BAB III
ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan - perbaikannya.

3.1.1 Analisis Game Sejenis yang Sudah Ada

Sebelum merancang game yang akan dibuat, terlebih dahulu dilakukan analisis pada game sejenis yang sudah ada. Ini bertujuan agar game yang sudah dibuat nanti memiliki solusi dari permasalahan game yang sudah ada.

Game yang memiliki tipe permainan yang hampir sama dengan Game 2D Gatotkaca Evolution sudah ada beberapa, diantaranya *Game Plane Fighter* Menggunakan Flash [16], *Fighter Plane Using Python* [17], *Game hybrid shooter side-scrolling destroyer* Garuda [18]. Setelah melakukan observasi terhadap tiga game tersebut, didapat kesimpulan seperti yang bisa dilihat pada tabel III.1

Tabel 3.1. Analisis Game yang Sudah Ada

Nama Game	Kelebihan	Kekurangan	Tampilan
Plane Fighter	- Terdapat menu pengaturan suara - Terdapat simpan High Score	- Efek animasi belum maksimal - Tidak terdapat AI pada NPC	

Gamma Bros	- Terdapat efek ledakan	- Tampilan game kurang menarik - Efek animasi masih kurang	
Cloud Soldier	- Terdapat simpan High Score	- Tidak terdapat AI pada NPC	

3.1.2 Analisis game yang akan dibangun

Game yang akan dibangun berupa game *Arcade* dengan menggunakan karakter utamanya yaitu tokoh perwayangan Gatotkaca. Game ini dibangun sebagai sarana untuk melestarikan tokoh Gatotkaca dan menciptakan karakter superhero asal Indonesia melalui aplikasi game. Dengan grafik tampilan yang cukup halus, efek-efek animasi yang digunakan dan setiap level yang berbeda baik tingkat kesulitannya maupun lokasinya, diharapkan dapat menjadi alternatif solusi dari permasalahan yang ada di game sejenis yang sudah ada.

Fitur-fitur yang akan dibuat pada game ini :

1. Game *Arcade*
2. Menggunakan Gatotkaca sebagai karakter utama
3. Efek animasi
4. Setiap level mengambil lokasi di kota-kota Indonesia

3.1.2.1 Storyline

Seorang Mahasiswa yang bernama Gatot melakukan pendakian ke Gunung Jamurdipa bersama kawan-kawannya. Namun dalam perjalanan, Gatot terpisah dari rombongannya. Hari sudah mulai gelap namun Gatot belum juga bertemu dengan kawan-kawannya. Tiba-tiba saja Gatot melihat

sebuah benda yang bersinar terang. Gatotpun mengambil benda tersebut yang ternyata adalah pusaka mastaba peninggalan Raden Gatotkaca. Kekuatan Gatotkaca yang terdapat dalam pustaka mastaba akhirnya masuk ke dalam tubuh Gatot yang membuat Gatot memiliki kekuatan super.

Di lain tempat, seorang ilmuwan yang memiliki ambisi menguasai Indonesia, berusaha menciptakan mahluk yang memiliki kekuatan super yang akan ia gunakan untuk menaklukan kota-kota besar di Indonesia. Gatot yang mendengar hal itu, berusaha untuk menghentikan niat ilmuwan tersebut. Namun, ilmuwan jahat sudah terlanjur melepaskan mahluk ciptaannya ke beberapa kota besar di Indonesia. Gatotpun harus bekerja keras untuk menghancurkan mahluk-mahluk ciptaan ilmuwan jahat itu.

1. Level 1

   Pemain akan berada di Kota Jakarta dengan misi menghancurkan musuh. Setelah berhasil mengalahkan musuh sesuai jumlah yang telah ditentukan maka pemain dapat melanjutkan ke level berikutnya.

2. Level 2

   Sama seperti level 1, dimana pemain harus menghancurkan musuh sesuai jumlah yang ditentukan untuk melanjutkan ke level 3, namun karakter NPC berbeda jenisnya. Tingkat kesulitannya pun berbeda dengan level 1. Selain itu, lokasi pertempuran di level 2 berada di Kota Yogyakarta.

