

LAPORAN PROYEK PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA 2024
THE FATEFORGED : TEXT-BASED ADVENTURE GAME



Oleh Kelompok 20

Anggota :

Ahmad Muslihul Khair F1D02310001

Wadis Freandly F1D02310094

I Putu Ananta Sugiarta F1D02310113

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MATARAM

2024

1.1 Judul Proyek

Judul proyek kami yaitu “*The Fateforged : Text-Based Adventure Game*”

1.2 Latar Belakang

Lalalallala

1.3 Deskripsi Program

Di dunia fantasi bernama Altheir, sebuah bencana besar yang terjadi akibat banyaknya pendosa di dunia tersebut menyebabkan para dewa murka dan mengancam keseimbangan antara dunia manusia dan dunia para dewa. Tiga pahlawan dengan takdir yang saling terkait bernama Ananta, Wadis, dan Khair harus bekerja sama untuk mencegah kehancuran dunia Altheir. Namun, setiap pahlawan memiliki motivasi dan rahasia yang dapat mempererat atau justru menghancurkan kerja sama mereka.

1. Karakter Utama

a. Ananta

- i. Ananta adalah Sang Penjaga Keseimbangan.
- ii. Memiliki Kepribadian yang Bijaksana
- iii. Ananta merupakan penjaga terakhir dari Pohon Kehidupan, sumber energi yang menjaga dunia tetap seimbang.
- iv. Ananta ingin mengembalikan energi Pohon Kehidupan sebelum para dewa menghancurkan dunia.

b. Wadis

- i. Wadis adalah Sang Pengelana Kegelapan.
- ii. Memiliki Kepribadian yang Tegas.
- iii. Wadis merupakan seorang pemburu bayaran yang dikenal karena kemampuannya bernegosiasi dengan roh-roh jahat. Ia terikat oleh kontrak dengan entitas misterius yang menawarkan kekuatan besar, tetapi dengan harga mahal: jiwanya.
- iv. Wadis ingin Membebaskan dirinya dari kontrak dan membalas dendam kepada entitas tersebut.

c. Khair

- i. Khair adalah Sang Penyembuh Cahaya.
- ii. Memiliki Kepribadian yang Optimis dan penuh keyakinan, tetapi terkadang naif.
- iii. Khair merupakan seorang penyembuh dari desa terpencil yang dihancurkan oleh makhluk kegelapan. Ia diberkati dengan kekuatan suci,

yang membuatnya menjadi target utama oleh dewa gerbang neraka bernama sagose.

- iv. Khair ingin Menyelamatkan sebanyak mungkin orang dan menghancurkan sumber kegelapan yang menjadi pemicu munculnya keinginan negatif manusia.

2. Alur Cerita

a. Awal Perjalanan (Prologue)

Ketiga karakter dipertemukan secara tidak sengaja ketika sebuah desa diserang oleh para prajurit dari kerajaan. Ananta berusaha melindungi penduduk, Wadis sedang mencari targetnya, dan Khair mencoba menyembuhkan para korban. Mereka bergabung sementara untuk menghadapi ancaman, tetapi masing-masing memiliki kecurigaan terhadap satu sama lain. Pemain memilih salah satu dari ketiga karakter untuk dimainkan. Hal ini akan menentukan bagaimana cerita akan berjalan dan menentukan ending seperti apa nantinya. Mereka melakukan perjalanan menuju alam dewa melalui 3 alam keajaiban di antaranya alam manusia, alam monster, dan alam roh. Setiap alam memiliki entitas yang harus dikalahkan antara Ananta, Wadis ataupun Khair. Mereka harus mengalahkan para entitas ini dengan menggunakan kekuatan spesial yang mereka miliki.

b. Pertemuan Pertama Alam Manusia (Act 1)

Pada alam manusia salah satu dari mereka melakukan sebuah misi untuk menemukan bunga Edelweis yang dipercaya dapat meningkatkan keberuntungan. Pada saat menjalani misi tersebut, salah satu dari mereka bertemu dengan seseorang bernama Fakhroni yang ingin membunuh mereka bertiga, Fakhroni dikenal sebagai pembunuh keji yang selalu membunuh orang demi mendapatkan energi roh korbannya guna meningkatkan status kekuatannya. Salah satu dari mereka harus berhasil mengalahkan Fakhroni.

c. Pengkhianatan dan Kepercayaan Alam Monster (Act 2)

