource(name: "/assets/bg.png");
17              if (bgUrl != null) {
18                  background = new ImageIcon(bgUrl).getImage();
19              }
20
21              URL heartUrl = getClass().getResource(name: "/assets/hearth.png");
22              if (heartUrl != null) {
23                  heartImage = new ImageIcon(heartUrl).getImage();
24              }
25
26              URL frameUrl = getClass().getResource(name: "/assets/frame.png");
27              if (frameUrl != null) {
28                  frameImage = new ImageIcon(frameUrl).getImage();
29              }
30
31              URL buttonUrl = getClass().getResource(name: "/assets/button.png");
32              if (buttonUrl != null) {
33                  buttonImage = new ImageIcon(buttonUrl).getImage();
34              }
35
36              InputStream is = getClass().getResourceAsStream(name: "/assets/PixelifySans-Medium");
37              if (is != null) {
38                  Font baseFont = Font.createFont(Font.TRUETYPE_FONT, is);
39                  gameFont = baseFont.deriveFont(size: 18f);
40                  gameFontBold = baseFont.deriveFont(Font.BOLD, size: 32f);
41                  is.close();
42              } else {
43                  gameFont = new Font(name: "Arial", Font.PLAIN, size: 18);
44                  gameFontBold = new Font(name: "Arial", Font.BOLD, size: 32);
45              }
46          } catch (Exception e) {
47              System.err.println(x: "Failed to load one or more assets. Using fallback.");
48              e.printStackTrace();
49          }
50      }
51
52      @Override
53      public void actionPerformed(ActionEvent e) {
54          if (e.getSource() == buttonImage) {
55              if (gameFont == null) {
56                  gameFont = new Font(name: "Arial", Font.PLAIN, size: 18);
57                  gameFontBold = new Font(name: "Arial", Font.BOLD, size: 32);
58              }
59              if (gameFontBold.isBold()) {
60                  gameFontBold = new Font(name: "Arial", Font.PLAIN, size: 18);
61                  gameFont = new Font(name: "Arial", Font.BOLD, size: 32);
62              } else {
63                  gameFontBold = new Font(name: "Arial", Font.BOLD, size: 32);
64                  gameFont = new Font(name: "Arial", Font.PLAIN, size: 18);
65              }
66          }
67
68          repaint();
69      }
70
71      @Override
72      public void keyTyped(KeyEvent e) {
73          if (e.getKeyCode() == KeyEvent.VK_UP) {
74              if (gameFont != null) {
75                  gameFont = new Font(name: "Arial", Font.PLAIN, size: 18);
76                  gameFontBold = new Font(name: "Arial", Font.BOLD, size: 32);
77              }
78          } else if (e.getKeyCode() == KeyEvent.VK_DOWN) {
79              if (gameFontBold != null) {
80                  gameFontBold = new Font(name: "Arial", Font.PLAIN, size: 18);
81                  gameFont = new Font(name: "Arial", Font.BOLD, size: 32);
82              }
83          }
84
85          repaint();
86      }
87
88      @Override
89      public void keyReleased(KeyEvent e) {
90
91      }
92
93      @Override
94      public void keyPressed(KeyEvent e) {
95
96      }
97
98      @Override
99      protected void paintComponent(Graphics g) {
100         super.paintComponent(g);
101
102         g.drawImage(background, 0, 0, null);
103
104         g.setFont(gameFont);
105         g.drawString("Pixelify Sans", 100, 100);
106
107         g.setFont(gameFontBold);
108         g.drawString("Pixelify Sans", 100, 150);
109
110         g.drawImage(heartImage, 200, 200, null);
111
112         g.drawImage(frameImage, 300, 300, null);
113
114         g.drawImage(buttonImage, 400, 400, null);
115     }
116
117     @Override
118     public void mouseClicked(MouseEvent e) {
119
120     }
121
122     @Override
123     public void mouseEntered(MouseEvent e) {
124
125     }
126
127     @Override
128     public void mouseExited(MouseEvent e) {
129
130     }
131
132     @Override
133     public void mousePressed(MouseEvent e) {
134
135     }
136
137     @Override
138     public void mouseReleased(MouseEvent e) {
139
140     }
141
142     @Override
143     public void mouseDragged(MouseEvent e) {
144
145     }
146
147     @Override
148     public void mouseMoved(MouseEvent e) {
149
150     }
151
152     @Override
153     public void mouseWheelMoved(MouseWheelEvent e) {
154
155     }
156
157     @Override
158     public void mouseEntered(MouseEvent e) {
159
160     }
161
162     @Override
163     public void mouseExited(MouseEvent e) {
164
165     }
166
167     @Override
168     public void mousePressed(MouseEvent e) {
169
170     }
171
172     @Override
173     public void mouseReleased(MouseEvent e) {
174
175     }
176
177     @Override
178     public void mouseDragged(MouseEvent e) {
179
180     }
181
182     @Override
183     public void mouseMoved(MouseEvent e) {
184
185     }
186
187     @Override
188     public void mouseWheelMoved(MouseWheelEvent e) {
189
190     }
191
192     @Override
193     public void mouseEntered(MouseEvent e) {
194
195     }
196
197     @Override
198     public void mouseExited(MouseEvent e) {
199
200     }
201
202     @Override
203     public void mousePressed(MouseEvent e) {
204
205     }
206
207     @Override
208     public void mouseReleased(MouseEvent e) {
209
210     }
211
212     @Override
213     public void mouseDragged(MouseEvent e) {
214
215     }
216
217     @Override
218     public void mouseMoved(MouseEvent e) {
219
220     }
221
222     @Override
223     public void mouseWheelMoved(MouseWheelEvent e) {
224
225     }
226
227     @Override
228     public void mouseEntered(MouseEvent e) {
229
230     }
231
232     @Override
233     public void mouseExited(MouseEvent e) {
234
235     }
236
237     @Override
238     public void mousePressed(MouseEvent e) {
239
240     }
241
242     @Override
243     public void mouseReleased(MouseEvent e) {
244
245     }
246
247     @Override
248     public void mouseDragged(MouseEvent e) {
249
250     }
251
252     @Override
253     public void mouseMoved(MouseEvent e) {
254
255     }
256
257     @Override
258     public void mouseWheelMoved(MouseWheelEvent e) {
259
260     }
261
262     @Override
263     public void mouseEntered(MouseEvent e) {
264
265     }
266
267     @Override
268     public void mouseExited(MouseEvent e) {
269
270     }
271
272     @Override
273     public void mousePressed(MouseEvent e) {
274
275     }
276
277     @Override
278     public void mouseReleased(MouseEvent e) {
279
280     }
281
282     @Override
283     public void mouseDragged(MouseEvent e) {
284
285     }
286
287     @Override
288     public void mouseMoved(MouseEvent e) {
289
290     }
291
292     @Override
293     public void mouseWheelMoved(MouseWheelEvent e) {
294
295     }
296
297     @Override
298     public void mouseEntered(MouseEvent e) {
299
300     }
301
302     @Override
303     public void mouseExited(MouseEvent e) {
304
305     }
306
307     @Override
308     public void mousePressed(MouseEvent e) {
309
310     }
311
312     @Override
313     public void mouseReleased(MouseEvent e) {
314
315     }
316
317     @Override
318     public void mouseDragged(MouseEvent e) {
319
320     }
321
322     @Override
323     public void mouseMoved(MouseEvent e) {
324
325     }
326
327     @Override
328     public void mouseWheelMoved(MouseWheelEvent e) {
329
330     }
331
332     @Override
333     public void mouseEntered(MouseEvent e) {
334
335     }
336
337     @Override
338     public void mouseExited(MouseEvent e) {
339
340     }
341
342     @Override
343     public void mousePressed(MouseEvent e) {
344
345     }
346
347     @Override
348     public void mouseReleased(MouseEvent e) {
349
350     }
351
352     @Override
353     public void mouseDragged(MouseEvent e) {
354
355     }
356
357     @Override
358     public void mouseMoved(MouseEvent e) {
359
360     }
361
362     @Override
363     public void mouseWheelMoved(MouseWheelEvent e) {
364
365     }
366
367     @Override
368     public void mouseEntered(MouseEvent e) {
369
370     }
371
372     @Override
373     public void mouseExited(MouseEvent e) {
374
375     }
376
377     @Override
378     public void mousePressed(MouseEvent e) {
379
380     }
381
382     @Override
383     public void mouseReleased(MouseEvent e) {
384
385     }
386
387     @Override
388     public void mouseDragged(MouseEvent e) {
389
390     }
391
392     @Override
393     public void mouseMoved(MouseEvent e) {
394
395     }
396
397     @Override
398     public void mouseWheelMoved(MouseWheelEvent e) {
399
400     }
401
402     @Override
403     public void mouseEntered(MouseEvent e) {
404
405     }
406
407     @Override
408     public void mouseExited(MouseEvent e) {
409
410     }
411
412     @Override
413     public void mousePressed(MouseEvent e) {
414
415     }
416
417     @Override
418     public void mouseReleased(MouseEvent e) {
419
420     }
421
422     @Override
423     public void mouseDragged(MouseEvent e) {
424
425     }
426
427     @Override
428     public void mouseMoved(MouseEvent e) {
429
430     }
431
432     @Override
433     public void mouseWheelMoved(MouseWheelEvent e) {
434
435     }
436
437     @Override
438     public void mouseEntered(MouseEvent e) {
439
440     }
441
442     @Override
443     public void mouseExited(MouseEvent e) {
444
445     }
446
447     @Override
448     public void mousePressed(MouseEvent e) {
449
450     }
451
452     @Override
453     public void mouseReleased(MouseEvent e) {
454
455     }
456
457     @Override
458     public void mouseDragged(MouseEvent e) {
459
460     }
461
462     @Override
463     public void mouseMoved(MouseEvent e) {
464
465     }
466
467     @Override
468     public void mouseWheelMoved(MouseWheelEvent e) {
469
470     }
471
472     @Override
473     public void mouseEntered(MouseEvent e) {
474
475     }
476
477     @Override
478     public void mouseExited(MouseEvent e) {
479
480     }
481
482     @Override
483     public void mousePressed(MouseEvent e) {
484
485     }
486
487     @Override
488     public void mouseReleased(MouseEvent e) {
489
490     }
491
492     @Override
493     public void mouseDragged(MouseEvent e) {
494
495     }
496
497     @Override
498     public void mouseMoved(MouseEvent e) {
499
500     }
501
502     @Override
503     public void mouseWheelMoved(MouseWheelEvent e) {
504
505     }
506
507     @Override
508     public void mouseEntered(MouseEvent e) {
509
510     }
511
512     @Override
513     public void mouseExited(MouseEvent e) {
514
515     }
516
517     @Override
518     public void mousePressed(MouseEvent e) {
519
520     }
521
522     @Override
523     public void mouseReleased(MouseEvent e) {
524
525     }
526
527     @Override
528     public void mouseDragged(MouseEvent e) {
529
530     }
531
532     @Override
533     public void mouseMoved(MouseEvent e) {
534
535     }
536
537     @Override
538     public void mouseWheelMoved(MouseWheelEvent e) {
539
540     }
541
542     @Override
543     public void mouseEntered(MouseEvent e) {
544
545     }
546
547     @Override
548     public void mouseExited(MouseEvent e) {
549
550     }
551
552     @Override
553     public void mousePressed(MouseEvent e) {
554
555     }
556
557     @Override
558     public void mouseReleased(MouseEvent e) {
559
560     }
561
562     @Override
563     public void mouseDragged(MouseEvent e) {
564
565     }
566
567     @Override
568     public void mouseMoved(MouseEvent e) {
569
570     }
571
572     @Override
573     public void mouseWheelMoved(MouseWheelEvent e) {
574
575     }
576
577     @Override
578     public void mouseEntered(MouseEvent e) {
579
580     }
581
582     @Override
583     public void mouseExited(MouseEvent e) {
584
585     }
586
587     @Override
588     public void mousePressed(MouseEvent e) {
589
590     }
591
592     @Override
593     public void mouseReleased(MouseEvent e) {
594
595     }
596
597     @Override
598     public void mouseDragged(MouseEvent e) {
599
600     }
601
602     @Override
603     public void mouseMoved(MouseEvent e) {
604
605     }
606
607     @Override
608     public void mouseWheelMoved(MouseWheelEvent e) {
609
610     }
611
612     @Override
613     public void mouseEntered(MouseEvent e) {
614
615     }
616
617     @Override
618     public void mouseExited(MouseEvent e) {
619
620     }
621
622     @Override
623     public void mousePressed(MouseEvent e) {
624
625     }
626
627     @Override
628     public void mouseReleased(MouseEvent e) {
629
630     }
631
632     @Override
633     public void mouseDragged(MouseEvent e) {
634
635     }
636
637     @Override
638     public void mouseMoved(MouseEvent e) {
639
640     }
641
642     @Override
643     public void mouseWheelMoved(MouseWheelEvent e) {
644
645     }
646
647     @Override
648     public void mouseEntered(MouseEvent e) {
649
650     }
651
652     @Override
653     public void mouseExited(MouseEvent e) {
654
655     }
656
657     @Override
658     public void mousePressed(MouseEvent e) {
659
660     }
661
662     @Override
663     public void mouseReleased(MouseEvent e) {
664
665     }
666
667     @Override
668     public void mouseDragged(MouseEvent e) {
669
670     }
671
672     @Override
673     public void mouseMoved(MouseEvent e) {
674
675     }
676
677     @Override
678     public void mouseWheelMoved(MouseWheelEvent e) {
679
680     }
681
682     @Override
683     public void mouseEntered(MouseEvent e) {
684
685     }
686
687     @Override
688     public void mouseExited(MouseEvent e) {
689
690     }
691
692     @Override
693     public void mousePressed(MouseEvent e) {
694
695     }
696
697     @Override
698     public void mouseReleased(MouseEvent e) {
699
700     }
701
702     @Override
703     public void mouseDragged(MouseEvent e) {
704
705     }
706
707     @Override
708     public void mouseMoved(MouseEvent e) {
709
710     }
711
712     @Override
713     public void mouseWheelMoved(MouseWheelEvent e) {
714
715     }
716
717     @Override
718     public void mouseEntered(MouseEvent e) {
719
720     }
721
722     @Override
723     public void mouseExited(MouseEvent e) {
724
725     }
726
727     @Override
728     public void mousePressed(MouseEvent e) {
729
730     }
731
732     @Override
733     public void mouseReleased(MouseEvent e) {
734
735     }
736
737     @Override
738     public void mouseDragged(MouseEvent e) {
739
740     }
741
742     @Override
743     public void mouseMoved(MouseEvent e) {
744
745     }
746
747     @Override
748     public void mouseWheelMoved(MouseWheelEvent e) {
749
750     }
751
752     @Override
753     public void mouseEntered(MouseEvent e) {
754
755     }
756
757     @Override
758     public void mouseExited(MouseEvent e) {
759
760     }
761
762     @Override
763     public void mousePressed(MouseEvent e) {
764
765     }
766
767     @Override
768     public void mouseReleased(MouseEvent e) {
769
770     }
771
772     @Override
773     public void mouseDragged(MouseEvent e) {
774
775     }
776
777     @Override
778     public void mouseMoved(MouseEvent e) {
779
780     }
781
782     @Override
783     public void mouseWheelMoved(MouseWheelEvent e) {
784
785     }
786
787     @Override
788     public void mouseEntered(MouseEvent e) {
789
790     }
791
792     @Override
793     public void mouseExited(MouseEvent e) {
794
795     }
796
797     @Override
798     public void mousePressed(MouseEvent e) {
799
800     }
801
802     @Override
803     public void mouseReleased(MouseEvent e) {
804
805     }
806
807     @Override
808     public void mouseDragged(MouseEvent e) {
809
810     }
811
812     @Override
813     public void mouseMoved(MouseEvent e) {
814
815     }
816
817     @Override
818     public void mouseWheelMoved(MouseWheelEvent e) {
819
820     }
821
822     @Override
823     public void mouseEntered(MouseEvent e) {
824
825     }
826
827     @Override
828     public void mouseExited(MouseEvent e) {
829
830     }
831
832     @Override
833     public void mousePressed(MouseEvent e) {
834
835     }
836
837     @Override
838     public void mouseReleased(MouseEvent e) {
839
840     }
841
842     @Override
843     public void mouseDragged(MouseEvent e) {
844
845     }
846
847     @Override
848     public void mouseMoved(MouseEvent e) {
849
850     }
851
852     @Override
853     public void mouseWheelMoved(MouseWheelEvent e) {
854
855     }
856
857     @Override
858     public void mouseEntered(MouseEvent e) {
859
860     }
861
862     @Override
863     public void mouseExited(MouseEvent e) {
864
865     }
866
867     @Override
868     public void mousePressed(MouseEvent e) {
869
870     }
871
872     @Override
873     public void mouseReleased(MouseEvent e) {
874
875     }
876
877     @Override
878     public void mouseDragged(MouseEvent e) {
879
880     }
881
882     @Override
883     public void mouseMoved(MouseEvent e) {
884
885     }
886
887     @Override
888     public void mouseWheelMoved(MouseWheelEvent e) {
889
890     }
891
892     @Override
893     public void mouseEntered(MouseEvent e) {
894
895     }
896
897     @Override
898     public void mouseExited(MouseEvent e) {
899
900     }
901
902     @Override
903     public void mousePressed(MouseEvent e) {
904
905     }
906
907     @Override
908     public void mouseReleased(MouseEvent e) {
909
910     }
911
912     @Override
913     public void mouseDragged(MouseEvent e) {
914
915     }
916
917     @Override
918     public void mouseMoved(MouseEvent e) {
919
920     }
921
922     @Override
923     public void mouseWheelMoved(MouseWheelEvent e) {
924
925     }
926
927     @Override
928     public void mouseEntered(MouseEvent e) {
929
930     }
931
932     @Override
933     public void mouseExited(MouseEvent e) {
934
935     }
936
937     @Override
938     public void mousePressed(MouseEvent e) {
939
940     }
941
942     @Override
943     public void mouseReleased(MouseEvent e) {
944
945     }
946
947     @Override
948     public void mouseDragged(MouseEvent e) {
949
950     }
951
952     @Override
953     public void mouseMoved(MouseEvent e) {
954
955     }
956
957     @Override
958     public void mouseWheelMoved(MouseWheelEvent e) {
959
960     }
961
962     @Override
963     public void mouseEntered(MouseEvent e) {
964
965     }
966
967     @Override
968     public void mouseExited(MouseEvent e) {
969
970     }
971
972     @Override
973     public void mousePressed(MouseEvent e) {
974
975     }
976
977     @Override
978     public void mouseReleased(MouseEvent e) {
979
980     }
981
982     @Override
983     public void mouseDragged(MouseEvent e) {
984
985     }
986
987     @Override
988     public void mouseMoved(MouseEvent e) {
989
990     }
991
992     @Override
993     public void mouseWheelMoved(MouseWheelEvent e) {
994
995     }
996
997     @Override
998     public void mouseEntered(MouseEvent e) {
999
1000    }
1001
1002     @Override
1003     public void mouseExited(MouseEvent e) {
1004
1005    }
1006
1007     @Override
1008     public void mousePressed(MouseEvent e) {
1009
1010    }
1011
1012     @Override
1013     public void mouseReleased(MouseEvent e) {
1014
1015    }
1016
1017     @Override
1018     public void mouseDragged(MouseEvent e) {
1019
1020    }
1021
1022     @Override
1023     public void mouseMoved(MouseEvent e) {
1024
1025    }
1026
1027     @Override
1028     public void mouseWheelMoved(MouseWheelEvent e) {
1029
1030    }
1031
1032     @Override
1033     public void mouseEntered(MouseEvent e) {
1034
1035    }
1036
1037     @Override
1038     public void mouseExited(MouseEvent e) {
1039
1040    }
1041
1042     @Override
1043     public void mousePressed(MouseEvent e) {
1044
1045    }
1046
1047     @Override
1048     public void mouseReleased(MouseEvent e) {
1049
1050    }
1051
1052     @Override
1053     public void mouseDragged(MouseEvent e) {
1054
1055    }
1056
1057     @Override
1058     public void mouseMoved(MouseEvent e) {
1059
1060    }
1061
1062     @Override
1063     public void mouseWheelMoved(MouseWheelEvent e) {
1064
1065    }
1066
1067     @Override
1068     public void mouseEntered(MouseEvent e) {
1069
1070    }
1071
1072     @Override
1073     public void mouseExited(MouseEvent e) {
1074
1075    }
1076
1077     @Override
1078     public void mousePressed(MouseEvent e) {
1079
1080    }
1081
1082     @Override
1083     public void mouseReleased(MouseEvent e) {
1084
1085    }
1086
1087     @Override
1088     public void mouseDragged(MouseEvent e) {
1089
1090    }
1091
1092     @Override
1093     public void mouseMoved(MouseEvent e) {
1094
1095    }
1096
1097     @Override
1098     public void mouseWheelMoved(MouseWheelEvent e) {
1099
1100    }
1101
1102     @Override
1103     public void mouseEntered(MouseEvent e) {
1104
1105    }
1106
1107     @Override
1108     public void mouseExited(MouseEvent e) {
1109
1110    }
1111
1112     @Override
1113     public void mousePressed(MouseEvent e) {
1114
1115    }
1116
1117     @Override
1118     public void mouseReleased(MouseEvent e) {
1119
1120    }
1121
1122     @Override
1123     public void mouseDragged(MouseEvent e) {
1124
1125    }
1126
1127     @Override
1128     public void mouseMoved(MouseEvent e) {
1129
1130    }
1131
1132     @Override
1133     public void mouseWheelMoved(MouseWheelEvent e) {
1134
1135    }
1136
1137     @Override
1138     public void mouseEntered(MouseEvent e) {
1139
1140    }
1141
1142     @Override
1143     public void mouseExited(MouseEvent e) {
1144
1145    }
1146
1147     @Override
1148     public void mousePressed(MouseEvent e) {
1149
1150    }
1151
1152     @Override
1153     public void mouseReleased(MouseEvent e) {
1154
1155    }
1156
1157     @Override
1158     public void mouseDragged(MouseEvent e) {
1159
1160    }
1161
1162     @Override
1163     public void mouseMoved(MouseEvent e) {
1164
1165    }
1166
1167     @Override
1168     public void mouseWheelMoved(MouseWheelEvent e) {
1169
1170    }
1171
1172     @Override
1173     public void mouseEntered(MouseEvent e) {
1174
1175    }
1176
1177     @Override
1178     public void mouseExited(MouseEvent e) {
1179
1180    }
1181
1182     @Override
1183     public void mousePressed(MouseEvent e) {
1184
1185    }
1186
1187     @Override
1188     public void mouseReleased(MouseEvent e) {
1189
1190    }
1191
1192     @Override
1193     public void mouseDragged(MouseEvent e) {
1194
1195    }
1196
1197     @Override
1198     public void mouseMoved(MouseEvent e) {
1199
1200    }
1201
1202     @Override
1203     public void mouseWheelMoved(MouseWheelEvent e) {
1204
1205    }
1206
1207     @Override
1208     public void mouseEntered(MouseEvent e) {
1209
1210    }
1211
1212     @Override
1213     public void mouseExited(MouseEvent e) {
1214
1215    }
1216
1217     @Override
1218     public void mousePressed(MouseEvent e) {
1219
1220    }
1221
1222     @Override
1223     public void mouseReleased(MouseEvent e) {
1224
1225    }
1226
1227     @Override
1228     public void mouseDragged(MouseEvent e) {
1229
1230    }
1231
1232     @Override
1233     public void mouseMoved(MouseEvent e) {
1234
1235    }
1236
1237     @Override
1238     public void mouseWheelMoved(MouseWheelEvent e) {
1239
1240    }
1241
1242     @Override
1243     public void mouseEntered(MouseEvent e) {
1244
1245    }
1246
1247     @Override
1248     public void mouseExited(MouseEvent e) {
1249
1250    }
1251
1252     @Override
1253     public void mousePressed(MouseEvent e) {
1254
1255    }
1256
1257     @Override
1258     public void mouseReleased(MouseEvent e) {
1259
1260    }
1261
1262     @Override
1263     public void mouseDragged(MouseEvent e) {
1264
1265    }
1266
1267     @Override
1268     public void mouseMoved(MouseEvent e) {
1269
1270    }
1271
1272     @Override
1273     public void mouseWheelMoved(MouseWheelEvent e) {
1274
1275    }
1276
1277     @Override
1278     public void mouseEntered(MouseEvent e) {
1279
1280    }
1281
1282     @Override
1283     public void mouseExited(MouseEvent e) {
1284
1285    }
1286
1287     @Override
1288     public void mousePressed(MouseEvent e) {
1289
1290    }
1291
1292     @Override
1293     public void mouseReleased(MouseEvent e) {
1294
1295    }
1296
1297     @Override
1298     public void mouseDragged(MouseEvent e) {
1299
1300    }
1301
1302     @Override
1303     public void mouseMoved(MouseEvent e) {
1304
1305    }
1306
1307     @Override
1308     public void mouseWheelMoved(MouseWheelEvent e) {
1309
1310    }
1311
1312     @Override
1313     public void mouseEntered(MouseEvent e) {
1314
1315    }
1316
1317     @Override
1318     public void mouseExited(MouseEvent e) {
1319
1320    }
1321
1322     @Override
1323     public void mousePressed(MouseEvent e) {
1324
1325    }
1326
1327     @Override
1328     public void mouseReleased(MouseEvent e) {
1329
1330    }
1331
1332     @Override
1333     public void mouseDragged(MouseEvent e) {
1334
1335    }
1336
1337     @Override
1338     public void mouseMoved(MouseEvent e) {
1339
1340    }
1341
1342     @Override
1343     public void mouseWheelMoved(MouseWheelEvent e) {
1344
1345    }
1346
1347     @Override
1348     public void mouseEntered(MouseEvent e) {
1349
1350    }
1351
1352     @Override
1353     public void mouseExited(MouseEvent e) {
1354
1355    }
1356
1357     @Override
1358     public void mousePressed(MouseEvent e) {
1359
1360    }
1361
1362     @Override
1363     public void mouseReleased(MouseEvent e) {
1364
1365    }
1366
1367     @Override
1368     public void mouseDragged(MouseEvent e) {
1369
1370    }
1371
1372     @Override
1373     public void mouseMoved(MouseEvent e) {
1374
1375    }
1376
1377     @Override
1378     public void mouseWheelMoved(MouseWheelEvent e) {
1379
1380    }
1381
1382     @Override
1383     public void mouseEntered(MouseEvent e) {
1384
1385    }
1386
1387     @Override
1388     public void mouseExited(MouseEvent e) {
1389
1390    }
1391
1392     @Override
1393     public void mousePressed(MouseEvent e) {
1394
1395    }
1396
1397     @Override
1398     public void mouseReleased(MouseEvent e) {
1399
1400    }
1401
1402     @Override
1403     public void mouseDragged(MouseEvent e) {
1404
1405    }
1406
1407     @Override
1408     public void mouseMoved(MouseEvent e) {
1409
1410    }
1411
1412     @Override
1413     public void mouseWheelMoved(MouseWheelEvent e) {
1414
1415    }
1416
1417     @Override
1418     public void mouseEntered(MouseEvent e) {
1419
1420    }
1421
1422     @Override
1423     public void mouseExited(MouseEvent e) {
1424
1425    }
1426
1427     @Override
1428     public void mousePressed(MouseEvent e) {
1429
1430    }
1431
1432     @Override
1433     public void mouseReleased(MouseEvent e) {
1434
1435    }
1436
1437     @Override
1438     public void mouseDragged(MouseEvent e) {
1439
1440    }
1441
1442     @Override
1443     public void mouseMoved(MouseEvent e) {
1444
1445    }
1446
1447     @Override
1448     public void mouseWheelMoved(MouseWheelEvent e) {
1449
1450    }
1451
1452     @Override
1453     public void mouseEntered(MouseEvent e) {
1454
1455    }
1456
1457     @Override
1458     public void mouseExited(MouseEvent e) {
1459
1460    }
1461
1462     @Override
1463     public void mousePressed(MouseEvent e) {
1464
1465    }
1466
1467     @Override
1468     public void mouseReleased(MouseEvent e) {
1469
1470    }
1471
1472     @Override
1473     public void mouseDragged(MouseEvent e) {
1474
1475    }
1476
1477     @Override
1478     public void mouseMoved(MouseEvent e) {
1479
1480    }
1481
1482     @Override
1483     public void mouseWheelMoved(MouseWheelEvent e) {
1484
1485    }
1486
1487     @Override
1488     public void mouseEntered(MouseEvent e) {
1489
1490    }
1491
1492     @Override
1493     public void mouseExited(MouseEvent e) {
1494
1495    }
1496
1497     @Override
1498     public void mousePressed(MouseEvent e) {
1499
1500    }
1501
1502     @Override
1503     public void mouseReleased(MouseEvent e) {
1504
1505    }
1506
1507     @Override
1508     public void mouseDragged(MouseEvent e) {
1509
1510    }
1511
1512     @Override
1513     public void mouseMoved(MouseEvent e) {
1514
1515    }
1516
1517     @Override
1518     public void mouseWheelMoved(MouseWheelEvent e) {
1519
1520    }
1521
1522     @Override
1523     public void mouseEntered(MouseEvent e) {
1524
1525    }
1526
1527     @Override
1528     public void mouseExited(MouseEvent e) {
1529
1530    }
1531
1532     @Override
1533     public void mousePressed(MouseEvent e) {
1534
1535    }
1536
1537     @Override
1538     public void mouseReleased(MouseEvent e) {
1539
1540    }
1541
1542     @Override
1543     public void mouseDragged(MouseEvent e) {
1544
1545    }
1546
1547     @Override
1548     public void mouseMoved(MouseEvent e) {
1549
1550    }
1551
1552     @Override
1553     public void mouseWheelMoved(MouseWheelEvent e) {
1554
1555    }
1556
1557     @Override
1558     public void mouseEntered(MouseEvent e) {
1559
1560    }
1561
1562     @Override
1563     public void mouseExited(MouseEvent e) {
1564
1565    }
1566
1567     @Override
1568     public void mousePressed(MouseEvent e) {
1569
1570    }
1571
1572     @Override
1573     public void mouseReleased(MouseEvent e) {
1574
1575    }
1576
1577     @Override
1578     public void mouseDragged(MouseEvent e) {
1579
1580    }
1581
1582     @Override
1583     public void mouseMoved(MouseEvent e) {
1584
1585    }
1586
1587     @Override
1588     public void mouseWheelMoved(MouseWheelEvent e) {
1589
1590    }
1591
1592     @Override
1593     public void mouseEntered(MouseEvent e) {
1594
1595    }
1596
1597     @Override
1598     public void mouseExited(MouseEvent e) {
1599
1600    }
1601
1602     @Override
1603     public void mousePressed(MouseEvent e) {
1604
1605    }
1606
1607     @Override
1608     public void mouseReleased(MouseEvent e) {
1609
1610    }
1611
1612     @Override
1613     public void mouseDragged(MouseEvent e) {
1614
1615    }
1616
1617     @Override
1618     public void mouseMoved(MouseEvent e) {
1619
1620    }
1621
1622     @Override
1623     public void mouseWheelMoved(MouseWheelEvent e) {
1624
1625    }
1626
1627     @Override
1628     public void mouseEntered(MouseEvent e) {
1629
1630    }
1631
1632     @Override
1633     public void mouseExited(MouseEvent e) {
1634
1635    }
1636
1637     @Override
1638     public void mousePressed(MouseEvent e) {
1639
1640    }
1641
1642     @Override
1643     public void mouseReleased(MouseEvent e) {
1644
1645    }
1646
1647     @Override
1648     public void mouseDragged(MouseEvent e) {
1649
1650    }
1651
1652     @Override
1653     public void mouseMoved(MouseEvent e) {
1654
1655    }
1656
1657     @Override
1658     public void mouseWheelMoved(MouseWheelEvent e) {
1659
1660    }
1661
1662     @Override
1663     public void mouseEntered(MouseEvent e) {
1664
1665    }
1666
1667     @Override
1668     public void mouseExited(MouseEvent e) {
1669
1670    }
1671
1672     @Override
1673     public void mousePressed(MouseEvent e) {
1674
1675    }
1676
1677     @Override
1678     public void mouseReleased(MouseEvent e) {
1679
1680    }
1681
1682     @Override
1683     public void mouseDragged(MouseEvent e) {
1684
1685    }
1686
1687     @Override
1688     public void mouseMoved(MouseEvent e) {
1689
1690    }
1691
1692     @Override
1693     public void mouseWheelMoved(MouseWheelEvent e) {
1694
1695    }
1696
1697     @Override
1698     public void mouseEntered(MouseEvent e) {
1699
1700    }
1701
1702     @Override
1703     public void mouseExited(MouseEvent e) {
1704
1705    }
1706
1707     @Override
1708     public void mousePressed(MouseEvent e) {
1709
1710    }
1711
1712     @Override
1713     public void mouseReleased(MouseEvent e) {
1714
1715    }
1716
1717     @Override
1718     public void mouseDragged(MouseEvent e) {
1719
1720    }
1721
1722     @Override
1723     public void mouseMoved(MouseEvent e) {
1724
1725    }
1726
1727     @Override
1728     public void mouseWheelMoved(MouseWheelEvent e) {
1729
1730    }
1731
1732     @Override

```

