COURSE OUTLINE PEMROGRAMAN BERORIENTASI OBJEK

Oleh: Abas Setiawan

Praktikum 3, 4, & 5

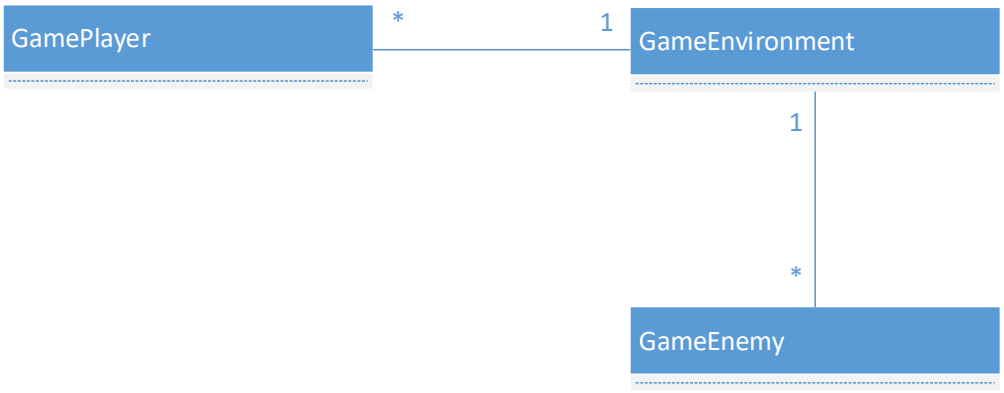
Contents (DAFTAR ISI)

Petunjuk Praktikum 1
GamePlayer Class 2
GameEnemy Class 3
GameEnvironment Class 3
Testing Class 4
REFERENSI 5
CHECKLIST PERTEMUAN 1 5
CHECKLIST PERTEMUAN 2 5
CHECKLIST PERTEMUAN 3 6
REKAPITULASI TUGAS 1 6

Petunjuk Praktikum

Pada praktikum 3 dan 4 kali ini, sedikit berbeda dengan praktikum sebelumnya. Mahasiswa diharapkan sudah mampu memahami secara konsep tentang Class, Object, dan Keterhubungan antar Object.

Terdapat 1 Class utama diberi nama GameEnvironment. Class GameEnvirontment akan berasosiasi dengan 2 Class lainya yaitu class GamePlayer dan GameEnemy. Class Berikut adalah keterkaitan asosiasi diantaranya yang digambarkan dengan UML Class diagram sebagai berikut:



Penjelasan:

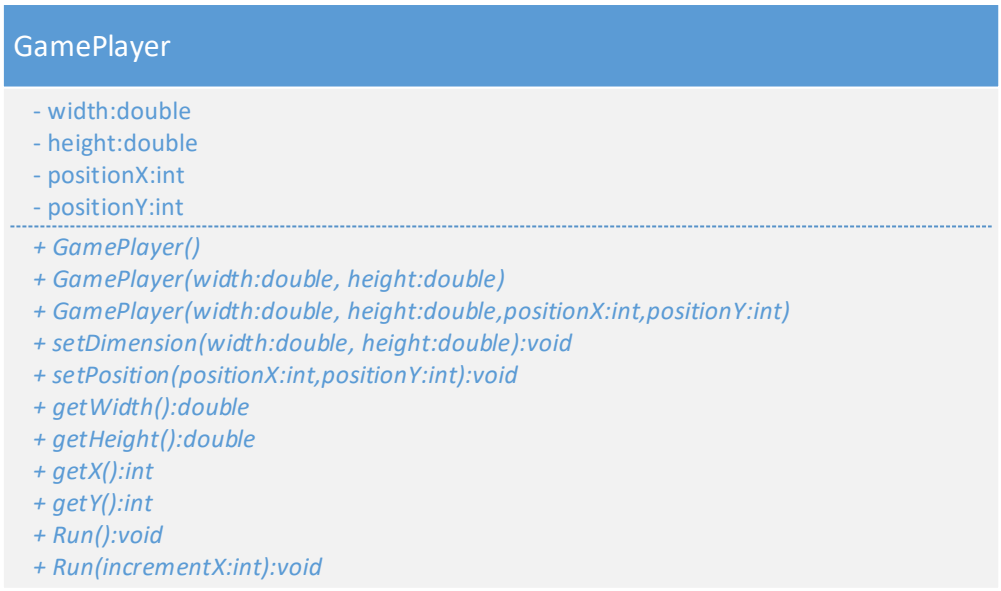
- Satu GameEnvirontment dapat memiliki objek-objek game seperti GamePlayer dan GameEnemy.
- Detail setiap Class akan dijelaskan pada bab-bab berikutnya.

