**Laporan Tugas Besar**

**Atomaders**

**Pemrograman Berorientasi Objek**

oleh :

**DIONISIUS PRATAMA / 1117002**

**JESSICA CECILIA / 1117011**

**ANDREAS CHRISTIAN / 1117035**



**PROGRAM STUDI INFORMATIKA**

**INSTITUT TEKNOLOGI HARAPAN BANGSA**

**2018**

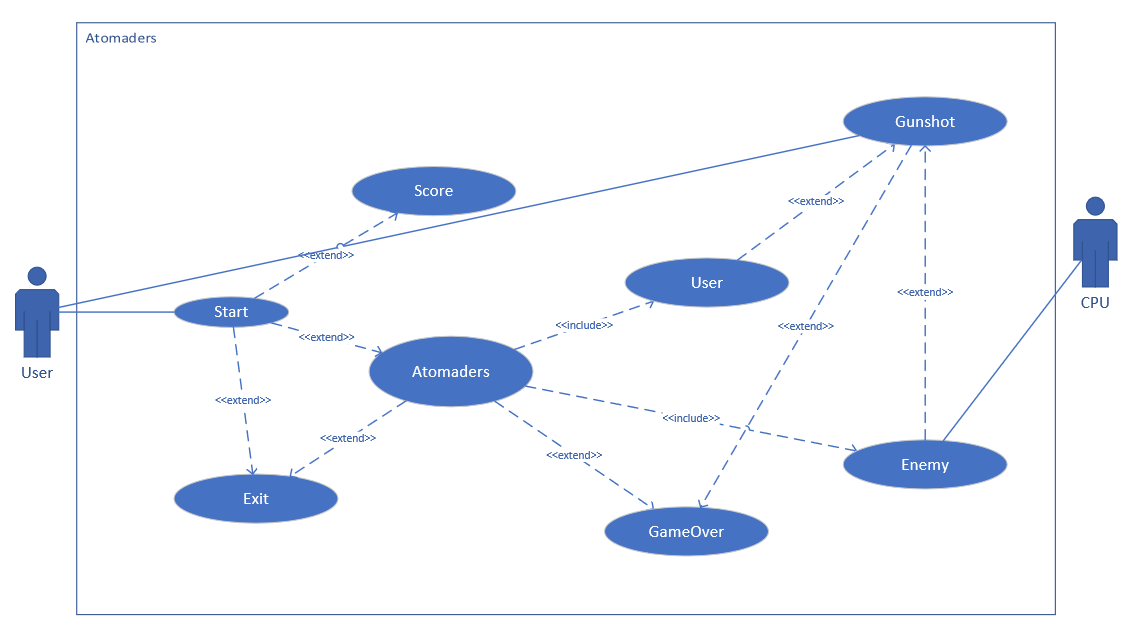
# Deskripsi Proyek

Judul tugas besar PBO kelompok 3 adalah Atomaders. Proyek ini merupakan sebuah game (permainan) sederhana yang tujuannya adalah menembak semua musuh yang ada di layer. Jika semua musuh telah ditembak, maka player sudah memenangkan game.

Selain memiliki fitur permainan, proyek ini juga memakai sedikit konsep database. Database digunakan dalam menampung score para player yang sudah bermain di waktu sebelumnya.

# Use Case Diagram

berikut adalah Use Case Diagram dari sistem ini.