```
src > J GameFrame.java > ...
12  public class GameFrame extends JPanel implements ActionListener, KeyListener {
73      private void loadAssets() {
108     // Use default font as a fallback
109     gameFont = new Font(name: "Arial", Font.PLAIN, size: 18);
110     gameFontBold = new Font(name: "Arial", Font.BOLD, size: 32);
111 }
112 }
113
114 @Override
115 protected void paintComponent(Graphics g) {
116     super.paintComponent(g);
117
118     if (background != null) {
119         g.drawImage(background, x: 0, y: 0, getWidth(), getHeight(), this);
120     } else {
121         g.setColor(Color.CYAN);
122         g.fillRect(x: 0, y: 0, getWidth(), getHeight());
123     }
124
125     level.draw(g, getWidth());
126     player.draw(g);
127     enemy.draw(g);
128
129     for (FloatingText text : floatingTexts) {
130         text.draw(g);
131     }
132
133     g.setColor(Color.BLACK);
134     g.setFont(gameFont);
135     g.drawString("Score: " + score.getScore(), x: 20, y: 30);
136
137     if (heartImage != null) {
138         int heartSize = 32;
139         for (int i = 0; i < lives; i++) {
140             g.drawImage(heartImage, 20 + i * (heartSize + 8), y: 40, heartSize, heartSize,
141         }
142     }
}
```