3. Level 3

   Level ini merupakan level terakhir pada game 2D Gatotkaca Evolution. Misinya pun sama seperti level 1 dan level 2, pada level terakhir ini lokasi pertempuran berada di kota Bandung.

3.1.2.2 Pelevelan

Dalam game ini terdapat 3 level, yang pada setiap levelnya mempunyai *gameplay* yang sama namun memiliki tingkat kesulitan yang berbeda. Yang membedakan pada setiap levelnya yaitu :

1. Musuh

   Semakin tinggi level, maka jenis musuh berbeda dan akan bertambah kuat sehingga permainan menjadi lebih sulit. Jenis – jenis musuh yang terdapat dalam game ini adalah :

   a. RoboKaca
   b. RoboCakil
   c. Rahwana
   d. Raksasa
   e. ButoCakil
   f. Naga
   g. Kurawa
   h. Bos

2. Lokasi

   Setiap level memiliki lokasi yang berbeda-beda. Pada level 1 lokasi akan berada di Kota Jakarta dengan mengambil latar Pantai Ancol. Level 2 lokasi berpindah ke Kota Yogyakarta dengan mengambil latar Candi Prambanan dan tugu Yogyakarta. Sedangkan di level 3 yaitu level terakhir lokasi berada di Kota Bandung dengan mengambil latar Gedung Sate.

3.1.2.3 Gameplay

Gameplay permainan ini adalah menghancurkan mahluk ciptaan ilmuwan jahat yang berambisi menguasai Indonesia, dimana pemain dapat menembakan bola-bola api dari tokoh Gatot yang memiliki kekuatan super Gatotkaca. Disetiap level, memiliki jenis musuh yang berbeda jenisnya. Setiap musuh memiliki jenis tembakan dan kekuatan yang berbeda-beda. Di level 3 yaitu level terakhir akan muncul karakter musuh yang paling kuat yang harus dihancurkan untuk menyelesaikan permainan.

3.1.2.4 Scoring

Pada game ini, *score* akan bertambah setiap kali pemain berhasil menghancurkan musuh. Setiap musuh memiliki point yang berbeda-beda apabila hancur. *Point* terkecil adalah 5 sedangkan point terbesar adalah 20. Pemain dapat melanjutkan ke level berikutnya apabila target menghancurkan musuh terpenuhi.

3.1.3 Analisis Algoritma

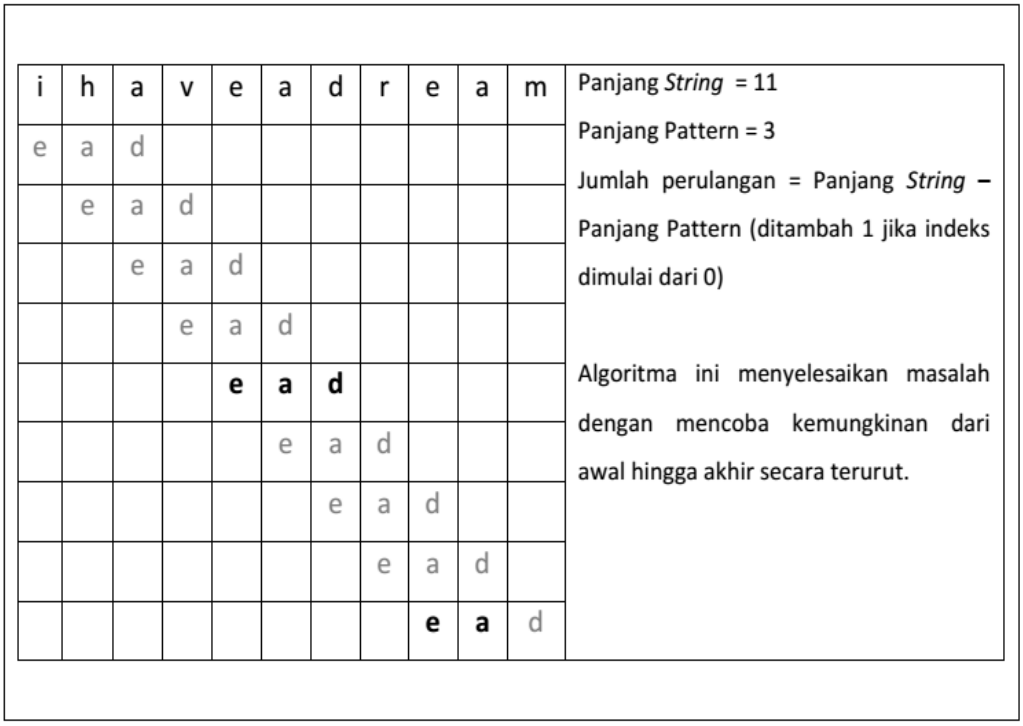
Algoritma yang dipakai pada game Gatotkaca Evolution ini adalah Algoritma Brute Force. Dimana Algoritma ini digunakan pada karakter NPC agar bergerak mengarah kepada karakter pemain. Untuk sampai pada posisi karakter pemain, maka NPC harus meng-copy posisi karakter pemain. Dalam hal ini, jarak antara karakter pemain dan NPC merupakan Array. Oleh karena itu, maka untuk menemukan posisi karakter pemain, posisi NPC harus dikurangi dengan setiap array.