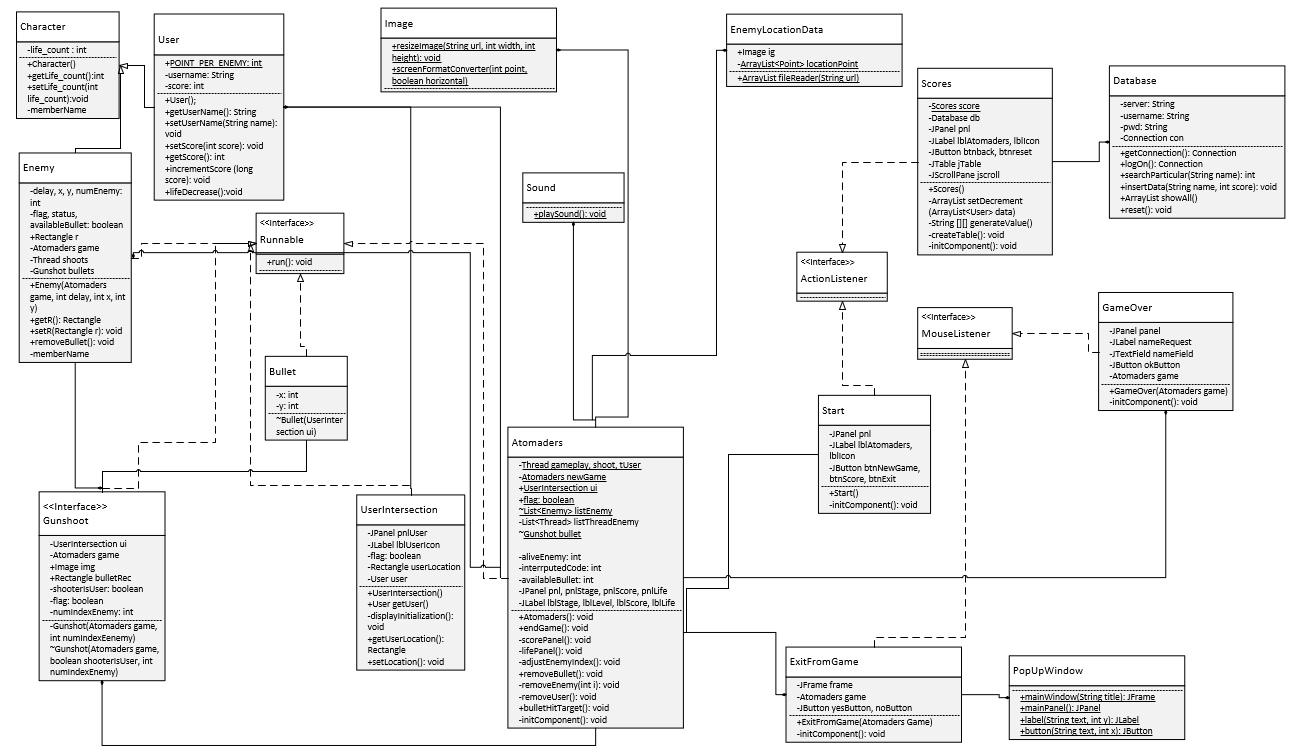


Dari Use Case Diagram sebelumnya, berikut adalah penjelasan setiap fungsi yang ada dan ditampilkan dalam bentuk tabel.

|  |  |
| --- | --- |
| ***Use Case*** | ***Deskripsi*** |
| Start | Kelas Start.java memiliki 3 button, masing-masing akan trigger class Score.java (melihat dan mereset Score), Atomaders.java (permainan utama), dan exit (keluar dari game) |
| Score | Melihat scores dan mereset, data disimpan di dalam sebuah database. |
| Exit | Mengeluarkan user dari game |
| Atomaders | Kelas yang memuat permainan utama, berisi user dan enemy yang masing-masing bisa saling menembak (dijelaskan lebih lanjut dalam user manual) |
| GameOver | Mengeluarkan user dari game karena user telah kalah (life sudah habis) |
| Enemy | Kelas yang memuat enemy yang muncul pada game |
| User | Kelas yang memuat user, user dapat menggerakan objek |
| Gunshot | Untuk menembak |

Class Diagram

Berdasarkan kebutuhan fungsi dalam game Atomaders, berikut adalah hasil analisa kelas – kelas yang dibutuhkan dalam game. Kelas akan ditampilkan dalam bentuk class diagram di bawah ini:



