Laporan Tugas Besar 3 IF 2210

Aplikasi Permainan Dimlysouls



Dipersiapkan oleh :

Kelompok 33 – Re: Master Game Dev

Adrian Mulyana N 13515075

Vincent Hendryanto Halim 13515089

Mikhael Artur Darmakesuma 13515099

Alif Ijlal Wafi 13515122

**Program Studi Teknik Informatika**

**Sekolah Teknik Elektro dan Informatika**

**Institut Teknologi Bandung**

**2017**

# Daftar Isi

[Daftar Isi 2](#_Toc480989576)

[1 Deskripsi Aplikasi 3](#_Toc480989577)

[1.1 Deskripsi Keseluruhan Aplikasi 3](#_Toc480989578)

[1.2 Deskripsi Fitur 3](#_Toc480989579)

[1.3 Deskripsi Perilaku Umum 3](#_Toc480989580)

[2 Rancangan Aplikasi 4](#_Toc480989581)

[2.1 Diagram Package 4](#_Toc480989582)

[2.2 Diagram Kelas 5](#_Toc480989583)

[2.2.1 Rancangan Diagram Kelas Awal 5](#_Toc480989584)

[2.2.2 Diagram Kelas Aplikasi 6](#_Toc480989585)

[2.3 Diagram Kolaborasi antar Kelas 7](#_Toc480989586)

[3 Deskripsi Algoritma 7](#_Toc480989587)

[3.1 Algoritma Generasi Peta 7](#_Toc480989588)

[4 Test Skenario 7](#_Toc480989589)

[5 Ringkasan Hasil Metriks 8](#_Toc480989590)

# Deskripsi Aplikasi

## Deskripsi Keseluruhan Aplikasi

Aplikasi yang dibuat merupakan sebuah aplikasi permainan dengan konsep game Endless 2D RPG. Pada permainan ini, pemain dapat memilih class yang akan digunakan untuk bermain. Tiap class memiliki skill spesial dan *stats* yang berbeda.

Dalam permainan pemain akan melalui peta yang digenerasi secara otomatis. Pemain dapat bertarung melawan musuh dan mendapatkan *experience*. Setiap map yang dilewati akan memperkuat musuh yang dilawan. *Experience* yang didapat dari mengalahkan musuh akan berguna untuk meningkatkan level serta kekuatan karakter.

## Deskripsi Fitur

Pada permainan ini, terdapat fitur-fitur sebagai berikut :

1. Menampilkan Main Menu
2. Memilih class dan membuat Player sesuai class
3. Menampilkan Stats dari Player
4. Menampilkan peta dan isinya
5. Menampilkan Interface Battle
6. Player dapat bergerak di peta
7. Player dapat menabrak musuh dan memulai battle
8. Player dapat berpindah map dan kembali lagi
9. Player dapat mengambil item
10. Player bisa melakukan level up
11. Musuh dapat bergerak

## Deskripsi Perilaku Umum

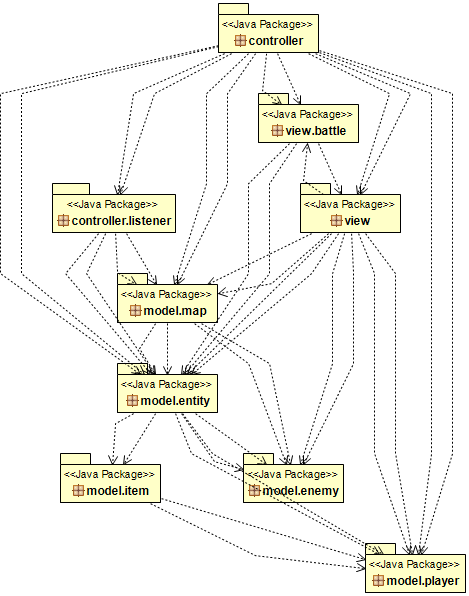
Ketika aplikasi dijalankan, aplikasi akan menawarkan pemain untuk memulai permainan dengan memilih *class* dari 4 pilihan class yang tersedia. Setelah itu aplikasi akan melakukan pembuatan map secara acak. Pemain akan dibebaskan untuk menjelajahi peta yang tersedia. Jika peta belum tersedia, *game* akan melakukan pembuatan peta baru.