```
src > J GameFrame.java > ...
12  public class GameFrame extends JPanel implements ActionListener, KeyListener {
115     protected void paintComponent(Graphics g) {
142     }
143
144     if (isGameOver) {
145         drawGameOver((Graphics2D) g);
146     }
147 }
148
149 private void drawGameOver(Graphics2D g2) {
150     int panelWidth = getWidth();
151     int panelHeight = getHeight();
152
153     int frameWidth = 650;
154     int frameHeight = 550;
155     int frameX = (panelWidth - frameWidth) / 2;
156     int frameY = (panelHeight - frameHeight) / 2;
157
158     if (frameImage != null) {
159         g2.drawImage(frameImage, frameX, frameY, frameWidth, frameHeight, observer: null);
160     }
161
162     g2.setFont(gameFontBold.deriveFont(size: 56f));
163     FontMetrics fmGameOver = g2.getFontMetrics();
164     String gameOverText = "GAME OVER";
165     int gameOverX = frameX + (frameWidth - fmGameOver.stringWidth(gameOverText)) / 2;
166     int gameOverY = frameY + 100;
167
168     g2.setColor(Color.BLACK);
169     g2.drawString(gameOverText, gameOverX + 2, gameOverY + 2);
170     g2.setColor(Color.RED);
171     g2.drawString(gameOverText, gameOverX, gameOverY);
172
173     g2.setFont(gameFont.deriveFont(size: 28f));
174     FontMetrics fmScore = g2.getFontMetrics();
175     String scoreText = "Score: " + score.getScore();
176     int scoreX = frameX + (frameWidth - fmScore.stringWidth(scoreText)) / 2;
```

