

**LAPORAN TUGAS BESAR  
PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK  
SEMESTER GANJIL 2019-2020**

---

---

**MARBLE GUN**

**Disusun oleh:**

Defrialdy Zuhair Hanif	(F1D017014)
I Ketut Nadi Anggara Permadi	(F1D017028)
Melisa Novita Sari	(F1D017048)
Muhammad Ilhan Jahiri	(F1D017060)
Muhammad Naufal Rizqullah	(F1D017062)

**Asisten pembimbing:**

Farhan Yakub Bawazir



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS MATARAM  
2019**

**LEMBAR PENGESAHAN  
LAPORAN TUGAS BESAR  
PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK**

**Dikerjakan oleh:**

<b>Defrialdy Zuhair Hanif</b>	<b>(F1D017014)</b>
<b>I Ketut Nadi Anggara Permadi</b>	<b>(F1D017028)</b>
<b>Melisa Novita Sari</b>	<b>(F1D017048)</b>
<b>Muhammad Ilhan Jahiri</b>	<b>(F1D017060)</b>
<b>Muhammad Naufal Rizqullah</b>	<b>(F1D017062)</b>

**Mengetahui  
Dosen Pengampu,**

**Koordinator Asisten,**

**(Royana Afwani, S.T., M.T.)**  
**NIP: 19850707 201404 2 001**

**(Muhammad Naufal)**  
**NIM: F1D016060**

**Kepala Laboratorium Praktikum,**

**(Ir. Sri Endang Anjarwani M.Kom)**  
**NIP: 19660403 200604 2 001**

## Daftar Isi

LEMBAR PENGESAHAN.....	ii
Daftar Isi.....	iii
BAB 1 PENDAHULUAN.....	1
1.1    Latar Belakang.....	1
1.2    Deskripsi Aplikasi .....	2
1.3    Batasan Masalah.....	2
1.4    Tujuan.....	2
BAB 2 ANALISA DAN DESAIN .....	3
2.1    Use Case Diagram .....	3
2.2    Class Diagram .....	4
2.3    Deskripsi Kelas.....	4
BAB 3 IMPLEMENTASI.....	13
3.1    Implementation Class Diagram .....	13
3.2    Implementasi Konsep Pemrograman Berorientasi Objek.....	13
BAB 4 PENUTUP.....	16
4.1    Kesimpulan.....	16
4.2    Saran .....	16
Daftar Pustaka .....	17
LAMPIRAN .....	18

# BAB 1 PENDAHULUAN

## 1.1 Latar Belakang

Pada zaman sekarang ini, terdapat banyak sekali jenis-jenis *game*. Mulai dari *game* berbasis *desktop* dan *game* berbasis *mobile*. *Game* yang terkenal saat ini yaitu *game* berbasis *Mobile* atau *Android*, dimana *game* – *game* tersebut ada yang *online* maupun *offline*. *Game* sendiri banyak yang diminati oleh berbagai kalangan mulai dari anak-anak hingga orang dewasa. Salah satu *game* contohnya adalah sebuah *game* sederhana yang disebut *Marble Gun*.

*Marble Gun* merupakan salah satu *game* tembak gelembung hingga hancur atau menghilang, yang mana dalam *game* ini pemain diharuskan untuk mengalahkan gelembung-gelembung tersebut untuk bisa melanjutkan *game* ke *wave* berikutnya. *Game Marble Gun* memiliki alur yang runtut dan sistematis sehingga permainan *Marble Gun* menjadi teratur dan menarik. *Game Marble Gun* memiliki cara kerja yang sederhana. Pertama pemain harus menekan tombol *new game* terlebih dahulu baru mengisi username, kemudian menekan tombol *play* untuk memulai permainan, setelah itu akan ditampilkan sebuah arena yang terdiri dari bola gelembung dan bola penghancur, dimana bola penghancur akan mengeluarkan pelurunya untuk menghabisi gelembung yang ada pada arena sampai habis dan *game* akan lanjut ke arena berikutnya dengan tingkat kesulitan yang berbeda-beda, karena pada *wave-wave* berikutnya bola gelembung yang akan dihancurkan semakin banyak.