Adapun penjelasan tiap kelas – kelas tersebut adalah sebagai berikut.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Deskripsi** | **Modifier** | **Tipe** | **Atribut** |
| Start | Memuat 3 button:   1. Mengarahkan player untuk memulai new game 2. Mengarahkan player untuk melihat scores (dari database) 3. Mengarahkan player untuk keluar dari game | private | JPanel | pnl |
|  | private | JLabel | lblAtomaders |
| private | JLabel | lblIcon |
| private | JButton | btnNewGame |
| private | JButton | btnScore |
| private | JButton | btnExit |
| Atomaders | Kelas utama   1. Disimpan Thread (rangkaian eksekusi dari sebuah aplikasi) untuk menggerakan player, enemy dan shooting  * Enemy disimpan dalam sebuah list, begitu juga dengan threadnya * Objek untuk menghidupkan kelas gunshot (untuk menembak) dan player disimpan disini  1. Game dimainkan dalam kelas ini | private | JPanel | pnl |
|  | private | JPanel | pnlStage |
| private | JPanel | pnlStage2 |
| private | JPanel | pnlScore |
| private | JPanel | pnlLife |
| private | JLabel | lblStage |
| private | JLabel | lblUser |
| private | JLabel | lblLevel |
| private | JLabel | lblScore |
| private | JLabel | lblLife |
| private | Tread | shoot |
| - | Gunshoot | bullet |
| public | UserIntersection | ui |
| private | Thread | tUser |
| private | List<Enemy> | listEnemy |
| private | List<Tread> | listThreadEnemy |
| private | int | aliveEnemy |
| private | int | interruptedCode |
| private | boolean | availableBullet |
| Bullet | Untuk meletakkan peluru tepat di atas object player | private | Image | Img |
|  | private | int | x |
| private | int | y |
| Character | Menyimpan atribut umum yang dimiliki karakter dalam game | private | int | life\_count |
| UserIntersection | Menyimpan thread yang ada di dalam kelas User, menginisialisasi user | private | Rectangle | userLocation |
|  | private | User | user |
| private | boolean | flag |
| private | JPanel | pnlUser |
| private | JLabel | lblUserIcon |
| Enemy | Inisialisasi Thread enemy | private | int | delay |
|  | private | boolean | flag |
| private | boolean | status |
| private | int | x |
| private | int | y |
| private | Rectangle | r |
| public | int | numEnemy |
| private | Atomaders | game |
| private | Tread | shoots |
| private | Gunshoot | bullets |
| private | boolean | avaiableBullet |
| ExitFromGame | Mengeluarkan player dari game saat player sedang memainkan game | private | JFrame | frame |
|  | private | Atomaders | game |
| private | JButton | yesButton |
| private | JButton | noButton |
| Gunshot | Setting posisi peluru, menginisialisasi peluru yang dimiliki oleh player dan enemy | private | Atomaders | game |
|  | public | Image | img |
| public | Rectangle | bulletRec |
| private | UserInsection | ui |
| private | boolean | shooterIsUser |
| private | boolean | flag |
| private | int | numIndexEnemy |
| GameOver | Kelas yang terpanggil saat user kalah di dalam game atau memilih keluar di tengah permainan | private | JPanel | panel |
|  | private | JLabel | nameRequest |
| private | JTextField | nameField |
| private | JButton | okButton |
| private | Atomaders | game |
| Scores | Menampung scores para player yang telah memainkan game, memanfaatkan database | private | Scores | score |
|  | private | JPanel | pnl |
| private | Jlabel | lblAtomaders |
| private | Jlabel | lblIcon |
| private | JButton | btnback |
| private | JButton | btnreset |
| private | Database | db |
| private | JTabel | jTabel |
| private | JScrollPane | jScroll |
| - | String[] | colName |
| User | Inisialisasi username, score, dan poin yang diperoleh user saat user berhasil menembak enemy | public | final int | POINT\_PER\_ENEMY |
|  | private | String | username |
| private | int | score |
| EnemyLocationData | Menyimpan lokasi enemy, implementasinya dalam file | public | Image | img |
|  | private | ArrayList<Point> | locationPoint |
| Image | Memuat void (prosedur) resizeImage | - | - | - |
| Sound | Memuat void (prosedur) untuk memainkan lagu | - | - | - |
| Database | Dalam proyek ini, kelas Database digunakan untuk menampung score para player saja | private | String | server |
|  | private | String | username |
| private | String | pwd |
| private | Connection | Con |
| PopUpWindow | Dipakai untuk layout dalam class ExitFromGame | - | - | - |