Karena berhasil mengalahkan Fakhroni, Ananta menerima sisa roh yang ada di dalam tubuh Fakhroni yang dapat mempengaruhi powernya secara random. Dari sini Wadis mulai menunjukkan sisi gelapnya, memanipulasi situasi untuk kepentingannya sendiri dikarenakan hanya Ananta yang menerima sisa roh walau walau yang membunuh Fakhroni adalah Wadis ataupun Khair. Ananta harus berusaha untuk merubah alur pemikiran wadis

dengan percakapan yang dipilih. Khair menghadapi dilema moral ketika kekuatan sucinya harus digunakan untuk tujuan yang merusak. Pilihan pemain akan memengaruhi apakah kelompok tetap bersatu atau pecah di titik ini. Setelah kejadian di Alam Manusia, perjalanan menuju Alam Monster dipenuhi oleh tantangan berat. Alam ini dipenuhi oleh makhluk-makhluk buas dan kejam yang terbentuk dari energi gelap dan sisa-sisa roh jahat. Monster-monster di sini tidak hanya kuat secara fisik, tetapi juga memiliki kemampuan untuk memanipulasi pikiran, memaksa ketiganya menghadapi konflik internal masing-masing.

d. Konflik Terbesar (Act 3)

Di puncak *Shrine of Eternity*, mereka menghadapi musuh utama:

- i. Roh Penghancur, entitas yang menciptakan Eclipse.

Dalam pertarungan klimaks, pemain dapat memilih untuk:

- i. Mengorbankan salah satu pahlawan untuk menyegel kegelapan.
- ii. Bekerja sama untuk menggunakan kekuatan mereka dan menyelamatkan dunia (jalan tersulit).
- iii. Mengkhianati tujuan utama dan membiarkan kegelapan menang untuk kepentingan pribadi.

e. Akhir Cerita (Multiple Endings)

- i. Jika mereka bekerja sama: Eclipse dihentikan, tetapi dengan harga mahal, seperti kehilangan kekuatan atau kehilangan seorang teman.
- ii. Jika mereka pecah: Dunia terpecah menjadi dua dimensi, dan masing-masing karakter mendapatkan akhir individu berdasarkan pilihan pemain.
- iii. Jika kegelapan menang: Dunia tenggelam dalam Eclipse, tetapi salah satu karakter menjadi penguasa baru dunia yang gelap.

1.4 Algoritma

Berikut ini merupakan algoritma yang digunakan untuk menyelesaikan permasalahan pada program:

1. Lalalallalalal
2. .
3. .
4. .
5. .

1.5 Penjelasan Kode

1.5.1 Dewa Class

```
package entity;

public class Dewa extends Karakter {
    String specialSkill;

    public Dewa(String name, int health, int power, String
personality, String specialSkill) {
        super(name, health, 0, power, personality);
        this.specialSkill = specialSkill;
    }

    public void performAction(String action) {
        System.out.println(this.name + " is performing action: " +
action);
    }

    @Override
    public void showKarakterInfo() {
        super.showKarakterInfo();
        System.out.println("Special Skill: " + this.specialSkill);
    }
}
```

Script di atas merupakan

1.5.2 Human Class

```
package entity;

public class Human extends Karakter {
    public Human(String name, int health, int energy, int power,
String personality) {
        super(name, health, energy, power, personality);
    }
}
```

Script di atas merupakan

1.5.3 Karakter Class

```
package entity;

public class Karakter {
    public String name;
```

```

public int health;
public int energy;
public int power;
public String personality;

    public Karakter(String name, int health, int energy, int
power, String personality) {
        this.name = name;
        this.health = health;
        this.energy = energy;
        this.power = power;
        this.personality = personality;
    }

    public void attack(Karakter opponent) {
        double damage = this.power * (this.energy / 250.0);
        opponent.health -= (int) damage;

        double energyCost = this.power * 0.2;
        this.energy -= (int) energyCost;

        if (this.energy < 0) this.energy = 0;
        if (opponent.health < 0) opponent.health = 0;
        if (opponent.energy < 0) opponent.energy = 0;
    }

    public void useEnergy(int amount) {
        if (this.energy >= amount) {
            this.energy -= amount;
            System.out.println(this.name + " menggunakan " + amount
+ " energi. Sisa energi: " + this.energy);
        } else {
            System.out.println(this.name + " Energi tidak
mencukupi");
        }
    }

    public void restoreEnergy() {
        int baseRecovery = 20;
        double healthFactor = this.health / 100.0;
        double powerFactor = 1 + (this.power / 50.0);
    }