PBO (Pemrograman Berorientasi Objek) adalah suatu konsep pemrograman yang memanfaatkan kelas dan objek untuk keperluan penulisan program. Salah satu contoh bahasa pemrograman dengan orientasi objek adalah *Java*. Khususnya, ada aplikasi bernama *Netbeans* yang dapat digunakan untuk membuat suatu program *Java* dengan basis GUI (*Graphical User Interface*). Aplikasi ini dapat mempermudah *developer* untuk mengembangkan berbagai jenis program *Java* dengan basis GUI.

Karena itulah, di sini akan dibuat suatu program *Java* bernama *Marble Gun*, dengan memanfaatkan segala fitur yang ada dalam aplikasi *Netbeans*.

## 1.2 Deskripsi Aplikasi

*Marble gun* adalah sebuah *game* menembak peluru agar gelembung-gelembung yang ada pada arena bisa hancur. *Game* ini dimainkan dengan cara menembakkan peluru pada gelembung agar gelembung-gelembung yang ada hancur dan pemain bisa lanjut ke *wave* atau arena berikutnya. *Game* ini juga memiliki tiga nyawa atau tiga kesempatan untuk pemain, dimana nyawa pemain akan berkurang jika menyentuh musuh atau gelembung yang ada. *Game* ini akan selesai jika sudah mencapai *wave* terakhir, *game* akan menampilkan nama pemain dan jumlah *score* yang diperolehnya.

Prosedur dari program yang dibuat adalah sebagai berikut:

1. Program akan menampilkan tampilan menu dimana pada tampilan ini terdapat beberapa menu seperti *new game*, *higscore*, *help*, *info* dan *exit*. Pada saat pemain menekan tombol *new game* akan diarahkan kehalaman berikutnya yaitu halaman untuk mengisi nama pemain.
2. Pada halaman kedua, pemain diperintahkan untuk mengisi nama agar bisa bermain *game Marble gun*. Selanjutnya meng-klik tombol 'PLAY' dan akan dialihkan ke halaman ketiga.
3. Pada halaman ketiga, pemain akan mulai permainan dari *game Marble gun*.

## 1.3 Batasan Masalah

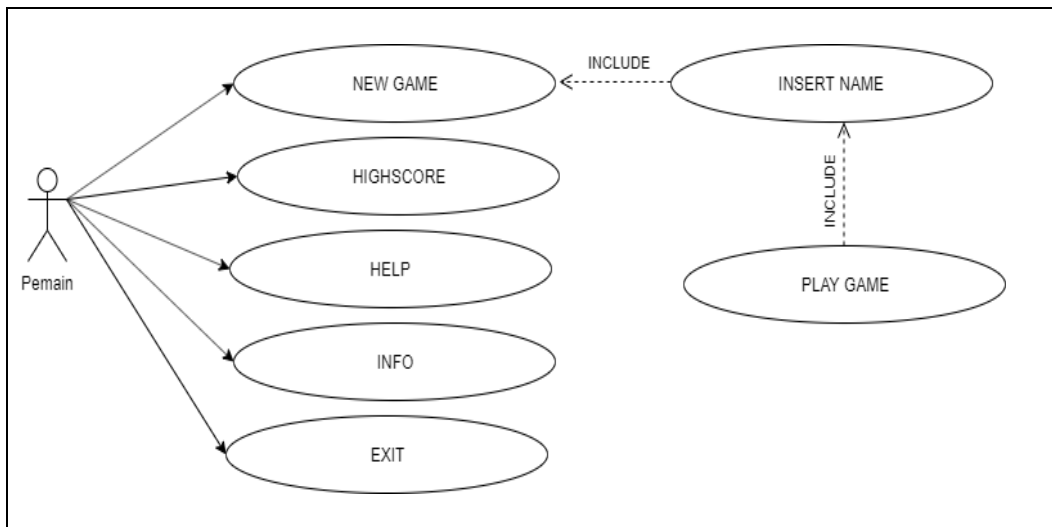
Batasan masalah dari aplikasi ini adalah hanya menggunakan satu pemain dan tiga nyawa, yang mana pemain hanya menembakkan peluru kepada gelembung-gelembung hingga meledak untuk bisa menuju ke *wave* atau arena berikutnya dan pemain tidak boleh menyentuh musuhnya agar nyawannya tidak habis.

