



**UNIVERSITAS INDONESIA**

**DESAIN SISTEM PERMAINAN - CONSOLE RPG**

**Kelompok 7**

<b>Javana Dzaki Muhammad</b>	<b>(2306161826)</b>
<b>M Arya Wiandra Utomo</b>	<b>(2306218295)</b>
<b>Nugroho Ulil Abshar</b>	<b>(2306229310)</b>
<b>Rivi Yasha Hafizhan</b>	<b>(2306250535)</b>

**Teknik Komputer**  
**Universitas Indonesia**  
**Depok**  
**2024**

## 1. Deskripsi Singkat

*Solluminaty* adalah permainan RPG berbasis teks di mana pemain berperan sebagai pahlawan dalam dunia fiksi yang penuh dengan musuh dan tantangan. Pemain mengontrol karakter utama dengan statistik seperti kesehatan, kekuatan serangan, dan level. Dalam permainan ini, hanya ada satu instance karakter utama yang dapat dikembangkan secara berkelanjutan, mengikuti pola desain Singleton. Pemain dapat bertempur melawan berbagai jenis musuh yang diciptakan menggunakan kelas pabrik, seperti Goblin, Orc, dan Naga, dengan tiap musuh memiliki atribut kesehatan dan kekuatan serangan yang berbeda.

Pertarungan dalam *Solluminaty* berlangsung secara bergiliran, dengan setiap karakter memiliki kesempatan menyerang. Jika berhasil mengalahkan musuh, pemain akan memperoleh peningkatan level, memperkuat karakter utama untuk menghadapi tantangan berikutnya. Game ini menyajikan elemen RPG klasik yang memungkinkan pemain untuk berkembang secara progresif, membangun karakter pahlawan dari pemula hingga pejuang yang tangguh melalui skenario pertarungan berbasis teks.

## 2. Kelas dan Struktur Program

Program ini dibangun dalam paket game dengan beberapa kelas penting yang mencakup entitas dasar, karakter utama, musuh, dan inventory:

- **Character:** Kelas utama yang mewakili karakter pemain.
- **Enemy:** Subkelas dari Entity yang mewakili berbagai tipe musuh.
- **Inventory:** Kelas yang mewakili sistem inventaris pemain.
- **EnemyFactory:** Kelas pabrik yang digunakan untuk menghasilkan musuh tertentu berdasarkan tipe.
- **Entity:** Kelas abstrak yang mendefinisikan atribut dan metode umum untuk karakter dan musuh.

## 3. Latar Belakang Cerita

Dalam permainan ini, pemain memulai sebagai pahlawan pemula yang terjun ke dunia penuh tantangan, di mana mereka harus mengalahkan musuh yang semakin kuat. Pemain

mengembangkan keterampilan melalui pertempuran, mengumpulkan item, dan naik level, yang memungkinkan mereka untuk menghadapi musuh yang lebih kuat.

## 4. Gameplay Inti

### 1. **Pertarungan:**

- Pemain dapat menyerang musuh, dan sebaliknya, musuh juga menyerang pemain.
- Pertarungan berlangsung sampai salah satu pihak kalah, yang ditentukan oleh nilai kesehatan (health).

### 2. **Level-Up:**

- Setelah mencapai kriteria tertentu, karakter pemain dapat naik level yang meningkatkan kesehatan dan kekuatan serangan.

### 3. **Inventaris:**

- Pemain memiliki inventaris untuk menyimpan item, meskipun implementasinya belum dirinci dalam kode.

## 5. Fitur Fondasi

- **Sistem Singleton** pada Kelas Character: Karakter utama diimplementasikan menggunakan pola desain Singleton, sehingga hanya ada satu instance dari karakter pemain selama permainan.
- **Factory Pattern untuk Enemy:** Kelas EnemyFactory memungkinkan pembuatan berbagai tipe musuh seperti Goblin, Orc, dan Naga.

## 6. Mekanika Permainan

### ● **Statistik Karakter:**

- **Kesehatan:** Berkurang selama pertempuran dan meningkat saat naik level.
- **Kekuatan Serangan:** Menentukan jumlah kerusakan yang diberikan selama pertempuran.

- **Sistem Level-Up:** Setiap kenaikan level meningkatkan kesehatan dan kekuatan serangan karakter pemain.

## 7. Antarmuka Pengguna

Antarmuka permainan berbasis teks, di mana setiap aksi, seperti menyerang atau naik level, ditampilkan dalam bentuk keluaran teks di konsol.

- **Output Pertarungan:** Setiap serangan menunjukkan kerusakan yang diterima oleh musuh atau pemain, dan hasil pertarungan ditampilkan dengan teks seperti "has been defeated!" atau "health remaining".

## 8. Pola Desain

- **Singleton Pattern** pada kelas Character untuk memastikan hanya ada satu karakter pemain.
- **Factory Pattern** pada EnemyFactory, memungkinkan kemudahan pembuatan objek musuh dengan tipe berbeda tanpa perlu menambah kode pada kelas utama.

## 9. Pengembangan Lanjutan

Beberapa fitur yang dapat dikembangkan lebih lanjut meliputi:

- **Sistem Inventaris:** Penambahan metode untuk mengelola item dalam kelas Inventory.
- **Quest atau Misi:** Penambahan sistem misi atau tantangan tambahan yang dapat memberikan imbalan khusus atau peningkatan level.