Library yang digunakan adalah Slick2D, dimana di dalamnya terdapat class Vector2f. Dalam class Vector2f terdapat 2 sumbu yaitu x dan y. Nilai x dan y sesuai dengan inputan. Dalam class Vector2f terdapat method copy dan sub(subtract). Method copy digunakan suatu kelas untuk meng-copy vector yang dimiliki oleh kelas lainnya. Pada game ini method copy berfungsi untuk menentukan posisi karakter pemain yang akan dituju oleh karakter musuh. Method sub(subtract) atau pengurangan digunakan untuk fungsi pengurangan antara masing-masing sumbu x dan y yang ada pada 2 kelas. Penjelasannya pada tabel berikut :

Tabel 3.2. Penjelasan Algoritma Brute Force pada game

Pseudocode	penjelasan
`Iterator<Enemy> i = enemys.iterator();`	Penginisialisasian *array*
`While(i.hasNext())`	Cek apakah

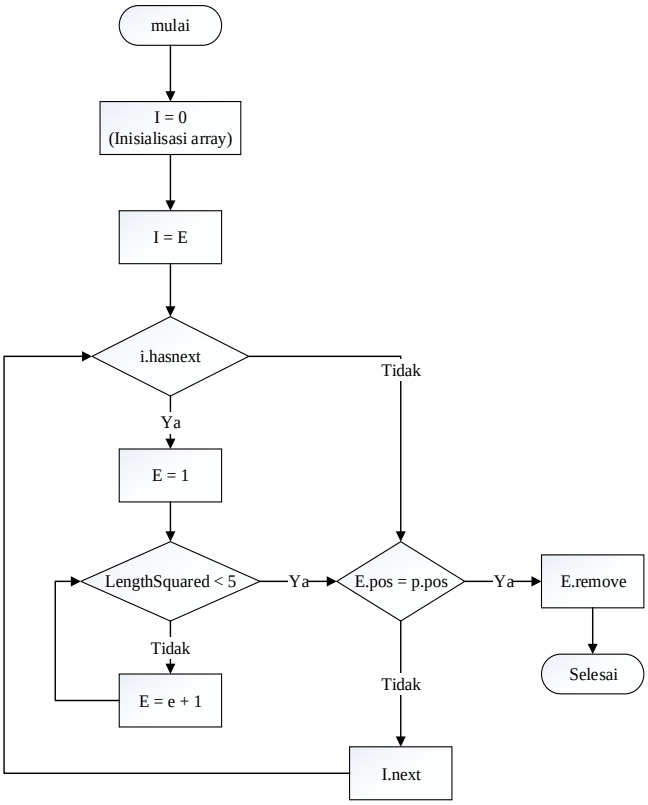
{
Enemy e = i.next();
 	
If(e.getPos().copy().sub(p.getPos()).lengthSquared()
< 5)
{
p.lives -= 2;
i.remove;
}
}
	terdapat elemen *array*, jika ada kurangi posisi *enemy* dengan posisi player. *Collision* akan terjadi jika nilai posisi *enemy* dan posisi *player* kurang dari 5, maka darah *player* dikurangi 2 dan *enemy* dihapus.