```

```

        int energyRecovery = (int) (baseRecovery * healthFactor /
powerFactor);

        this.energy += energyRecovery;
        if (this.energy > 100) {
            this.energy = 100;
        }
    }

    public void showKarakterInfo() {
        System.out.println("=== Karakter Info ===");
        System.out.println("Name: " + this.name);
        System.out.println("Health: " + this.health);
        System.out.println("Energy: " + this.energy);
        System.out.println("Power: " + this.power);
        System.out.println("Personality: " + this.personality);
        System.out.println("=====");
    }
}

```

Script di atas merupakan

1.5.4 Monster Class

```

package entity;

public class Monster extends Karakter {
    String monsterType;

    public Monster(String name, int health, int energy, int
power, String personality, String monsterType) {
        super(name, health, energy, power, personality);
        this.monsterType = monsterType;
    }

    public void performAction(String action) {
        System.out.println(this.name + " is performing action: " +
action);
    }

    @Override
    public void showKarakterInfo() {
        super.showKarakterInfo();
    }
}

```

```
        System.out.println("Monster Type: " + this.monsterType);
    }
}
```

Script di atas merupakan

1.5.5 Player Class

```
package entity;

public class Player extends Karakter {
    String role;
    String goal;

    public Player(String name, String role, int health, int
energy, int power, String personality, String goal) {
        super(name, health, energy, power, personality);
        this.role = role;
        this.goal = goal;
    }

    @Override
    public void showKarakterInfo() {
        super.showKarakterInfo();
        System.out.println("Role: " + this.role);
        System.out.println("Goal: " + this.goal);
    }

    public void performAction(String action) {
        System.out.println(this.name + " is performing action: " +
action);
    }
}
```

Script di atas merupakan

1.5.6 Roh Class

```
package entity;

public class Roh extends Karakter {
    String derajat;

    public Roh(String name, int health, int energy, int power,
String personality, String derajat) {
```



```

        super(name, health, energy, power, personality);
        this.derajat = derajat;
    }

    public void performAction(String action) {
        System.out.println(this.name + " is performing action: " +
action);
    }
}

```

Script di atas merupakan

1.5.7 AlurCerita Class

```

package util;

import java.util.Scanner;
import entity.*;

public class AlurCerita {
    public Cerita root;
    Scanner scanner = new Scanner(System.in);
    QueuePlayer queuePlayer = new QueuePlayer();

    // Membuat objek karakter Ananta
    Player player1 = new Player(
        "Ananta",
        "Sang Penjaga Keseimbangan",
        100, // Health
        80,  // Energy
        90,  // Power
        "Bijaksana",
        "Mengembalikan energi Pohon Kehidupan sebelum para dewa
menghancurkan dunia"
    );

    // Membuat objek karakter Wadis
    Player player2 = new Player(
        "Wadis",
        "Sang Pengelana Kegelapan",
        100, // Health
        100, // Energy
        95,  // Power

```

```

        "Tegas",
        "Membebaskan dirinya dari kontrak dan membalas dendam
        kepada entitas tersebut"
    );

    // Membuat objek karakter Khair
    Player player3 = new Player(
        "Khair",
        "Sang Penyembuh Cahaya",
        100, // Health
        100, // Energy
        75, // Power
        "Optimis dan penuh keyakinan, tetapi terkadang naif",
        "Menyelamatkan sebanyak mungkin orang dan menghancurkan
        sumber kegelapan"
    );

    // Membuat objek karakter Fakhroni
    Human human = new Human(
        "Fakhroni",
        100, // Health
        100, // Energy
        100, // Power
        "Periang dan sangat sopan kepada orang baru"
    );

    // Membuat objek Monster Xenomorph
    Monster monster1 = new Monster(
        "Xenomorph",
        300, // Health
        200, // Energy
        120, // Power
        "Agresif dan pemaarah",
        "Alien"
    );

    // Membuat objek Monster Carberus
    Monster monster2 = new Monster(
        "Carberus",
        150, // Health
        150, // Energy

```

```

        90, // Power
        "Patuh dan Tegas",
        "Anjing Penjaga"
    );

    // Membuat objek Roh Umbra
    Roh roh = new Roh(
        "Umbra",
        200, // Health
        150, // Energy
        99, // Power
        "Independen dari kegelapan",
        "Penjaga"
    );

    // Membuat objek Dewa Ares
    Dewa dewa1 = new Dewa(
        "Ares",
        800, // Health
        500, // Power
        "Liar, impulsif, dan haus darah",
        "Penyembuhan Masif"
    );

    // Membuat objek Dewa Moros
    Dewa dewa2 = new Dewa(
        "Moros",
        500, // Health
        400, // Power
        "Tidak berperasaan dan netral",
        "Aura Kehampaan"
    );

    public AlurCerita() {
        root = null;
        queuePlayer.enqueue(player1);
        queuePlayer.enqueue(player2);
        queuePlayer.enqueue(player3);
    }

    public void createRoot(String kode) {