## 1.4 Tujuan

Tujuan dibuatnya *game "Marble Gun"* ini adalah untuk mengimplemetasikan Pemrograman Berorientasi Objek agar bisa disebarluaskan kepada masyarakat luas dan dijadikan hiburan.

## BAB 2 ANALISA DAN DESAIN

### 2.1 Use Case Diagram



**Gambar 2.1** Use Case Diagram Marble Gun

Pada **Gambar 1** merupakan *Use Case* dari permainan *Marble Gun* dari sisi pemain. Yang mana, *Use Case* ini berisi hal-hal yang dapat dilakukan oleh pemain. Hal-hal yang dapat dilakukan oleh *user* antara lain sebagai berikut:

1. *New Game*
2. *Insert Name*
3. *Play Game*
4. *View high score*
5. *Help*
6. *Info*
7. *Exit*



*Method* yang berfungsi menampilkan *frame Help*

f. `main(String args)`

mengeksekusi program menu untuk dijalankan yang didalam nya terdapat *method* “`run`” untuk menampilkan *frame* menu ketika program dieksekusi

## 2. **Kelas *Game Panel***

Kelas ini merupakan kelas yang terdapat di *package Marble Gun* yang mengextends “`JPanel`” dan mengimplementasi “`Runnable`” dan “`KeyListener`”, berfungsi untuk Pada kelas ini berisikan *method-method* sebagai berikut :

a. `GamePanel()`

*Method* konstruktor dieksekusi pada saat pembuatan objek pada kelas “`GamePanel`”, yang didalamnya memanggil *method* “`super()`” untuk mempresentasikan kelas induk.

b. `String getNama()`

*Method* yang berfungsi untuk mengambil nilai variabel “`nama`”

c. `setNama(String nama)`

*Method* yang berfungsi untuk memberi nilai pada variabel “`nama`”

d. `addNotify()`

*Method* untuk membangunkan fungsi *thread* dan memanggil *method* “`start()`” dan memanggil fungsi *event* “`KeyListener`” yang digunakan untuk *event-event* yang berhubungan dengan tombol keyboard.

e. `run()`

*Method* yang akan menjalankan *loop game* dan juga membuat objek “`players`”, “`bullets`”, “`enemies`”, “`texts`”.

f. `gameUpdate()`

*Method* untuk membuat fungsi *wavenumber (level)* pada *game*, memunculkan *players*, *enemies*, *peluru* mengecek tabrakan antara *peluru* dengan musuh (*bullet-enemy collision*), mengecek tabrakan antara *player* dengan musuh (*player-enemy collision*), mengecek jika *player* mati (*check dead player*) dan mengecek jika musuh mati (*check dead enemies*).

g. `gameRender()`

*Method* untuk member gambar *background*, menggambar *player*, menggambar *peluru*, menggambar musuh, menggambar teks, menggambar



*wavenuumber*, menggambar nyawa *player*, menggambar *score player* dengan memanggil kelas dari masing masing fungsinya.

h. `gameDraw()`

*Method* untuk menggambar pada *game*.

i. `createNewEnemies()`

*Method* untuk membuat *enemy* baru saat selesai menyelesaikan *game* pada *wavenuumber* pertama.

j. `keyTyped(KeyEvent key)`

*Method* abstraks dari event *KeyListener* yang berfungsi saat *keyboard* ditekan.

k. `keyPressed(KeyEvent key)`

*Method* abstraks dari event *KeyListener* yang berfungsi saat *keyboard* ditahan.

l. `keyReleased(KeyEvent key)`

*Method* abstraks dari event *KeyListener* yang berfungsi saat *keyboard* dilepas.