```
src > GameFrame.java > ...
12  public class GameFrame extends JPanel implements ActionListener, KeyListener {
149     private void drawGameOver(Graphics2D g2) {
177         int scoreY = gameOverY + 100;
178
179         g2.setColor(Color.BLACK);
180         g2.drawString(scoreText, scoreX + 2, scoreY + 2);
181         g2.setColor(Color.WHITE);
182         g2.drawString(scoreText, scoreX, scoreY);
183
184         if (buttonImage != null) {
185             int buttonWidth = 180;
186             int buttonHeight = 70;
187             int buttonX = frameX + (frameWidth - buttonWidth) / 2;
188             int buttonY = scoreY + 70;
189
190             buttonBounds = new Rectangle(buttonX, buttonY, buttonWidth, buttonHeight);
191             g2.drawImage(buttonImage, buttonX, buttonY, buttonWidth, buttonHeight, observer: null);
192         }
193     }
194
195     @Override
196     public void actionPerformed(ActionEvent e) {
197         if (isGameOver) return;
198
199         level.updateGroundY(getHeight());
200         player.update(level);
201         enemy.update(level);
202
203         Iterator<FloatingText> textIterator = floatingTexts.iterator();
204         while (textIterator.hasNext()) {
205             FloatingText text = textIterator.next();
206             text.update();
207             if (!text.isAlive()) {
208                 textIterator.remove();
209             }
210         }
211     }
212 }
```