```

```

        root = new Cerita(kode);
    }

    public void inorderInput(Cerita node, Cerita newNode, String
direction, String kode) {
        if (node != null) {
            inorderInput(node.left, newNode, direction, kode);
            if (node.kode.equals(kode)) {
                if (direction.equals("kanan")) {
                    node.right = newNode;
                } else {
                    node.left = newNode;
                }
            }
            return;
        }
        inorderInput(node.right, newNode, direction, kode);
    }
}

    public void PreOrderInputDialog(Cerita node, String
direction, String kode, String dialog) {
        if (node != null) {
            if (node.kode.equals(kode)) {
                if (direction.equals("kanan")) {
                    node.right.tambahDialog(dialog);
                } else {
                    node.left.tambahDialog(dialog);
                }
            }
            return;
        }
        PreOrderInputDialog(node.left, direction, kode, dialog);
        PreOrderInputDialog(node.right, direction, kode,
dialog);
    }
}

    public void levelOrderDisplay() {
        if (root == null) {
            return;
        }
    }
}

```

```

Queue queue = new Queue();
queue.enqueue(root);

while (queue.front != null) {
    Cerita current = queue.dequeue();
    System.out.print(current.kode);
    System.out.println();

    if (current.left != null) {
        queue.enqueue(current.left);
    }
    if (current.right != null) {
        queue.enqueue(current.right);
    }
}

}

public void battle(Karakter opponent) {
    String choose;
    QueuePlayer current = queuePlayer;
    Player currentPlayer;
    while (true) {
        currentPlayer = current.dequeue();

        // Tampilkan opsi
        System.out.println("=== Pertarungan Dimulai ===");
        System.out.println("1. Serang");

        if (currentPlayer.energy < 10) {
            System.out.println("2. Pulihkan Energi");
        }
        System.out.println("3. Menyerah");
        System.out.println("-----");
        System.out.println(currentPlayer.name + " [Health: " +
currentPlayer.health + ", Energy: " + currentPlayer.energy +
"]");
        System.out.println(opponent.name + " [Health: " +
opponent.health + ", Energy: " + opponent.energy + "]");
        System.out.print("Pilih aksi: ");
        choose = scanner.nextLine();
    }
}

```

```

// Bersihkan layar (opsional, tergantung terminal)
System.out.print("\033[H\033[2J");
System.out.flush();

// Periksa kondisi kemenangan sebelum aksi
if (currentPlayer.health <= 0) {
    System.out.println(currentPlayer.name + " telah kalah!
Pemenangnya adalah " + opponent.name + "!");
    break;
}
if (opponent.health <= 0) {
    System.out.println(opponent.name + " telah kalah!
Pemenangnya adalah " + currentPlayer.name + "!");
    break;
}

// Proses aksi pemain
if (choose.equals("1")) {
    currentPlayer.attack(opponent);
    if (opponent.health > 0) {
        opponent.attack(currentPlayer);
    }
} else if (choose.equals("2") && currentPlayer.energy <
10) {
    currentPlayer.restoreEnergy();
    if (opponent.health > 0) {
        opponent.attack(currentPlayer);
    }
} else if (choose.equals("3")) {
    System.out.println(currentPlayer.name + " memilih
menyerah. " + opponent.name + " adalah pemenangnya!");
    currentPlayer.health = 0; // Menyatakan currentPlayer
kalah
    break;
} else {
    System.out.println("Pilihan tidak valid. Silakan coba
lagi.");
}

if (opponent.energy < 15) {
    opponent.restoreEnergy();
}