### 3. **Kelas *Game***

Kelas ini merupakan kelas main yang terdapat di *package MarbleGun*, berfungsi untuk mengeksekusi program. Pada kelas ini berisikan *method-method* sebagai berikut :

a. `void run()`

*Method* yang berfungsi untuk membuat *jframe* utama *marble gun* dan berisi nilai untuk mengatur posisi *jframe*.

### 4. **Kelas *Enemy***

Kelas ini mengextends kelas *sub* dan mengimplementasi kelas “*Aktor*”. merupakan kelas yang terdapat di *package MarbleGun*, berfungsi untuk membuat musuh pada *game Marble Gun*. Pada kelas ini berisikan *method-method* sebagai berikut :

a. `Enemy (int type , int rank)`

*Method* konstruktor yang berfungsi dalam pembuatan objek dengan parameter `int type` dan `int rank`.

b. `getX()`

*Method* yang berfungsi mengambil nilai “*x*”.

c. `getY()`

- Method* yang berfungsi mengambil nilai “y”.
- d. `getr()`  
*Method* yang berfungsi mengambil nilai “r”.
- e. `getType()`  
*Method* yang berfungsi mengambil nilai “type”.
- f. `getRank()`  
*Method* yang berfungsi mengambil nilai “rank”.
- g. `isDead()`  
*Method* yang berfungsi memberi nilai “dead”.
- h. `hit()`  
*Method* yang berfungsi untuk menghitung nilai “health” *enemy*.
- i. `update ()`  
*Method* untuk memunculkan musuh.
- j. `draw (Graphics2D g)`  
*Method* untuk mengambar objek *enemy*.

## 5. **Kelas *Bullet***

Kelas ini meng*extends* kelas sub. Merupakan kelas yang terdapat di *package MarbleGun*, berfungsi untuk membuat peluru. Pada kelas ini berisikan *method-method* sebagai berikut :

- a. `Bullet (double angle, double x, double y)`  
*Method* konstruktor yang berfungsi dalam pembuatan objek dengan parameter “type” dan “rank”.
- b. `getx()`  
*Method* yang berfungsi mengambil nilai “x”.
- c. `gety()`  
*Method* yang berfungsi mengambil nilai “y”.
- d. `getr()`  
*Method* yang berfungsi mengambil nilai “r”.
- e. `update ()`  
*Method* untuk memunculkan peluru.
- f. `draw (Graphics2D g)`  
*Method* untuk mengambar objek peluru.

## 6. Kelas *High Score*

Kelas ini merupakan kelas yang terdapat di *package MarbleGun*, berfungsi untuk membuat *frame Highscore* dengan menampilkan nama dan score. Pada kelas ini berisikan *method-method* sebagai berikut :

### a. HighScore()

*Method* konstruktor yang berfungsi dalam pembuatan objek didalam kelas “Highscore” dan berisi nilai untuk mengatur posisi *frame* serta memanggil kelas “koneksi” untuk mengambil nilai dari *database*.

### b. tampil()

*Method* untuk memanggil “nama” dan “score” dari *database* untuk ditampilkan.

### c. anginActionPerformed(java.awt.event.ActionEvent evt)

*Method* berfungsi untuk

### d. appaActionPerformed(java.awt.event.ActionEvent evt)

*Method* nyang berfungsi untuk

### e. gas()

*Method* untuk menampilkan *frame Highscore*.