```
src > GameFrame.java > ...
12  public class GameFrame extends JPanel implements ActionListener, KeyListener {
196     public void actionPerformed(ActionEvent e) {
211         checkCollisions();
212         repaint();
213     }
214 }
215
216     private void checkCollisions() {
217         if (!enemy.isAlive() || !player.getBounds().intersects(enemy.getBounds())) {
218             return;
219         }
220
221         if (player.getDY() > 0 && player.getBounds().y + player.getBounds().height < enemy.getBounds().y) {
222             enemy	die();
223             score.addScore(value: 100);
224             score.addKill();
225             player.bounce();
226             audioManager.playEnemyHit();
227
228             String killText = "+" + score.getKillCount() + " Kill";
229             floatingTexts.add(new FloatingText(killText, enemy.getBounds().x, enemy.getBounds().y));
230
231             if (!enemyRespawning) {
232                 enemyRespawning = true;
233                 Timer respawnTimer = new Timer(delay: 1000, ev -> {
234                     enemy.respawn(level);
235                     enemyRespawning = false;
236                 });
237                 respawnTimer.setRepeats(flag: false);
238                 respawnTimer.start();
239             }
240         } else {
241             player.takeHit();
242             loseLife();
243         }
244     }
245 }
```

```
src > J GameFrame.java > ...
12  public class GameFrame extends JPanel implements ActionListener, KeyListener {
245
246      private void loseLife() {
247          lives--;
248          if (lives <= 0) {
249              lives = 0;
250              gameOver();
251          } else {
252              enemy.respawn(level);
253          }
254      }
255
256      private void gameOver() {
257          isGameOver = true;
258          timer.stop();
259      }
260
261      private void respawnGame() {
262          lives = 3;
263          isGameOver = false;
264
265          int groundY = level.getGroundY();
266          player = new Player(x: 100, groundY - 60);
267          enemy.respawn(level);
268          score.reset();
269          floatingTexts.clear();
270
271          enemyRespawning = false;
272          buttonBounds = null;
273          timer.start();
274      }
275
276      @Override
277      public void keyPressed(KeyEvent e) {
278          int code = e.getKeyCode();
279
280          if (code == KeyEvent.VK_RIGHT || code == KeyEvent.VK_D) {
```

```

src > J GameFrame.java > ...
12  public class GameFrame extends JPanel implements ActionListener, KeyListener {
277      public void keyPressed(KeyEvent e) {
280          if (code == KeyEvent.VK_RIGHT || code == KeyEvent.VK_D) {
281              player.moveRight();
282          } else if (code == KeyEvent.VK_LEFT || code == KeyEvent.VK_A) {
283              player.moveLeft();
284          } else if (code == KeyEvent.VK_SPACE || code == KeyEvent.VK_W || code == KeyEvent.VK_U) {
285              if (player.isOnGround()) {
286                  player.jump();
287                  audioManager.playJump();
288              }
289          }
290      }
291
292      @Override
293      public void keyReleased(KeyEvent e) {
294          int code = e.getKeyCode();
295          if (code == KeyEvent.VK_RIGHT || code == KeyEvent.VK_D ||
296              code == KeyEvent.VK_LEFT || code == KeyEvent.VK_A) {
297              player.stop();
298          }
299      }
300
301      @Override
302      public void keyTyped(KeyEvent e) {}
303
304      Run | Debug
305      public static void main(String[] args) {
306          SwingUtilities.invokeLater(() -> {
307              JFrame frame = new JFrame("Mini Mario Bros");
308              GameFrame game = new GameFrame();
309              frame.add(game);
310              frame.setSize(WIDTH, HEIGHT);
311              frame.setLocationRelativeTo(null);
312              frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
313              frame.setResizable(resizable: true);
314              frame.setVisible(b: true);
315          });
316      }
317  }

```

Inheritance & Interfaces : extends JPanel, implements ActionListener, KeyListener

Variabel game :

- background, frameImage, buttonImage, heartImage : gambar & UI
- Player player, Enemy enemy, Level level, Score score : objek game
- lives : nyawa pemain
- floatingTexts : teks efek (+Kill, dll.)

- AudioManager audioManager : play sound efek
- timer : game loop
- isGameOver, enemyRespawning : status game

Ukuran game : WIDTH=1920, HEIGHT=1020

Constructor GameFrame()

- Load assets (gambar, font)
- Inisialisasi AudioManager dan load suara
- Inisialisasi level, player, enemy, score
- Setup timer loop (20 ms)
- Mouse listener : klik tombol respawn saat game over
- Key listener : kontrol player

loadAssets() : muat semua gambar & font, fallback pakai default kalau gagal

paintComponent(Graphics g) :

- Gambar background, level, player, enemy
- Gambar floatingTexts
- Gambar score & nyawa (hearts)
- Jika isGameOver : panggil drawGameOver()

drawGameOver(Graphics2D g2):

Gambar frame game over, teks "GAME OVER", skor, dan tombol respawn

actionPerformed(ActionEvent e) :

- Update level, player, enemy
- Update floatingTexts dan hapus yang mati
- Cek collision
- Repaint panel

checkCollisions() :

- Jika player menabrak enemy dari atas : kill enemy, add score, bounce player, play sound, floatingText + respawn enemy
- Jika tidak : player takeHit, kurangi nyawa (loseLife())

loseLife() :

- lives--
- Jika 0 : gameOver()
- Jika masih ada nyawa : respawn enemy

gameOver() : stop timer, set isGameOver = true

respawnGame() : reset lives, reset player/enemy/score/floatingTexts, start timer

Key controls :

- moveRight() / moveLeft() : geser player
- stop() : berhenti horizontal
- jump() : lompat, play sound jump

Main method :

- Buat JFrame, tambahkan GameFrame, set ukuran & visible

Main method :

- Buat JFrame, tambahkan GameFrame, set ukuran & visible

Class level

Buka link dari laptop



Level.java