```

```

    }

    // Periksa kondisi setelah aksi
    if (currentPlayer.energy <= 0 && opponent.energy <= 0) {
        System.out.println("Kedua karakter kehabisan energi
dan tidak bisa bertarung lagi.");
        break;
    }
    if (currentPlayer.health <= 0) {
        System.out.println(currentPlayer.name + " telah kalah!
Pemenangnya adalah " + opponent.name + "!");
        break;
    }
    if (opponent.health <= 0) {
        System.out.println(opponent.name + " telah kalah!
Pemenangnya adalah " + currentPlayer.name + "!");
        break;
    }

    current.enqueue(currentPlayer);
}
}

public void nextAct () {
    String choose1;
    Cerita current = root;
    while (true) {
        choose1 = scanner.nextLine();
        if (choose1.equals("")) {
            current.displayDialog();
            if (current.kode.equals("prolog")) {
                current = current.left;
            } else if (current.kode.equals("act1")){
                while (true) {
                    battle(human);
                    if (human.health < 1) {
                        current = current.left;
                        break;
                    } else {
                        current = current.right;
                        break;
                    }
                }
            }
        }
    }
}

```

```

    }
}
} else if (current.kode.equals("act2.1")) {
    while (true) {
        battle(monster1);
        if (monster1.health < 1) {
            current = current.left;
            break;
        } else {
            current = current.right;
            break;
        }
    }
}
} else if (current.kode.equals("act2.2")) {
    while (true) {
        battle(monster2);
        if (monster2.health < 1) {
            current = current.left;
            break;
        } else {
            current = current.right;
            break;
        }
    }
}
} else if (current.kode.equals("act3.1")) {
    while (true) {
        battle(roh);
        if (roh.health < 1) {
            current = current.left;
            break;
        } else {
            current = current.right;
            break;
        }
    }
}
} else if (current.kode.equals("act4.1")) {
    while (true) {
        battle(dewa1);
        if (dewa1.health < 1) {
            current = current.left;
            break;

```



```
        } else {
            current = current.right;
            break;
        }
    }
} else if (current.kode.equals("act4.2")) {
    while (true) {
        battle(dewa2);
        if (dewa2.health < 1) {
            current = current.left;
            break;
        } else {
            current = current.right;
            break;
        }
    }
}
}
```

Script di atas merupakan

1.5.8 Cerita Class

```
package util;

import java.util.Scanner;

public class Cerita {
    String kode;
    Dialog head;
    Cerita left;
    Cerita right;
    Scanner scanner = new Scanner(System.in);

    public Cerita(String kode) {
        this.kode = kode;
        this.head = null;
        this.left = null;
        this.right = null;
    }
}
```

```

public void tambahDialog(String dialog) {
    if (head == null) {
        head = new Dialog(dialog);
    } else {
        Dialog current = head;
        while (current.next != null) {
            current = current.next;
        }
        current.next = new Dialog(dialog);
    }
}

public void displayDialog () {
    String choose1;
    Dialog current = head;
    while (current != null) {
        choose1 = scanner.nextLine();
        System.out.print("\033[H\033[2J");
        System.out.flush();
        if (choose1.equals("")){
            System.out.println("\n" + current.dialog);
            current = current.next;
        }
    }
}
}

```

Script di atas merupakan

1.5.9 Dialog Class

```

package util;

public class Dialog {
    String dialog;
    Dialog next;

    public Dialog (String dialog) {
        this.dialog = dialog;
        this.next = null;
    }
}

```

Script di atas merupakan

1.5.10 NodeKarakter Class

```
package util;

import entity.Karakter;

public class NodeKarakter {
    public Karakter nodeKarakter;
    public NodeKarakter next;

    public NodeKarakter (Karakter nodeKarakter) {
        this.nodeKarakter = nodeKarakter;
        this.next = null;
    }
}
```

Script di atas merupakan

1.5.11 NodePlayer Class

```
package util;

import entity.Player;

public class NodePlayer {
    public Player nodePlayer;
    public NodePlayer next;

    public NodePlayer (Player nodePlayer) {
        this.nodePlayer = nodePlayer;
        this.next = null;
    }
}
```

Script di atas merupakan

1.5.12 NodeQueue Class

```
package util;

public class NodeQueue {
    Cerita nodeCerita;
    NodeQueue next;

    public NodeQueue (Cerita nodeCerita) {
```

```
        this.nodeCerita = nodeCerita;
        this.next = null;
    }
}
```

Script di atas merupakan

1.5.13 Queue Class

```
package util;

public class Queue {
    NodeQueue front;
    NodeQueue rear;

    public Queue() {
        this.front = null;
        this.rear = null;
    }

    public void enqueue(Cerita nodeCerita) {
        NodeQueue newNode = new NodeQueue(nodeCerita);
        if (rear == null) {
            front = rear = newNode;
        } else {
            rear.next = newNode;
            rear = newNode;
        }
    }

    public Cerita dequeue() {
        if (front == null) {
            System.out.println("Queue is empty. Cannot dequeue.");
            return null;
        }

        Cerita nodeCerita = front.nodeCerita;
        front = front.next;

        if (front == null) {
            rear = null;
        }

        return nodeCerita;
    }
}
```

```
}  
}
```