Gambar 3.1. Cara kerja Algoritma Brute Force pada pencocokan pola String

Implementasi Algoritma *Brute Force* dalam pencocokan pola *string* sederhana dengan memeriksa dan membandingkan *Pattern* yang dicari dengan *Pattern* yang ada di *String*. Bila hasil cocok, maka program melakukan keluaran

sesuai perintah, sedangkan bila hasil belum ditemukan maka program akan melakukan perulangan hingga hasil ditemukan.



Gambar 3.2. Flowchart Algoritma Brute Force pada game

Inti dari Algoritma Brute Force sendiri yaitu penyelesaian masalah dengan mencoba kemungkinan dari awal hingga akhir secara terurut. Perhitungannya dimulai dari sumbu x lalu dilanjutkan dengan sumbu y. urutannya dapat dilihat sebagai berikut :

1. Posisi awal musuh (8,5) Posisi awal pemain (1,1)
   X = 8 – 1 = 7 hasil (7,5)
2. Posisi musuh (7,5) Posisi pemain (1,1)
   Y = 5 – 1 = 4 hasil (7,4)
3. Posisi musuh (7,4) Posisi pemain (1,1)
   X = 7 – 1 = 6 hasil (6,4)
4. Posisi musuh (6,4) Posisi pemain (1,1)
   Y = 4 – 1 = 3 hasil (6,3)

5. Posisi musuh (6,3) Posisi pemain (1,1)
   X = 6 – 1 = 5 hasil (5,3)
6. Posisi musuh (5,3) Posisi pemain (1,1)
   Y = 3 – 1 = 2 hasil (5,2)
7. Posisi musuh (5,2) Posisi pemain (1,1)
   X = 5 – 1 = 4 hasil (4,2)
8. Posisi musuh (4,2) Posisi pemain (1,1)
   Y = 2 – 1 = 1 hasil (4,1) titik Y ditemukan
9. Posisi musuh (4,1) Posisi pemain (1,1)
   X = 4 – 1 = 3 hasil (3,1)
10. Posisi musuh (3,1) Posisi pemain (1,1)
    X = 3 – 1 = 2 hasil (2,1)
11. Posisi musuh (2,1) Posisi pemain (1,1)
    X = 2 – 1 = 1 hasil (1,1) posisi pemain ditemukan

	0	1	2	3	4	5	6	7	8
0									
1		pemain							
2									
3									
4									
5									musuh

Gambar 3.3. Cara kerja Algoritma Brute Force

3.1.4 Analisis Kebutuhan Non-Fungsional

Analisis non-fungsional merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen-komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukan yang diperlukan sistem keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk mengolah masukan sehingga menghasilkan suatu keluaran yang diinginkan.

Pada analisis kebutuhan sistem non-fungsional ini dijelaskan analisis kebutuhan perangkat lunak, analisis kebutuhan perangkat keras, dan analisis pengguna.

3.1.4.1 Analisis Kebutuhan Perangkat Lunak

Perangkat keras komputer tidak berarti tanpa perangkat lunak, begitu juga sebaliknya. Jadi perangkat lunak dan perangkat keras saling mendukung satu sama lain. Perangkat keras hanya berfungsi jika diberikan instruksi – intruksi kepadanya. Instruksi - instruksi inilah yang disebut dengan perangkat lunak.

1. Dari sisi pengembang

   Perangkat lunak yang digunakan untuk mengembangkan game ini adalah sebagai berikut:

   1. Sistem operasi Windows 7
   2. Eclipse sebagai *Integrated Development Environment* (IDE) nya
   3. JDK Versi 7
   4. Aplikasi pengolahan grafis Adobe Photoshop CS5

2. Dari sisi pemain

   Untuk menjalankan game Gatotkaca Evolution ini dibutuhkan perangkat lunak, antara lain:

   1. Sistem operasi Windows, Mac OS, atau Linux
   2. JDK Versi 6 ke atas
   3. Java Plugin

3.1.4.2 Analisis Kebutuhan Perangkat Keras

Agar aplikasi dapat berjalan dengan baik, maka dibutuhkan perangkat keras yang sesuai dengan kebutuhan aplikasi. Di bagian ini akan dijelaskan analisis kebutuhan perangkat keras dari sisi pengembang dan dari sisi pemain.