```
src > J Level.java > ...
1  import javax.swing.*;
2  import java.awt.*;
3  import java.net.URL;
4
5  /**
6   * Manages the game's ground level.
7   */
8  public class Level {
9      private int groundSurfaceY;
10     private final int groundHeight = 250;
11     private final int surfaceThickness = 85;
12     private Image groundImage;
13
14     public Level() {
15         try {
16             URL groundUrl = getClass().getResource(name: "/assets/Ground.png");
17             if (groundUrl != null) {
18                 groundImage = new ImageIcon(groundUrl).getImage();
19             }
20         } catch (Exception e) {
21             System.err.println(x: "Failed to load ground image.");
22             groundImage = null;
23         }
24     }
25
26     /**
27      * Draws the ground, stretching to the panel's width.
28      */
29     public void draw(Graphics g, int panelWidth) {
30         if (groundImage == null) return;
31
32         int yDraw = groundSurfaceY - (groundHeight - surfaceThickness);
33         g.drawImage(groundImage, x: 0, yDraw, panelWidth, groundHeight, observer: null);
34     }
35
36     public int getGroundY() {
37         return groundSurfaceY;
38     }
39
40     /**
41      * Updates the ground's Y position based on the panel's height.
42      */
43     public void updateGroundY(int panelHeight) {
44         groundSurfaceY = panelHeight - surfaceThickness;
45     }
46 }
47
```

Variabel :

- groundSurfaceY : posisi permukaan tanah
- groundHeight : tinggi tanah
- surfaceThickness : ketebalan permukaan tanah
- groundImage : gambar tanah

Constructor Level() :

- Load gambar tanah dari /assets/Ground.png

- Jika gagal : groundImage = null
- draw(Graphics g, int panelWidth) :
- Gambar tanah dengan lebar panel
 - Y posisi digeser sesuai tinggi & ketebalan permukaan
- getGroundY() :
- Mengembalikan posisi permukaan tanah
- updateGroundY(int panelHeight) :
- Update groundSurfaceY berdasarkan tinggi panel

Class Enemy

Buka file di laptop



Enemy.java

```
src > J Enemy.java > ...
1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.geom.AffineTransform;
4 import java.net.URL;
5
6 public class Enemy {
7
8     private enum AnimationState {
9         IDLE, RUN, DIE
10    }
11
12    private int x, y;
13    private int width = 120, height = 120;
14    private int hitboxWidth = 60, hitboxHeight = 60;
15    private int dx = 2;
16    private int dy = 0;
17    private boolean alive = true;
18    private boolean facingRight = true;
19
20    private AnimationState currentState = AnimationState.IDLE;
21    private int dieAnimTimer = 0;
22
23    private Image idlesheet, runsheet, diesheet;
24
25    private final int idleFrameCount = 8;
26    private final int runFrameCount = 8;
27    private final int dieFrameCount = 15;
28
29    private final int idleAnimSpeed = 8;
30    private final int runAnimSpeed = 2;
31    private final int dieAnimSpeed = 3;
32
33    private int frameIndex = 0;
34    private int animCounter = 0;
35
36    public Enemy(int x, int y) {
37        this.x = x;
```

```
src > J Enemy.java > ...
  6  public class Enemy {
36      public Enemy(int x, int y) {
38          this.y = y;
39          loadSpriteSheets();
40      }
41
42      private void loadSpriteSheets() {
43          try {
44              URL idleUrl = getClass().getResource(name: "/assets/Mushroom-Idle.png");
45              if (idleUrl != null) idleSheet = new ImageIcon(idleUrl).getImage();
46
47              URL runUrl = getClass().getResource(name: "/assets/Mushroom-Run.png");
48              if (runUrl != null) runSheet = new ImageIcon(runUrl).getImage();
49
50              URL dieUrl = getClass().getResource(name: "/assets/Mushroom-Die.png");
51              if (dieUrl != null) dieSheet = new ImageIcon(dieUrl).getImage();
52          } catch (Exception e) {
53              System.err.println(x: "Failed to load enemy sprite sheets.");
54              e.printStackTrace();
55          }
56      }
57
58      public void update(Level level) {
59          if (!alive) {
60              if (dieAnimTimer > 0) {
61                  dieAnimTimer--;
62                  updateAnimation();
63              }
64              return;
65          }
66
67          x += dx;
68          if (x <= 0 || x + width >= GameFrame.WIDTH) {
69              dx *= -1;
70          }
71          facingRight = (dx > 0);
72      }

```

```
src > J Enemy.java > ...
  6  public class Enemy {
  58      public void update(Level level) {
  73          dy += 1; // Gravity
  74          y += dy;
  75
  76          int groundY = level.getGroundY();
  77          if (y >= groundY - height) {
  78              y = groundY - height;
  79              dy = 0;
  80          }
  81
  82          currentState = (dx != 0) ? AnimationState.RUN : AnimationState.IDLE;
  83          updateAnimation();
  84      }
  85
  86      private void updateAnimation() {
  87          animCounter++;
  88          int currentAnimSpeed = 1;
  89          int currentFrameCount = 1;
  90
  91          switch (currentState) {
  92              case IDLE:
  93                  currentAnimSpeed = idleAnimSpeed;
  94                  currentFrameCount = idleFrameCount;
  95                  break;
  96              case RUN:
  97                  currentAnimSpeed = runAnimSpeed;
  98                  currentFrameCount = runFrameCount;
  99                  break;
 100             case DIE:
 101                 currentAnimSpeed = dieAnimSpeed;
 102                 currentFrameCount = dieFrameCount;
 103                 break;
 104         }
 105
 106         if (animCounter >= currentAnimSpeed) {
 107             animCounter = 0;
```

```
src > J Enemy.java > ...
6  public class Enemy {
86      private void updateAnimation() {
107         animCounter = 0;
108         if (currentState == AnimationState.DIE) {
109             if (frameIndex < currentFrameCount - 1) {
110                 frameIndex++;
111             }
112         } else {
113             frameIndex = (frameIndex + 1) % currentFrameCount;
114         }
115     }
116 }
117
118 public void draw(Graphics g) {
119     Image currentSheet = getCurrentImageSheet();
120     int currentFrameCount = getCurrentFrameCount();
121
122     if (currentSheet == null || currentSheet.getWidth(observer: null) <= 0) {
123         return;
124     }
125
126     int frameW = currentSheet.getWidth(observer: null) / currentFrameCount;
127     int sx1 = frameIndex * frameW;
128
129     Graphics2D g2d = (Graphics2D) g.create();
130     if (facingRight) {
131         g2d.drawImage(currentSheet, x, y, x + width, y + height, sx1, sy1: 0, sx1 + frameW);
132     } else {
133         g2d.drawImage(currentSheet, x + width, y, x, y + height, sx1, sy1: 0, sx1 + frameW);
134     }
135     g2d.dispose();
136 }
137
138 private Image getCurrentImageSheet() {
139     if (!alive && dieAnimTimer > 0) {
140         return diesheet;
141     }
}
```

```
src > J Enemy.java > ...
6  public class Enemy {
138      private Image getCurrentImageSheet() {
141          }
142          if (alive) {
143              switch (currentState) {
144                  case IDLE: return idleSheet;
145                  case RUN: return runSheet;
146                  default: return idleSheet;
147              }
148          }
149          return null;
150      }
151
152      private int getCurrentFrameCount() {
153          if (!alive && dieAnimTimer > 0) {
154              return dieFrameCount;
155          }
156          if (alive) {
157              switch (currentState) {
158                  case IDLE: return idleFrameCount;
159                  case RUN: return runFrameCount;
160                  default: return idleFrameCount;
161              }
162          }
163          return 1;
164      }
165
166      public Rectangle getBounds() {
167          int hitboxX = x + (width - hitboxWidth) / 2;
168          int hitboxY = y + (height - hitboxHeight); // Align hitbox with bottom
169          return new Rectangle(hitboxX, hitboxY, hitboxWidth, hitboxHeight);
170      }
171
172      public boolean isAlive() {
173          return alive;
174      }
175
```

```

src > J Enemy.java > ...
6  public class Enemy {
170 }
171
172     public boolean isAlive() {
173         return alive;
174     }
175
176     public void die() {
177         if (!alive) return;
178         alive = false;
179         currentState = AnimationState.DIE;
180         dieAnimTimer = dieFrameCount * dieAnimSpeed; // Animation duration
181         frameIndex = 0;
182         animCounter = 0;
183     }
184
185     public void respawn(Level level) {
186         alive = true;
187         currentState = AnimationState.IDLE;
188         dieAnimTimer = 0;
189         frameIndex = 0;
190
191         int maxX = GameFrame.WIDTH - width;
192         x = (int) (Math.random() * (maxX > 0 ? maxX : 0));
193
194         y = level.getGroundY() - height;
195         dy = 0;
196
197         dx = (Math.random() > 0.5) ? 2 : -2;
198     }
199 }
200

```

AnimationState enum : IDLE, RUN, DIE

Variabel posisi & ukuran : x, y, width, height, hitboxWidth, hitboxHeight

Variabel gerak & status : dx, dy, alive, facingRight

Animasi : currentState, dieAnimTimer, frameIndex, animCounter

Sprite sheets : idleSheet, runSheet, dieSheet

Frame & animasi speed : idleFrameCount/Speed, runFrameCount/Speed, dieFrameCount/Speed

Constructor Enemy(x, y) :

- Set posisi awal
- Load sprite sheets

loadSpriteSheets() :

- Muat gambar idle, run, die dari /assets/

update(Level level) :

- Jika mati : jalankan animasi DIE sampai selesai
- Jika hidup : update posisi horizontal (patah balik di tepi layar)
- Update arah hadap (facingRight)
- Terapkan gravitasi (dy)
- Check tanah (y >= ground)

- Set state IDLE / RUN sesuai dx
- Update animasi

updateAnimation() :

- Tambah animCounter
- Update frameIndex sesuai speed & jumlah frame
- DIE : frameIndex berhenti di akhir animasi

draw(Graphics g) :

- Ambil sheet saat ini (getCurrentImageSheet())
- Hitung frame width
- Draw frame, flip horizontal jika menghadap kiri

getCurrentImageSheet() : ambil sprite sheet sesuai state & status alive

getCurrentFrameCount() : ambil jumlah frame sesuai state & alive

getBounds() :

- Hitbox rectangle untuk collision detection

isAlive() : return status alive

die() :

- Set alive = false
- Set state DIE
- Set dieAnimTimer = durasi animasi
- Reset frameIndex & animCounter

respawn(Level level) :

- Set alive = true, state = IDLE, reset animasi
- Set x random di dalam layar
- Set y di atas ground
- Set dx random arah kiri/kanan

Classs Floating

Buka file dari laptop



FloatingText.java