Script di atas merupakan

1.5.14 QueueKarakter Class

```
package util;  
  
import entity.Karakter;  
  
public class QueueKarakter {  
    NodeKarakter front, rear;  
  
    public QueueKarakter () {  
        this.front = null;  
        this.rear = null;  
    }  
  
    public void enqueue(Karakter nodeKarakter) {  
        NodeKarakter newNode = new NodeKarakter(nodeKarakter);  
        if (front == null) {  
            front = rear = newNode;  
        } else {  
            rear.next = newNode;  
            rear = newNode;  
        }  
    }  
  
    public Karakter dequeue() {  
        if (front == null) {  
            return null;  
        }  
        Karakter nodeKarakter = front;  
        front = front.next;  
        return nodeKarakter;  
    }  
}
```

Script di atas merupakan

1.5.15 QueuePlayer Class

```
package util;
```

```

import entity.Player;

public class QueuePlayer {
    NodePlayer front, rear;

    public QueuePlayer () {
        this.front = null;
        this.rear = null;
    }

    public void enqueue(Player nodePlayer) {
        NodePlayer newNode = new NodePlayer(nodePlayer);
        if (front == null) {
            front = rear = newNode;
        } else {
            rear.next = newNode;
            rear = newNode;
        }
    }

    public Player dequeue() {
        if (front == null) {
            return null;
        }
        Player nodePlayer = front.nodePlayer;
        front = front.next;
        return nodePlayer;
    }
}

```

Script di atas merupakan

1.5.16 MainCerita Class

```

package main;

import entity.*;
import util.*;
import util.AlurCerita;

public class MainCerita {
    public static void main(String[] args) {

```

```

        AlurCerita ceritaUtama = new AlurCerita();

        ceritaUtama.createRoot("prolog");

        // Input Ke kiri
        ceritaUtama.inorderInput(ceritaUtama.root, new
        Cerita("act1"), "kiri", "prolog");
        ceritaUtama.inorderInput(ceritaUtama.root, new
        Cerita("act2.1"), "kiri", "act1");
        ceritaUtama.inorderInput(ceritaUtama.root, new
        Cerita("act2.2"), "kanan", "act1");
        ceritaUtama.inorderInput(ceritaUtama.root, new
        Cerita("act3.1"), "kiri", "act2.1");
        ceritaUtama.inorderInput(ceritaUtama.root, new
        Cerita("act4.1"), "kiri", "act3.1");
        ceritaUtama.inorderInput(ceritaUtama.root, new
        Cerita("act4.2"), "kanan", "act3.1");
        ceritaUtama.inorderInput(ceritaUtama.root, new
        Cerita("end1"), "kiri", "act4.1");
        ceritaUtama.inorderInput(ceritaUtama.root, new
        Cerita("end2"), "kanan", "act4.1");

        // Input Ke Kanan
        ceritaUtama.inorderInput(ceritaUtama.root, new
        Cerita("end3"), "kanan", "act3.1");
        ceritaUtama.inorderInput(ceritaUtama.root, new
        Cerita("end4"), "kanan", "act2.1");
        ceritaUtama.inorderInput(ceritaUtama.root, new
        Cerita("end7"), "kanan", "act2.2");
        ceritaUtama.inorderInput(ceritaUtama.root, new
        Cerita("act4.3"), "kiri", "act2.2");
        ceritaUtama.inorderInput(ceritaUtama.root, new
        Cerita("end5"), "kiri", "act4.3");
        ceritaUtama.inorderInput(ceritaUtama.root, new
        Cerita("end6"), "kanan", "act4.3");

        // Input Dialog pada Prolog
        ceritaUtama.root.tambahDialog("[Narator] Di sebuah desa
        yang terletak di lembah hijau, kedamaian terusik oleh teriakan
        dan suara pedang.");