## 7. Kelas *Enter Name*

Kelas ini merupakan kelas yang terdapat di *package MarbleGun*, berfungsi untuk menampilkan *frame enter name* yang berisi masukkan nama pada *game*. Pada kelas ini berisikan *method-method* sebagai berikut :

### a. EnterName()

*Method* konstruktor yang berfungsi dalam pembuatan objek didalam kelas “Highscore” dan berisi nilai untuk mengatur posisi *frame* serta memanggil kelas “koneksi” untuk mengambil nilai dari *database*.

### b. PLAYActionPerformed(java.awt.event.ActionEvent evt)

*Method* untuk menutup *frame enter name* dan menuju *frame menu*.

### c. BACKActionPerformed(java.awt.event.ActionEvent evt)

*Method* untuk menutup *frame enter name* dan menuju *frame menu*.

### d. run ()

*Method* untuk menampilkan *frame Enter name*.

## 8. Kelas *Player*

Kelas ini mengimplementasi kelas *Aktor* merupakan kelas yang terdapat di *package MarbleGun*, berfungsi untuk membuat *player* pada *game*. Pada kelas ini berisikan *method-method* sebagai berikut :

a. *Player()*

*Method* konstruktor yang berfungsi dalam pembuatan objek.

b. *getX()*

*Method* yang berfungsi mengambil nilai “x”.

c. *getY()*

*Method* yang berfungsi mengambil nilai “y”.

d. *getr()*

*Method* yang berfungsi mengambil nilai “r”.

e. *getScore()*

*Method* yang berfungsi mengambil nilai “score”.

f. *getLives()*

*Method* yang berfungsi mengambil nilai “live”.

g. *isDead()*

*Method* yang berfungsi memberi nilai “dead”.

h. *isRecovering()*

*Method* yang berfungsi memberi nilai “recovering”.

i. *setLeft(boolean b)*

*Method* dengan parameter boolean b yang berfungsi memberi nilai “left”.

j. *setRigh(boolean b)*

*Method* dengan parameter boolean b yang berfungsi memberi nilai “right”.

k. *setUp(boolean b)*

*Method* dengan parameter boolean b yang berfungsi memberi nilai “up”.

l. *setDown(boolean b)*

*Method* dengan parameter boolean b yang berfungsi memberi nilai “down”.

m. *setFiring(boolean b)*

*Method* dengan parameter boolean b yang berfungsi memberi nilai “firing”.

n. *addScore(int i)*

*Method* dengan parameter i yang berfungsi memberi nilai pada “score”.

o. loseLife()

*Method* yang berfungsi mengecek saat player kehilangan nyawa maka “recovering” sama dengan “true”.

p. update()

*Method* untuk memunculkan *player*.

q. draw (Graphics2D g)

*Method* untuk menggambar objek *player*.

## 9. Kelas Teks

Kelas ini merupakan kelas yang terdapat di *package MarbleGun*, berfungsi untuk membuat teks *pada game*. Pada kelas ini berisikan *method-method* sebagai berikut :

a. Text(double x, double y, long time, String s)

*Method* konstruktor dengan parameter “double x”, “double y”, “long time”, dan “String s” berfungsi dalam pembuatan objek di dalam kelas “text”.

b. update()

*Method* untuk memunculkan objek *text*.

c. draw (Graphics2D g)

*Method* untuk menggambar objek *text*.

## 10. Kelas Info

Kelas ini merupakan kelas yang terdapat di *package MarbleGun*, berfungsi untuk menampilkan *frame info* pada *game*. Pada kelas ini berisikan *method-method* sebagai berikut :

a. Info()

*Method* konstruktor yang berfungsi dalam pembuatan objek di dalam kelas “Info” dan berisi nilai untuk mengatur posisi *frame*.

b. OKActionPerformed(java.awt.event.ActionEvent evt)

*Method* yang berfungsi untuk menutup *frame info* dengan menampilkan kembali *frame* menu.

c. gas()

*Method* untuk menampilkan *frame Info*.