```
src > J FloatingText.java > ⚡ FloatingText > ⚡ update()
1  import java.awt.*;
2
3  public class FloatingText {
4      private String text;
5      private int x, y;
6      private int life;
7      private final int maxLife;
8      private Color color;
9      private Font font;
10
11     public FloatingText(String text, int x, int y, int duration, Color color, Font font) {
12         this.text = text;
13         this.x = x;
14         this.y = y;
15         this.maxLife = duration;
16         this.life = duration;
17         this.color = color;
18         this.font = font;
19     }
20
21
22     public void update() {
23         if (isAlive()) {
24             life--;
25             y -= 2;
26         }
27     }
28
29     public boolean isAlive() {
30         return life > 0;
31     }
32
33
34     public void draw(Graphics g) {
35         if (!isAlive()) return;
36
37         Graphics2D g2d = (Graphics2D) g.create();
```

```

33
34     public void draw(Graphics g) {
35         if (!isAlive()) return;
36
37         Graphics2D g2d = (Graphics2D) g.create();
38
39         float alpha = (float) life / maxLife;
40         g2d.setComposite(AlphaComposite.getInstance(AlphaComposite.SRC_OVER, Math.max(a: 0, al
41
42         g2d.setFont(font);
43         FontMetrics fm = g2d.getFontMetrics();
44         int textWidth = fm.stringWidth(text);
45         int drawX = x - textWidth / 2;
46
47         // Draw outline
48         g2d.setColor(Color.BLACK);
49         g2d.drawString(text, drawX + 1, y + 1);
50         g2d.drawString(text, drawX - 1, y - 1);
51         g2d.drawString(text, drawX + 1, y - 1);
52         g2d.drawString(text, drawX - 1, y + 1);
53
54         // Draw main text
55         g2d.setColor(color);
56         g2d.drawString(text, drawX, y);
57
58         g2d.dispose();
59     }
60 }
61

```

Variabel :

- text : teks yang ditampilkan
- x, y : posisi teks di layar
- life : sisa waktu teks tampil
- maxLife : durasi teks
- color : warna teks
- font : font teks

Constructor FloatingText(text, x, y, duration, color, font) :

- Inisialisasi teks, posisi, durasi, warna, font

update() :

- Kurangi life tiap frame
- Geser teks ke atas ($y -= 2$)

isAlive() :

- Return true jika $life > 0$

draw(Graphics g) :

- Hentikan jika !isAlive()
- Gunakan Graphics2D untuk alpha/fade-out
- Hitung alpha : $life / maxLife$
- Draw outline hitam di sekitar teks

- Draw teks utama dengan warna color
- Dispose Graphics2D setelah selesai

Class Score

Buka file dari laptop



Score.java

```
src > J Score.java > Score Score
1 | public class Score {
2 |     private int score = 0;
3 |     private int killCount = 0;
4 |
5 |     public void addScore(int value) {
6 |         if (value > 0) {
7 |             score += value;
8 |         }
9 |     }
10 |
11    public int getScore() {
12        return score;
13    }
14
15    public void addKill() {
16        killCount++;
17    }
18
19    public int getKillCount() {
20        return killCount;
21    }
22
23    public void reset() {
24        score = 0;
25        killCount = 0;
26    }
27
28 }
29
```

Variabel :

- score : skor pemain
- killCount : jumlah musuh yang dikalahkan

addScore(int value) :

- Tambahkan nilai ke score jika value > 0

getScore() :

- Mengembalikan skor saat ini

addKill() :

- Tambahkan 1 ke killCount

getKillCount() :

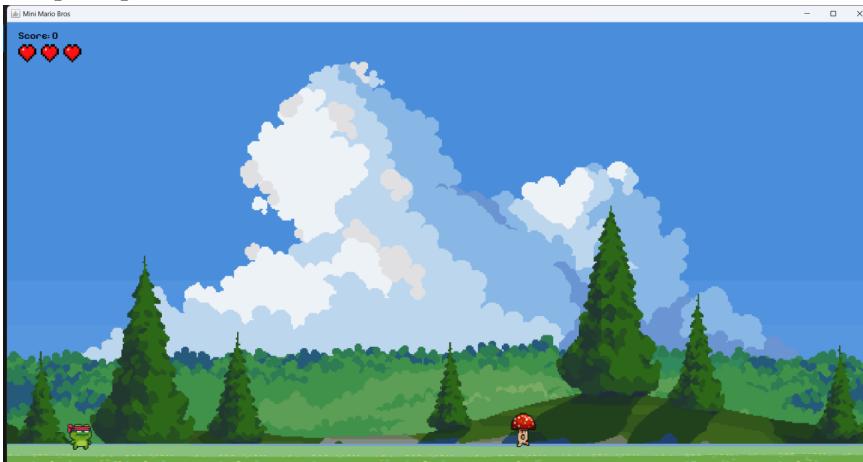
- Mengembalikan jumlah kill saat ini

reset() :

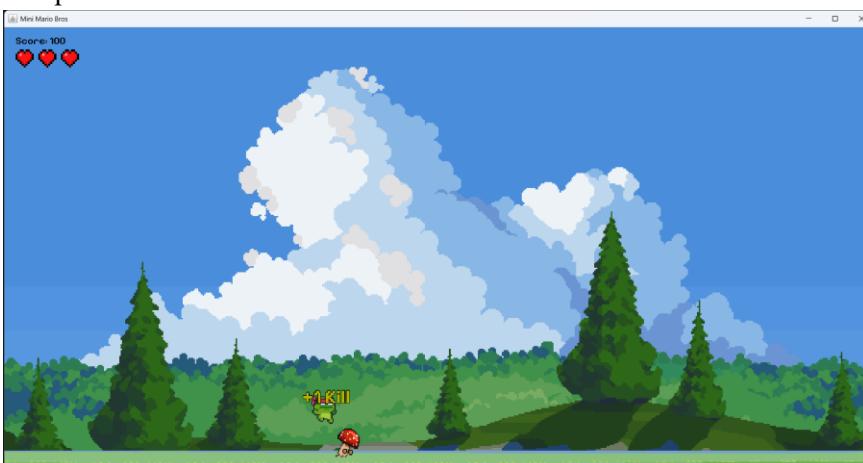
- Reset score dan killCount ke 0

G. Penjelasan Screenshot Tampilan yang Dihasilkan

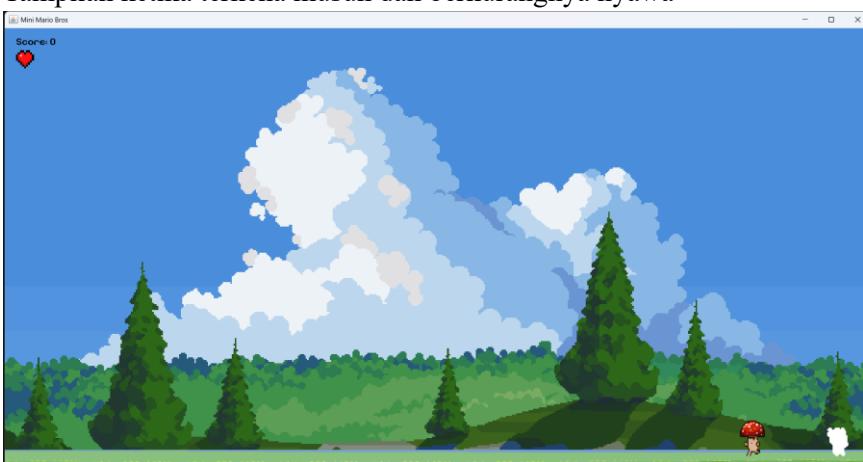
Tampilan pertama ketika run code



Tampilan ketika kill musuh



Tampilan ketika terkena musuh dan berkurangnya nyawa



Tampilan ketika game over dan tampilan tombol play untuk bermain ulang



H. Penjelasan Screenshot Status Unggah GitHub/GitLab

1. File/Folder Project Terbaru

- Semua file yang sudah dibuat dan dimodifikasi berhasil diunggah.
- Menunjukkan struktur project di repository sama dengan lokal.

2. Commit Terakhir

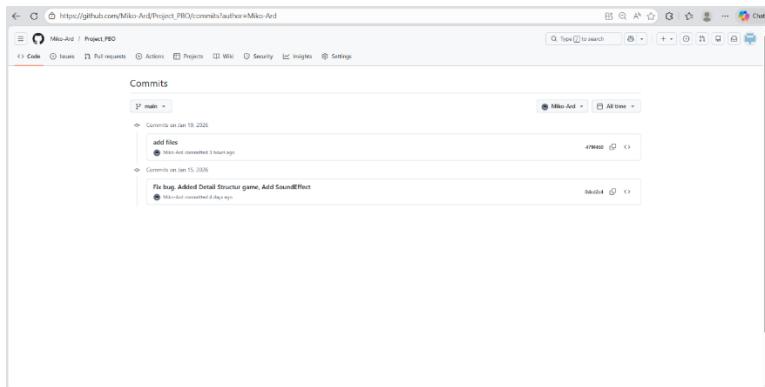
- Terdapat pesan commit sesuai yang dibuat saat push.
- Membuktikan perubahan terbaru berhasil dicatat di repository.

3. Tanggal dan Waktu Commit (Opsiional)

- Memberikan bukti kapan project terakhir diunggah.

Dengan screenshot ini, dapat dipastikan bahwa project telah berhasil diunggah ke repository GitHub/GitLab dan siap untuk diakses atau dinilai.

A screenshot of a GitHub repository page for "Project_PBO". The page shows a list of files: .idea, .gitignore, and TotalAlhiric.html. There are three commits listed, all made by "Miko-Ard" 5 days ago. The commits are: "first commit", "add files", and "first commit". The repository has 0 stars, 0 forks, and 0 watchers. It also has 0 releases, 0 packages, and 2 contributors: "Miko-Ard" and "akay9114". The repository uses Java as its language.



Commits

2 main ·

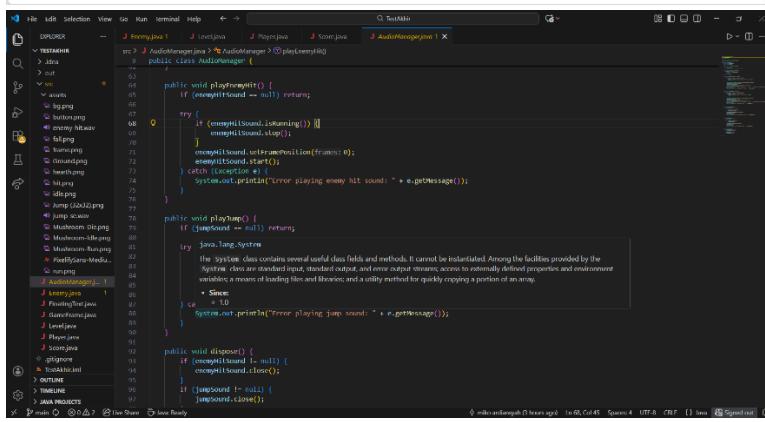
• Commits on Jan 19, 2020

• add file

• fix bug, Added Detail Struct game, Add SoundEffect

• Commits on Jan 15, 2020

• fix bug, Added Detail Struct game, Add SoundEffect