# Implementasi Program

1. **Start**



Ini adalah start screen saat pertama kali game dibuka (game dibuka pada class Start.java). Ada 3 button yang akan muncul disana, yaitu:

1. New Game : memulai permainan baru (Player akan dibawa pada class Atomaders.java)
2. Scores : melihat score para pemain sebelumnya (Player akan dibawa pada class

Scores.java)

1. Exit : player akan dibawa keluar dari game
2. **New Game (memulai permainan)**



3

2

1

4

1. **Stage**

Informasi stage mana dan level berapa. Untuk saat ini, baru dikembangkan 1 level saja.

1. **Score**

Player akan memperoleh score saat player sukses menembak musuh yang ada. 1 musuh bernilai 150 poin.

1. **Life**

Saat memulai permainan, player akan menerima “nyawa” sebanyak 3. Bila musuh menembak Player, maka nyawa Player akan berkurang 1. Bila nyawa Player sudah mencapai angka 0, permainan selesai.

1. **User plane**

Ini adalah spaceship yang dapat player kendalikan untuk menghancurkan musuh.

1. Pergerakan ke kiri dan ke kanan saja. Pergerakan menggunakan arrow keys yang ada pada keyboard player.
2. Untuk menembak musuh, Player dapat menekan “space” pada keyboard. Dalam 1 waktu, hanya dapat dilakukan 1 tembakan saja.

Peluru yang dapat dikeluarkan user

1. **Game Over**



2

1

Permainan dinyatakan selesai bila nyawa Player sudah habis (life: 0) [lihat kotak 2]. Saat game berakhir, player akan diminta mengisikan nama [lihat kotak 1] untuk nantinya diinput ke table Scores pada class Scores.java.

1. **Escape from Game**



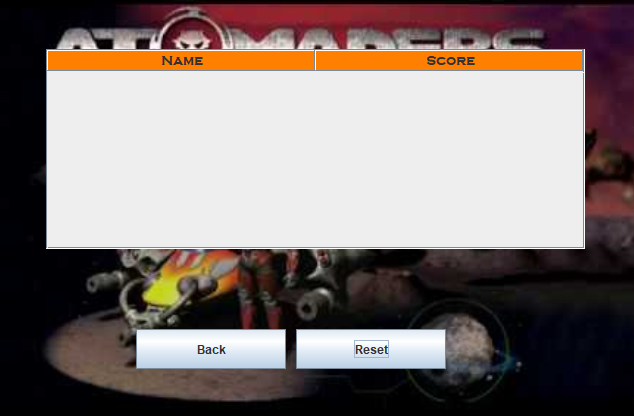
1

Jika player ingin keluar di tengah waktu bermain, player cukup menekan tombol esc, dan game akan berakhir bila player menekan YES dengan mouse [lihat kotak 1]. Setelahnya, player akan diminta lagi untuk menginput nama untuk dimasukkan ke dalam table score.

1. **Scores**



Pada menu ini, player dapat melihat semua score dari permainan-permainan sebelumnya. Bila player menekan tombol reset, maka data dalam table score semuanya akan hilang seperti berikut:



Bila player menekan tombol back, maka player akan dibawa kembali menuju start screen.