```

```

ceritaUtama.root.tambahDialog("[Narator] Para prajurit
kerajaan utusan dewa menyerbu tanpa ampun, membakar rumah dan
membunuh penduduk tanpa alasan yang jelas.");

ceritaUtama.root.tambahDialog("[Narator] Ananta dan Wadis
datang ketika mendengar kegaduhan tersebut.");

ceritaUtama.root.tambahDialog("[Narator] Dari kejauhan
terlihat Khair yang sedang menyembuhkan korban kegaduhan
tersebut.");

ceritaUtama.root.tambahDialog("[Narator] Khair saat itu
tidak memiliki pelindung atau bahkan teman bersama untuk
mengatasi kegaduhan.");

ceritaUtama.root.tambahDialog("Ananta : 'Sepertinya kita
perlu membantu mereka'");

ceritaUtama.root.tambahDialog("Wadis : 'Ayo!!!'");

ceritaUtama.root.tambahDialog("[Narator] Mereka akhirnya
bekerjasama untuk menaklukan prajurit-prajurit kejam
tersebut.");

// Input Dialog pada Act 1
ceritaUtama.PreOrderInputDialog(ceritaUtama.root, "kiri",
"prolog",
"[Narator] Setelah meninggalkan desa dan mengalahkan
prajurit kerajaan, perjalanan membawa mereka ke hutan lebat
yang penuh dengan hal tak masuk akal."
);

ceritaUtama.PreOrderInputDialog(ceritaUtama.root, "kiri",
"prolog",
"[Narator] Hal tersebut mengatakan bahwa terdapat sebuah
item surgawi tersembunyi yang dapat digunakan untuk
membangkitkan orang yang telah mati."
);

ceritaUtama.PreOrderInputDialog(ceritaUtama.root, "kiri",
"prolog",
"[Narator] Namun tidak pernah ada satupun orang yang
berhasil mendapatkannya, melainkan mereka menghilang tanpa
jejak dan tidak pernah kembali."
);

ceritaUtama.PreOrderInputDialog(ceritaUtama.root, "kiri",
"prolog",

```


"[Narator] Namun, Mereka mengabaikan rumor tersebut. Mereka akhirnya berada di sebuah bangunan tua dan berspekulasi bahwa item surgawi tersebut berada di dalamnya."

);

ceritaUtama.PreOrderInputDialog(ceritaUtama.root, "kiri", "prolog",

"[Narator] Seseorang dengan aura membunuh, berdiri tepat di atas bangunan tersebut dan langsung menyerang Ananta."

);

ceritaUtama.PreOrderInputDialog(ceritaUtama.root, "kiri", "prolog",

"[Narator] Pertempuran tidak dapat dihindari, yang pada akhirnya membuat mereka bertiga harus membentuk strategi dalam mengalahkan orang tak dikenal tersebut."

);

// Input Dialog pada Act 2.1

ceritaUtama.PreOrderInputDialog(ceritaUtama.root, "kiri", "act1",

"Wadis & Khair : 'ANANTA!!!, Kamu tidak apa apa?'"

);

ceritaUtama.PreOrderInputDialog(ceritaUtama.root, "kiri", "act1",

"[Narator] Ananta sempat jatuh dan kehilangan kesadaran ketika sebuah cahaya dan kabut yang keluar dari tubuh orang tersebut mendekati mendekati dirinya."

);

ceritaUtama.PreOrderInputDialog(ceritaUtama.root, "kiri", "act1",

"Ananta : 'HAAAAA!!!'. Ananta terdengar terengah engah ketika sadar."

);

ceritaUtama.PreOrderInputDialog(ceritaUtama.root, "kiri", "act1",

"[Narator] Wadis dan Khair pun menjadi sedikit tenang dari kekhawatirannya tersebut, karena Ananta terlihat baik-baik saja."

);

ceritaUtama.PreOrderInputDialog(ceritaUtama.root, "kiri", "act1",

```

    "[Narator] Namun, Wadis menaruh curiga yang sangat mendalam
ketika melihat jasad manusia yang hendak membunuh mereka
bertiga ternyata merupakan teman kampungnya yaitu Fakhroni."
);
    ceritaUtama.PreOrderInputDialog(ceritaUtama.root, "kiri",
"act1",
    "[Narator] Tanpa memperdulikan kejadian tersebut mereka
bertiga tetap melanjutkan perjalanan menembus hutan luas
tersebut, dan akhirnya mereka sampai pada sebuah portal."
);
    ceritaUtama.PreOrderInputDialog(ceritaUtama.root, "kiri",
"act1",
    "[Narator] Mereka masuk ke dalam portal dan berjalan hingga
sampai di suatu tempat yang terlihat tidak seperti bumi."
);
    ceritaUtama.PreOrderInputDialog(ceritaUtama.root, "kiri",
"act1",
    "[Narator] Sampai suatu ketika satu serangan yang sangat
cepat datang menghampiri mereka dan melukai Wadis."
);
    ceritaUtama.PreOrderInputDialog(ceritaUtama.root, "kiri",
"act1",
    "Wadis : 'ITU XENOMORPH!!!, HATI HATI DENGAN SERANGANNYA!'"
);
    ceritaUtama.PreOrderInputDialog(ceritaUtama.root, "kiri",
"act1",
    "[Narator] Mereka dipaksa bertempur lagi."
);

    // Input Dialog pada Act 2.2
    ceritaUtama.PreOrderInputDialog(ceritaUtama.root,
"kanan", "act1",
    "[Narator] Mereka bertiga terluka parah setelah dikalahkan
oleh manusia tidak dikenal tersebut dan berhasil melarikan diri
ke alam monster."
);
    ceritaUtama.PreOrderInputDialog(ceritaUtama.root,
"kanan", "act1",
    "[Narator] Di alam ini mereka dipertemukan oleh monster
Carberus yang merupakan monster penjaga gerbang menuju alam
dewa."