CATATAN PENTING:

- Pertemuan ke-3 mengerjakan GamePlayer atau GameEnemy dahulu.
- Pertemuan ke-4 mengerjakan GameEnvirontment.
- Pertemuan ke-5 mengerjakan asosiasi dan agregrasi dengan memanfaatkan Array dari Objek.
- Silahkan buka buku "1300"-an halaman dan slide presentasi sebagai pedoman untuk mengerjakan praktikum ini.

GamePlayer Class

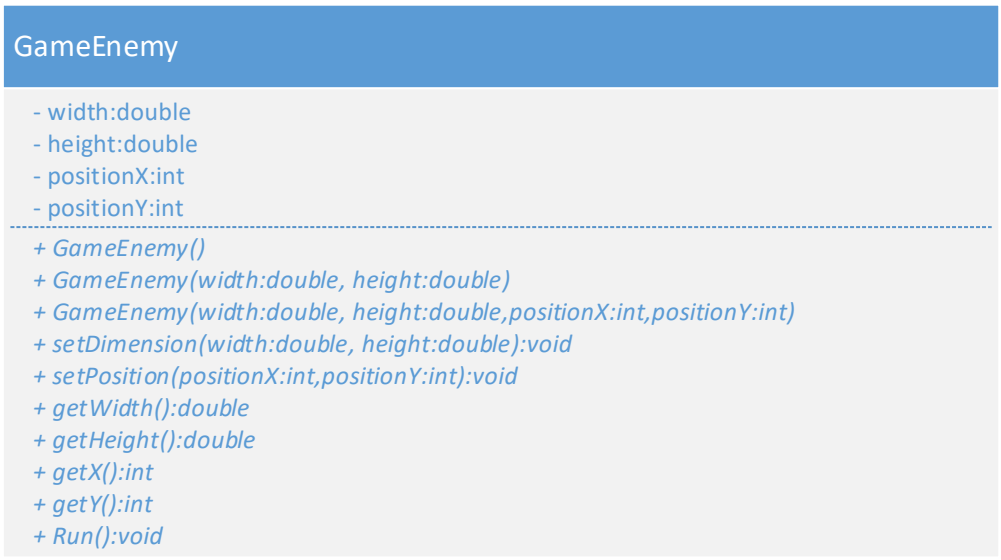
GamePlayer Class merupakan kelas pembentuk objek karakter player 2D. Karakter player 2D biasanya memiliki panjang/width dan leba/height dan berada pada suatu titik koordinat posisi x dan y. Berikut adalah gambar Class diagram dari GamePlayer.java.



- Method setDimension merupakan setter untuk property width dan height.
- Method setPosition merupakan setter untuk property positionX dan positionY.
- Method getWidth, getHeight,getX, dan getY merupakan getter untuk property width,height, positionX, dan positionY.
- Method Run() berisi tulisan "Player is running "
- Method Run(inceremenX) berisi penambahan nilai posisi sebelumnya ditambah dengan increment dan men-output-kan tulisan "Player still running...current X position= 90" jika inceremenX = 30 dan positionX sebelumnya adalah 60.

GameEnemy Class

GameEnemy Class merupakan kelas pembentuk objek karakter musuh 2D. Karakter musuh 2D biasanya memiliki panjang/width dan leba/height dan berada pada suatu titik koordinat posisi x dan y. Berikut adalah gambar Class diagram dari GameEnemy.java.



- Method setDimension merupakan setter untuk property width dan height.
- Method setPosition merupakan setter untuk property positionX dan positionY.
- Method getWidth, getHeight, getX, dan getY merupakan getter untuk property width,height, positionX, dan positionY.
- Method Run() berisi tulisan “Enemy is running "

GameEnvironment Class

GameEnvironment Class merupakan kelas yang berisi interaksi antara objek player dengan objek enemy dengan memanfaatkan kelas GamePlayer dan GameEnemy. Berikut adalah gambar class diagram dari GameEnvironment.java.