Jika player bertemu dengan musuh maka layar akan berubah menjadi layar untuk battle dimana player dapat menyerang musuh.

# Rancangan Aplikasi

## Diagram Package

Diagram package yang terdapat pada aplikasi adalah sebagai berikut :

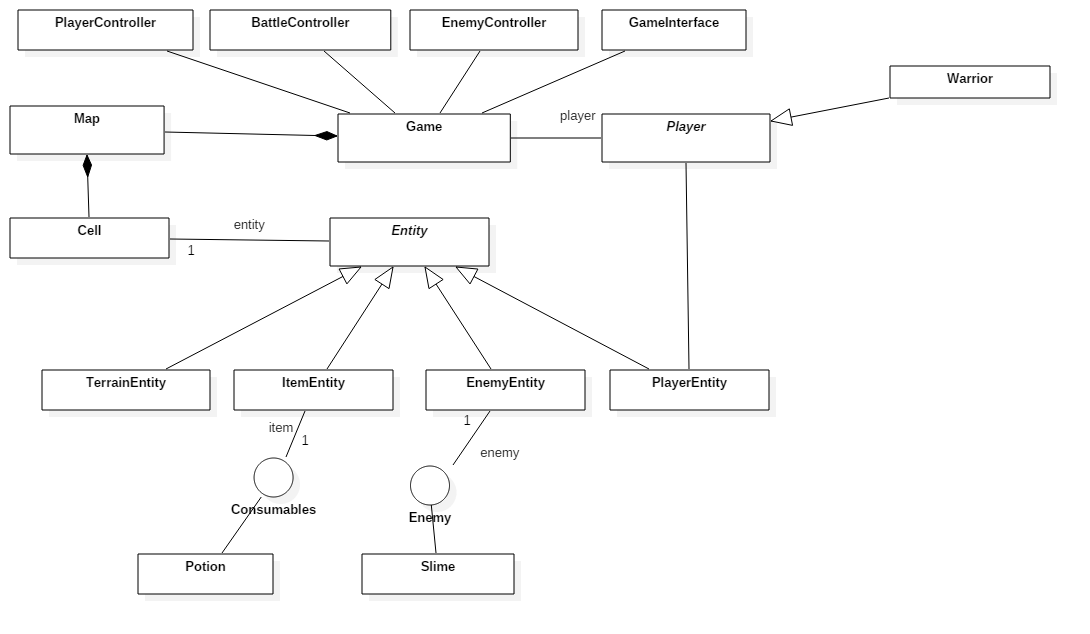


Gambar 1. Diagram Package Aplikasi

## Diagram Kelas

### Rancangan Diagram Kelas Awal

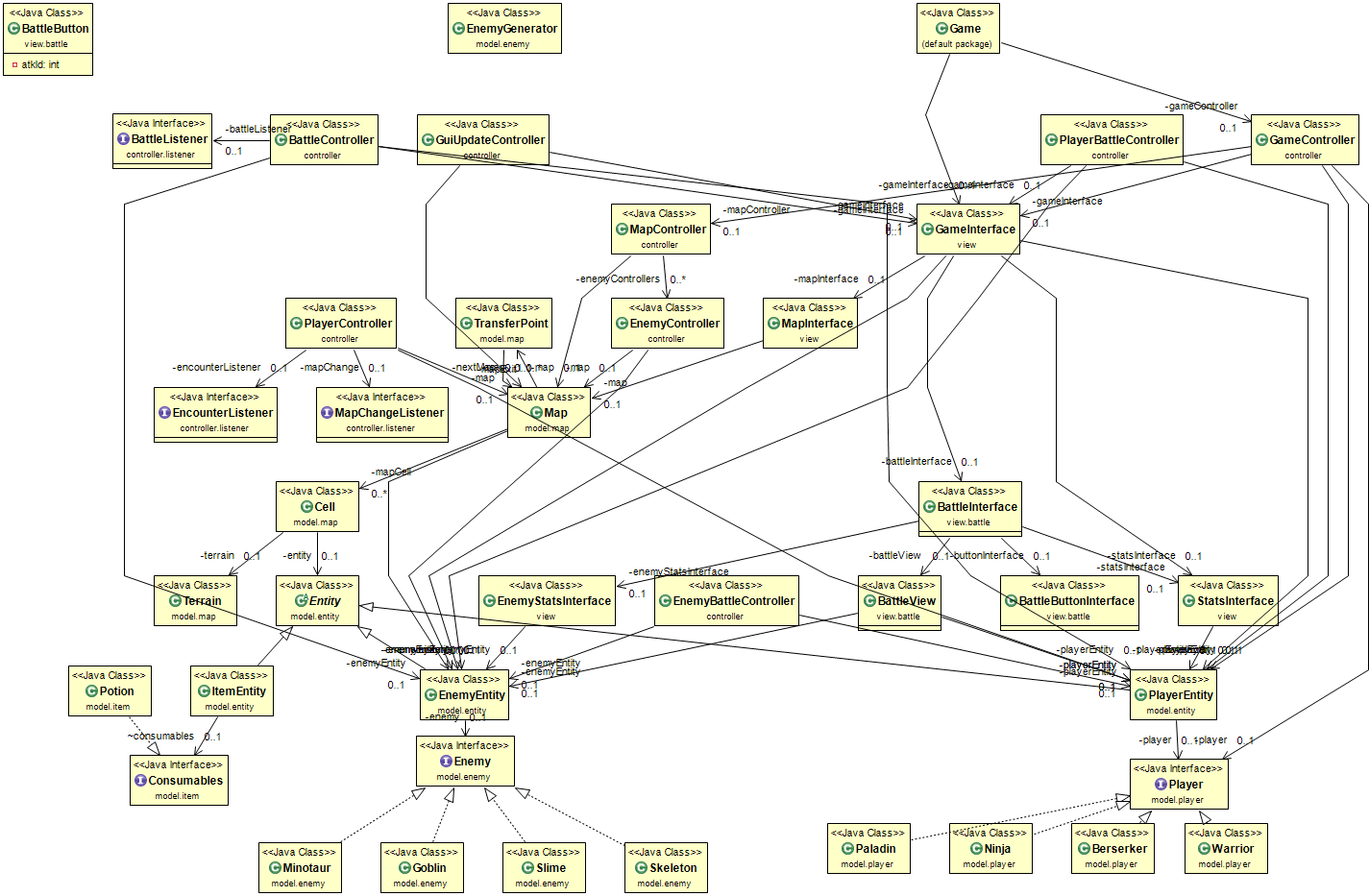
Pada awal pembuatan aplikasi, dirancang diagram kelas sebagai berikut :



Gambar 2. Rancangan Diagram Kelas Awal