```

```

    );
    ceritaUtama.PreOrderInputDialog(ceritaUtama.root,
    "kanan", "act1",
    "[Narator] Mereka harus mengalahkan Carberus walau hidup
atau mati! demi keberlangsungan hidup manusia"
    );
    ceritaUtama.PreOrderInputDialog(ceritaUtama.root,
    "kanan", "act1",
    "Khair : 'Sepertinya Carberus tidak memiliki nafsu
membunuh, namun kita harus berhati hati jika suatu ketika
Carberus berubah menjadi murka'"
    );
    ceritaUtama.PreOrderInputDialog(ceritaUtama.root,
    "kanan", "act1",
    "[Narator] Mereka pun langsung melontarkan serangan kepada
Carberus dan pertempuran tak dapat dihindari!."
    );
    ceritaUtama.PreOrderInputDialog(ceritaUtama.root,
    "kanan", "act1",
    "[Narator] Mereka pun langsung melontarkan serangan kepada
Carberus dan pertempuran tak dapat dihindari!."
    );

    // Input Dialog pada Act 3.1
    ceritaUtama.PreOrderInputDialog(ceritaUtama.root, "kiri",
    "2.1",
    "[Narator] Mengalahkan Xenomorph bukanlah hal yang mudah,
karena mereka harus kehilangan banyak darah dan energi mereka."
    );
    ceritaUtama.PreOrderInputDialog(ceritaUtama.root, "kiri",
    "2.1",
    "[Narator] Ananta, Wadis dan Khair menyempatkan diri mereka
untuk beristirahat sejenak untuk memulihkan tenaga mereka
sembari berbincang rencana selanjutnya."
    );
    ceritaUtama.PreOrderInputDialog(ceritaUtama.root, "kiri",
    "2.1",
    "[Narator] Mereka beranjak dan terus menelusuri alam
monster, hingga mereka sampai disebuah portal yang menghubungi
dunia monster dengan dunia para roh."
    );

```

```

        ceritaUtama.PreOrderInputDialog(ceritaUtama.root, "kiri",
"2.1",
        "[Narator] Tanpa pikir panjang mereka pun langsung
menyusuri portal tersebut hingga sampai di suatu tempat yang
benar benar datar, suasana di tempat itu benar benar membuat
jiwa benar benar termenung."
        );
        ceritaUtama.PreOrderInputDialog(ceritaUtama.root, "kiri",
"2.1",
        "[Narator] Di kala suasana yang hening tersebut tiba tiba
Ananta jatuh dan mengalami kejang kejang."
        );
        ceritaUtama.PreOrderInputDialog(ceritaUtama.root, "kiri",
"2.1",
        "Wadis : 'ANANTA?! Kamu kenapa lagi?'"
        );
        ceritaUtama.PreOrderInputDialog(ceritaUtama.root, "kiri",
"2.1",
        "[Narator] Tiba tiba ananta menyerang mereka berdua,
terlihat sifat Ananta ini sama dengan Fakhroni. Sehingga Wadis
mengerti apa yang sebenarnya terjadi."
        );
        ceritaUtama.PreOrderInputDialog(ceritaUtama.root, "kiri",
"2.1",
        "[Narator] Mereka saat ini sedang melawan Roh Umbra yang
dikenal akan kekuatannya merasuki tubuh manusia! Mereka berdua
harus mengalahkan Ananta yang sedang dirasuki kekuatan super
dahsyat."
        );

        // Start
        ceritaUtama.nextAct();
        // End

    }
}

```

Script di atas merupakan