```
public class AudioManager {  
    private Thread soundThread;  
    private Queue<Runnable> queue;  
    private boolean isRunning;  
    private final Object lock = new Object();  
  
    public void playEnemyHit() {  
        if (soundThread == null) return;  
        try {  
            if (enemysound.isRunning()) {  
                enemysound.pause();  
            }  
            enemysound.setFramePosition(frame: 0);  
            enemysound.start();  
        } catch (Exception e) {  
            System.out.println("Error playing enemy hit sound: " + e.getMessage());  
        }  
    }  
  
    public void playJump() {  
        if (jumpsound == null) return;  
        try {  
            if (jumpsound.isRunning()) {  
                jumpsound.pause();  
            }  
            System.out.println("Error playing jump sound: " + e.getMessage());  
        } catch (Exception e) {  
            System.out.println("Error playing jump sound: " + e.getMessage());  
        }  
    }  
  
    public void dispose() {  
        if (soundThread != null) {  
            enemysound.close();  
        }  
        if (jumpsound != null) {  
            jumpsound.close();  
        }  
    }  
}
```

The image shows two screenshots side-by-side. The left screenshot is from an IDE (IntelliJ IDEA) displaying Java code for `GameFrame.java`. The code initializes a window, sets its size and location, and makes it visible. The right screenshot is from a GitHub repository page for `Project_PBO`, specifically showing the commit history for the file `TestAkhir.iml`. The commit was made by `abay9114` and is described as the first commit.

IDE Screenshot (Left):

```
public class GameFrame extends JPanel implements ActionListener, KeyListener {    public static void main(String[] args) {        GameFrame game = new GameFrame();        game.setVisible(true);        frame.setSize(800, 600);        frame.setLocationRelativeTo(null);        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);        frame.setResizable(false);        frame.setVisible(true);    }}
```

Github Commit History (Right):

Project_PBO / TestAkhir.iml

first commit

c05c0d5 · 5 days ago

```
<?xml version="1.0" encoding="UTF-8"?><module type="JAVA_MODULE" version="4">    <component name="ModuleRootManager" inherit-compiler-output="true">        <exclude-output>        <content url="file://$MODULE_DIR$">            <sourceFolder url="file://$MODULE_DIR$/src" isTestSource="false" />        </content>        <orderEntry type="inheritSdk" />        <orderEntry type="sourceFolder" forTests="false" />    </component></module>
```

I. Analisis Pengerjaan Proyek

1. Waktu

- Proyek ini dikerjakan selama **1 minggu**.
- Terdapat kendala waktu karena **banyak tugas mata kuliah lain** yang juga harus dikerjakan.
- Urutan pengerjaan proyek:
 1. Membuat repository (GitHub/GitLab)
 2. Analisis game dan spesifikasi
 3. Pembuatan kode program
 4. Desain GUI dan pengujian

2. Ketercapaian Spesifikasi

- Beberapa fitur **belum berhasil diimplementasikan**, seperti:
 - Musik latar (background music)
 - Halaman play/menu awal
 - Tipe musuh yang lebih banyak
- Fitur yang **paling sukses** adalah **penataan GUI** (tampilan antarmuka, skor, indikator nyawa, game over, respawn).
- Fitur yang **paling sulit** adalah **sound effect** karena harus mencari dan mengatur aset audio yang sesuai.

3. Biaya

- Biaya yang dibutuhkan: **0 rupiah**, karena semua software dan aset menggunakan versi gratis.
- Secara keseluruhan, biaya **sesuai estimasi** awal.
- Strategi efisiensi biaya: menggunakan **asset gratis**, termasuk gambar, sprite, font, dan sound effect.

4. Kendala

- Kendala teknis:
 - Error dalam pengaturan sprite animasi dan audio
 - Sound effect sulit disinkronisasi
- Kendala non-teknis:
 - Waktu terbatas karena banyak tugas lain
 - Mencari aset gratis yang sesuai dan berkualitas
- Cara mengatasi:
 - Mencari lebih banyak aset gratis melalui internet
 - Mengatur prioritas pengerjaan fitur yang paling penting (GUI, logika game)

5. Tantangan Masa Depan

- Fitur yang dapat ditambahkan:
 - Musik latar (background music)
 - Halaman play/menu awal
 - High score (leaderboard)
 - Tipe musuh yang lebih bervariasi
 - Mode permainan tambahan (misal time attack, endless mode)

- Optimisasi lain:
 - Meningkatkan performa game agar animasi lebih smooth
 - Penambahan AI musuh lebih kompleks
- Peluang pengembangan:
 - Bisa dijadikan **proyek portofolio** untuk tugas akhir atau pameran
 - Dapat dikembangkan menjadi game kecil komersial dengan asset berlisensi dan musik original

6. Lain-lain

- **Pelajaran penting:**
 - Perencanaan awal sangat membantu (membuat repo → analisis → coding → GUI)
 - Mengatur prioritas fitur penting membuat penggerjaan tetap selesai walau terbatas waktu
 - Manajemen waktu menjadi kunci saat mengerjakan proyek dengan banyak tugas lain
- **Jika dikerjakan ulang:**
 - Akan memulai lebih awal agar bisa menambahkan semua fitur seperti musik, menu, dan tipe musuh tambahan
 - Mencari asset lebih banyak sejak awal
- **Saran untuk pengembang lain:**
 - Gunakan repository versi kontrol (Git) dari awal
 - Prioritaskan fitur inti (GUI, logika game) sebelum menambahkan fitur tambahan
 - Manfaatkan asset gratis untuk prototipe, tapi pastikan kualitasnya sesuai kebutuhan game