Pada rancangan di atas, terdapat sebuah objek Game yang memiliki komponen berupa map. Map tersebut terdiri dari kumpulan cell yang disusun dalam matriks. Setiap cell memiliki sebuah entity. Entity merupakan objek yang ditampilkan di map dan dapat berupa TerrainEntity, ItemEntity, EnemyEntity, atau PlayerEntity. Data dari Entity tersebut disimpan dalam nama kelas atau implementasi interface sesuai nama entity tersebut. Pada diagram ini, Controller dan Interface masih belum dikembangkan lebih lanjut.

### Diagram Kelas Aplikasi

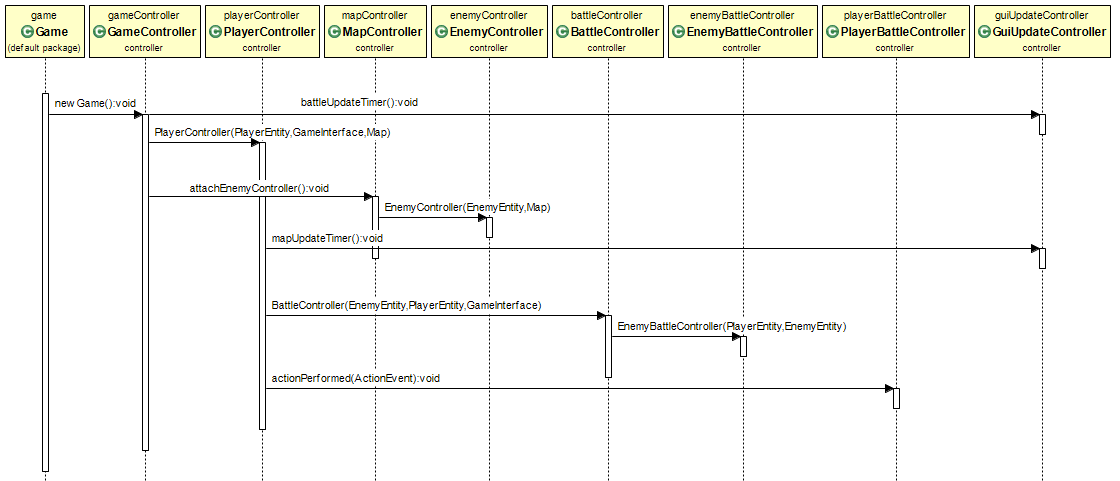
Diagram kelas aplikasi yang digunakan adalah sebagai berikut :