## 11. Kelas *Help*

Kelas ini merupakan kelas yang terdapat di *package MarbleGun*, berfungsi untuk menampilkan *frame Help* pada *game*. Pada kelas ini berisikan *method-method* sebagai berikut :

a. *Help()*

*Method* konstruktor yang berfungsi dalam pembuatan objek di dalam kelas “*Help*” dan berisi nilai untuk mengatur posisi *frame*.

b. *OKActionPerformed(java.awt.event.ActionEvent evt)*

*Method* yang berfungsi untuk menutup *frame Help* dengan menampilkan kembali *frame* menu.

c. *Gas()*

*Method* untuk menampilkan *frame Info*.

## 12. Kelas *Sub*

Kelas ini yang terdapat di *package MarbleGun*, merupakan kelas abstrak dari *game MarbleGun*. Pada kelas ini berisikan *method-method* sebagai berikut :

a. *gety()*

*Method* abstraks yang berfungsi mengambil nilai “*y*”.

b. *getx()*

*Method* abstraks yang berfungsi mengambil nilai “*x*”.

c. *getr()*

*Method* abstraks yang berfungsi mengambil nilai “*r*”.

d. *draw(Graphics2D g)*

*Method* abstraks dengan parameter “*Graphics2D g*” yang berfungsi menggambar objek.

## 13. Kelas *Aktor*

Kelas ini merupakan kelas *interface* yang terdapat di *package MarbleGun*. Pada kelas ini berisikan *method-method* sebagai berikut :

a. *isDead()*

*Method* yang berfungsi memberi nilai “*dead*”.

b. *update ()*

*Method* untuk memunculkan objek.

c. *draw(Graphics2D g)*

*Method* abstract dengan parameter “Graphics2D g” yang berfungsi menggambar objek.

#### **14. Kelas *Play Musik***

Kelas ini merupakan yang terdapat di *package MarbleGun*, berfungsi untuk memutar musik pada game. Pada kelas ini berisikan *method-method* sebagai berikut :

- a. *playMusic(String filepath)*

*Method* untuk memutar musik pada game.

#### **15. Kelas *Image***

Kelas ini merupakan kelas yang terdapat di *package MarbleGun*, berfungsi untuk menampilkan gambar pada game. Pada kelas ini berisikan *method-method* sebagai berikut :

- a. *getResourceBufferedImage (String patch)*

*Method* untuk mengambil gambar untuk diletakkan pada game.

#### **16. Kelas *Gambar***

Kelas ini merupakan kelas yang terdapat di *package Marble Gun*, berfungsi untuk menampilkan gambar pada game. Pada kelas ini berisikan *method-method* sebagai berikut :

- a. *getResourceImage(String path)*

*Method* untuk mengambil gambar untuk diletakkan pada game.

#### **17. Kelas *Koneksi***

Kelas ini merupakan kelas yang terdapat di *package Marble Gun*, berfungsi untuk menghubungkan nilai pada program ke dalam *database*. Pada kelas ini berisikan *method-method* sebagai berikut :