- Property width dan height merupakan dimensi panjang dan lebar GameEnvironment. Dalam hal ini kita dapat membayangkan bahwa GameEnvironment adalah sebuah scene game.
- Property arrPlayer dan arrEnemy merupakan list array dari Player dan Enemy.
- Method addPlayer untuk menambahkan elemen GamePlayer pada array list arrPlayer.
- Method removePlayer untuk mengurangi elemen GamePlayer pada array list arrPlayer.
- Method getAllPlayers untuk menampilkan elemen-elemen arraylist dari arrPlayer.
- Method addEnemy untuk menambahkan elemen GameEnemy pada array list arrEnemy.
- Method removeEnemy untuk mengurangi elemen GameEnemy pada array list arrEnemy.
- Method getAllEnemies untuk menampilkan elemen-elemen arraylist dari arrEnemy.
- Method Interaction berisi seperti pseudocode dibawah ini:

```
If arrPlayer == null or arrEnemy == null then
  Output("Player or enemy not sets");

For i=0 to arrPlayer.size():
  For j=0 to arrEnemy.size():
   If arrPlayer.get(i).getY() != arrEnemy.get(j).getY() then
      Output("Player: " + arrPlayer.get(i) + " and Enemy: " + arrEnemy.get(j) +
      " not in the same Y position" )
   If EuclideanDistance(arrPlayer.get(i).getX(),arrPlayer.get(i).getY(),
      arrEnemy.get(j).getX(),arrEnemy.get(j).getY()) < 2 then
      Output("Player: " + arrPlayer.get(i) + " Attacked")
      Output("Enemy: " + arrEnemy.get(j) + " loses")
      removeEnemy(arrEnemy.get(j))
   else
      Output("==> Player "+arrPlayer.get(i));
      arrPlayer.get(i).Run((int)Math.ceil(Math.random()*10));
      Output("current x position =  " + arrPlayer.get(i).getX() +" <==")
```

- Static Method EuclideanDistance berisi rumus jarak Euclidean vektor 2D antara (x1,y1) dan (x2,y2). Method ini mengembalikan nilai integer. Berikut adalah rumus dari Euclidean distance 2D dengan titik (p1,p2) dan (q1,q2).

$$\mathrm{d}(\mathbf{p}, \mathbf{q}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2}.$$

Sumber: https://en.wikipedia.org/wiki/Euclidean_distance

Testing Class

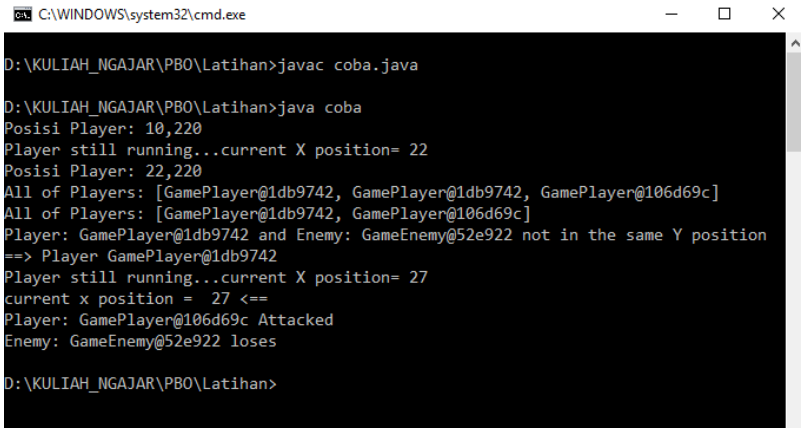
Buat satu class diberi nama TestGame.java berisi method main sebagai entry point. Untuk melakukan testing pada class-class yang sudah dibuat sebelumnya, dapat tambahkan kode berikut pada main.

```
GamePlayer hero = new GamePlayer();
hero.setDimension(12,12);
hero.setPosition(10,220);
System.out.println("Posisi Player: "+hero.getX()+","+hero.getY());
hero.Run(12);
System.out.println("Posisi Player: "+hero.getX()+","+hero.getY());
GamePlayer hero2 = new GamePlayer();
hero2.setDimension(12,32);
hero2.setPosition(10,10);

GameEnemy monster = new GameEnemy();
monster.setDimension(12,32);
monster.setPosition(10,10);

GameEnvironment scene = new GameEnvironment();
scene.addPlayer(hero);
scene.addPlayer(hero);
scene.addPlayer(hero2);
scene.getAllPlayers();
scene.removePlayer(hero);
scene.getAllPlayers();
scene.addEnemy(monster);
scene.Interaction();
```

Output dari kode diatas ini seharusnya sebagai berikut:



REFERENSI

Introduction to Java Programming, Comprehensive Version (10th Edition) by Y. Daniel Liang, 2015, Pearson Education, Inc.

CHECKLIST PERTEMUAN 1

KETERANGAN	NILAI
Kehadiran	**10**
GamePlayer (FULL)	**60**
GameEnemy (Partial)	**30**
TOTAL NILAI	**100**

CHECKLIST PERTEMUAN 2

KETERANGAN	NILAI
Kehadiran	**10**
GameEnemy (Full)	**30**
GameEnvironment (partial)	**60**
TOTAL NILAI	**100**

CHECKLIST PERTEMUAN 3

KETERANGAN	NILAI
Kehadiran	**10**
GameEnvironment	**40**
Testing	**40**
TOTAL NILAI	**100**

REKAPITULASI TUGAS 1

TUGAS 2 = (HASIL CHECKLIST 1 + HASIL CHECKLIST 2 + HASIL CHECKLIST 3)/3