Gambar 3. Diagram Kelas Aplikasi

Pada diagram kelas di atas, perubahan yang terjadi dari rancangan diagram kelas awal adalah diubahnya player menjadi interface, jika kedepannya aplikasi akan ditambahkan *class* Player lain. Selain itu, jumlah controller diperbanyak untuk melakukan kontrol terhadap hal-hal lain secara mendetail. Terdapat juga *class* yang termasuk dalam golongan *class* listener (yang berupa interface) untuk mengirimkan pesan antar class

## Diagram Kolaborasi antar Kelas

Diagram kolaborasi antar kelas dari aplikasi adalah sebagai berikut :



Gambar 4. Diagram Kolaborasi antar Kelas

# Deskripsi Algoritma

## Algoritma Generasi Peta

Pada permainan ini, digunakan algoritma untuk melakukan generasi peta secara acak. Algoritma ini akan memilih 1 titik pada ujung peda dan kemudian memilih titik selanjutnya secara acak. Untuk membuat percabangan, digunakan *queue* yang mengakibatkan beberapa titik akan *digenerate* secara bergantian.

Apabila titik sudah mencapai jalan keluar atau titik sudah tidak dapat dilanjutkan (jalan buntu) titik akan dibuang dari *queue*. Apabila *queue* sudah habis, maka algoritma pencarian akan dihentikan

# Test Skenario

|  |  |  |  |
| --- | --- | --- | --- |
| No | Skenario | Hasil  (OK/NO) | Keterangan |
| **User Interface** | | | |
| 1 | Menampilkan Main Menu | OK |  |
| 2 | Memilih class dan membuat Player sesuai class | OK |  |
| 3 | Menampilkan Stats dari Player | OK |  |
| 4 | Menampilkan peta dan isinya | OK |  |
| 5 | Menampilkan Interface Battle | OK |  |
| **Fungsional** | | | |
| 1 | Player dapat bergerak di peta | OK |  |
| 2 | Player dapat menabrak musuh dan memulai battle | OK |  |
| 3 | Player dapat berpindah map dan kembali lagi | OK |  |
| 4 | Player dapat mengambil item | OK |  |
| 5 | Player bisa melakukan level up | OK |  |
| 6 | Musuh dapat bergerak | OK |  |

# Ringkasan Hasil Metriks

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Package | Concrete Class | Abstract Class | Ca | Ce | A | I | D |
| Default | 1 | 0 | 0 | 3 | 0 | 1 | 0 |
| controller | 15 | 0 | 1 | 13 | 0 | 0,93 | 0,07 |
| controller.listener | 0 | 3 | 1 | 3 | 1 | 0,75 | 0,75 |
| model.enemy | 6 | 1 | 4 | 2 | 0,17 | 0,33 | 0,5 |
| model.entity | 4 | 1 | 5 | 5 | 0,25 | 0,5 | 0,25 |
| model.item | 3 | 1 | 2 | 3 | 0,33 | 0,6 | 0,07 |
| model.map | 4 | 0 | 3 | 6 | 0 | 0,67 | 0,33 |
| model.player | 5 | 1 | 4 | 1 | 0,2 | 0,2 | 0,6 |
| view | 4 | 0 | 3 | 10 | 0 | 0,77 | 0,23 |
| view.battle | 4 | 0 | 2 | 6 | 0 | 0,75 | 0,25 |