- a. *push(String nama, int score)*

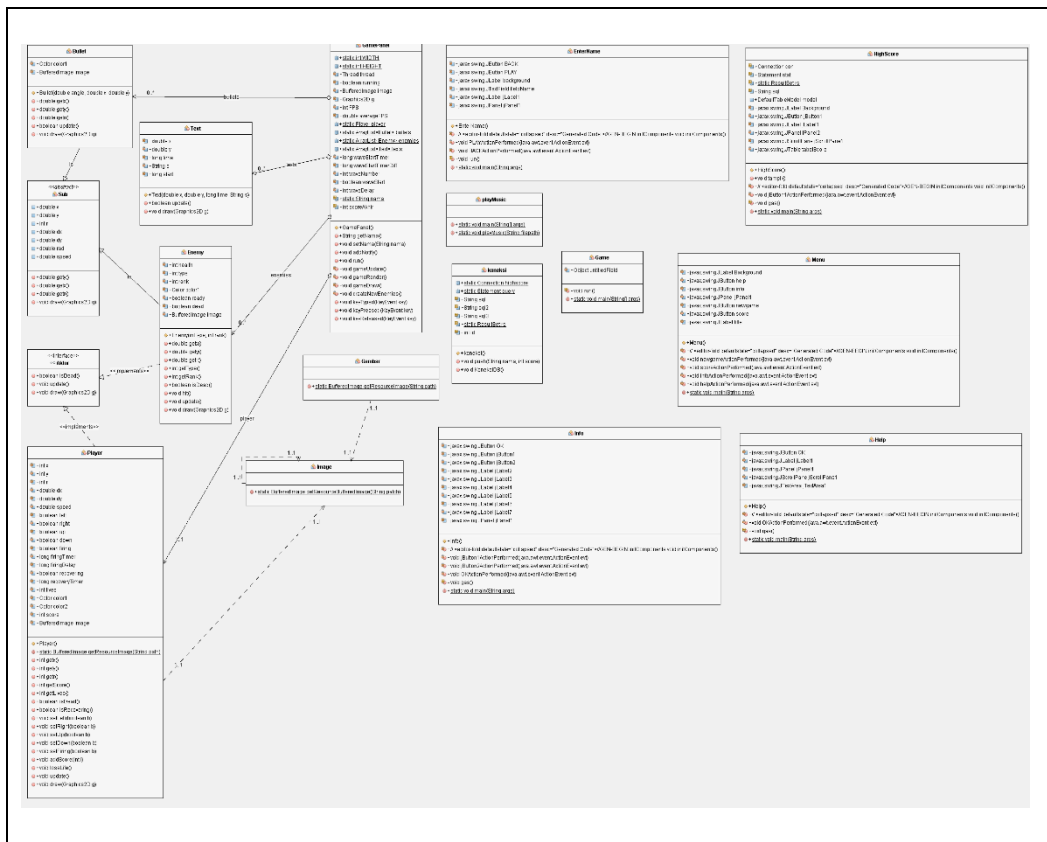
*Method* dengan parameter String nama dan int score untuk mengambil nilai nama dan score untuk dimasukkan ke dalam *database*.

- b. *KoneksiDB()*

*Method* untuk menghubungkan nilai ke dalam *database*.

## BAB 3 IMPLEMENTASI

## 1. Implementation Class Diagram



**Gambar 3.1** *Class Diagram Marble Gun*

## 2. Implementasi Konsep Pemrograman Berorientasi Objek

## 1. Abstraksi

Abstraksi adalah suatu cara melihat suatu objek dalam bentuk sederhana. Kelas merupakan bentuk abstrak dari suatu objek. Pada *project game* ini terdapat beberapa beberapa kelas yang akan dijadikan sebagai objek utama pada *game* ini yaitu Player, Bullet, Enemy, dan GamePanel.

a. Palyer

*Class* Player untuk membuat dan mengatur perilaku apa saja yang dapat dilakukan oleh objek yang akan dibuat, objek dari *class* ini dijadikan sebagai karakter yang dikendalikan oleh pemain.

b. Bullet

*Class* Bullet untuk membuat atau mengatur tempat muncul dan pergerakan dari objek *class* Bullet ini, objek dari *class* adalah sebagai peluru yang ditembakkan oleh pemain.



c. Enemy

*Class* Enemy untuk membuat dan mengatur perilaku dari objek *class* Enemy ini, objek dari *class* ini sebagai musuh yang menjadi target untuk ditembak dan dihindari oleh pemain. Objek dari *class* ini juga ada beberapa tipe untuk membuat *game* lebih seru.

d. GamePanel

*Class* GamePanel ini digunakan untuk membuat objek *field* dan rangkaian *game* serta semua proses yang ada di *game* ini.

## 2. Enkapsulasi

Enkapsulasi adalah mekanisme menyembunyikan informasi yang menjadikan data suatu kelas tidak bisa diakses secara sembarangan[2]. Pembungkusan variabel dan *method* dalam sebuah objek dalam sebuah objek dalam bagian yang terlindungi disebut sebagai enkapsulas. Penyembunyian informasi dilakukan dengan menggunakan *keywoard private* dan *protected* pada elemen data.

Enkapsulasi pada *project game* tugas besar ini terletak pada pada *class* Player, Enemy, Bullet, dan beberapa *class* lain yang menggunakan variabel *private*. *Class* yang paling banyak menggunakan *private* adalah *class* Player, Bullet, Enemy, dan GamePanel karena terdapat banyak variabel yang hanya boleh diakses oleh *class* tersebut, sedangkan untuk *method* tidak ada yang *private* karena semua *method* dari suatu *class* dibutuhkan pada *class* yang lain untuk membuat *game* bisa berjalan dengan baik.

## 3. Pewarisan

Pewarisan adalah proses penciptaan kelas baru dengan mewarisi karakteristik kelas yang ada, ditambah dengan karakteristik unik dari kelas baru tersebut. Pada *project* tugas besar ini hanya terdapat pewarisan tunggal yaitu antara lain:

- a. Pewarisan dari *abstract class* Sub ke *class* Bullet.
- b. Pewarisan dari *abstract class* Sub ke *class* Enemy.

## 4. Polymorphism

*Polymorphism* adalah kemampuan suatu *method* object yang memiliki banyak bentuk [1]. Pada *project game* ini terdapat beberapa *polymorphism* berupa *override method*, sedangkan tidak terdapat *overload*. *Overirde* yang digunakan antara lain:

- a. *Class* Bullet meng-*override* semua *abstract method* dari *abstract class* Sub.
- b. *Class* Enemy meng-*override* semua *abstract method* dari *abstract class* Sub.
- c. *Class* Palyer meng-*override* semua *abstract method* dari *interface* Aktor.
- d. *Class* Enemy meng-*override* semua *abstract method* dari *interface* Aktor.

## **5. Interface**

Pada *project game* ini terdapat *interface* yang digunakan yaitu *interface* Aktor yang diimplementasikan pada *class* Player dan Enemy karena objek dari kedua *class* tersebut merupakan aktor utama dari game ini. Pada *class* GamePanel juga mengimplementasikan *interface* Runnable dan KeyListener yaitu *interface* bawaan dari *library swing* pada Java.

# **BAB 4 PENUTUP**

## **4.1 Kesimpulan**

Kesimpulan dari tugas akhir Praktikum Pemrograman Berorientasi Objek adalah sebagai berikut:

1. Mata kuliah Pemrograman Berorientasi Objek menuntut mahasiswa untuk dapat memahami konsep pembuatan program Java serta dapat mengimplementasikannya ke dalam Pemrograman Berorientasi Objek, seperti pembuatan *game* atau aplikasi.
2. Dalam pembuatan *project* akhir kuliah Pemrograman Berorientasi Objek, menggunakan pengimplementasian jurnal dari masing-masing modul pada saat praktikum.
3. Pada tema *project* akhir kuliah Pemrograman Berorientasi Objek, tema yang digunakan adalah membuat *game* edukasi yaitu *game Marble Gun* yang merupakan *game* tembak atau menghancurkan gelembung.

#### **4.2 Saran**

Saran untuk projek memperbanyak *refrensi* untuk mengetahui fungsi-fungsi pemrograman *game* java, dan tiap-tiap anggota wajib memahami secara garis besar tentang judul projek mereka atau memahami secara mendetail.

## DAFTAR PUSTAKA

- [1] Indrayani. 2014. *Database System Case Study All in One*. Jakarta: PT Gramedia.
- [2] Ridho Barakbah, Ali. 2006. *Pemrograman Berbasis Obyek*. Institut Teknologi 10 November.