1. Dari sisi pengembang

   Perangkat keras yang digunakan untuk membangun game ini adalah sebagai berikut :

   1. Prosesor 2 GHz
   2. Memori 2 GB
   3. Space kosong di harddisk 5 GB
   4. VGA card 512 MB
   5. Monitor 1280 x 900
   6. Mouse dan keyboard
   7. Speaker

2. Dari sisi pemain

   Spesifikasi perangkat keras yang direkomendasikan agar dapat menjalankan game Gatotkaca Evolution ini adalah sebagai berikut :

   1. Prosesor dengan kecepatan 1.8 Ghz
   2. Memori 256 Mb
   3. Space kosong di harddisk 15 MB
   4. VGA card 256 MB
   5. Monitor
   6. Mouse dan keyboard

7. Speaker

3.1.4.3 Analisis Pengguna

Analisis pengguna merupakan analisis terhadap pengguna yang akan menggunakan aplikasi yang telah dibangun. Oleh karena itu, dibuatlah analisis pengguna dengan kriteria minimal yang harus dimiliki oleh pengguna antara lain sebagai berikut :

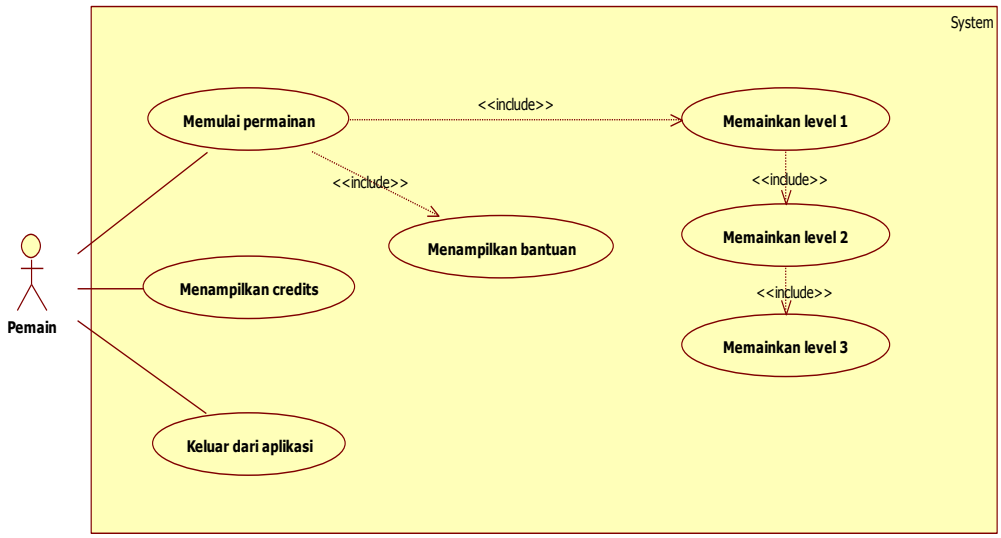
1. Bisa mengoperasikan komputer
2. Biasa menggunakan sistem operasi Windows
3. Pernah memainkan game yang sejenis dengan game Gatotkaca *Evolution*
4. Bisa membaca dengan lancar

3.1.5 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik sesuai kebutuhan. Pada bagian ini akan dibahas tentang *use case diagram*, *activity diagram*, *class diagram*, dan *sequence diagram*.

3.1.5.1 Use Case Diagram

Pada sub bab berikut ini dijelaskan bagaimana *Use Case* yang terjadi dalam pembangunan game ini yang dimodelkan dalam sekumpulan *Use Case* dan *Actor* dan bagaimana hubungan-hubungannya yang akan dipetakan dalam diagram *Use Case*, dilengkapi dengan skenario untuk menjelaskan dari gambaran *Use Case* yang ada dengan menggambarkan skenario per *Use Case*. *Use Case Diagram* dapat dilihat pada



Gambar 3.4. Use Case Diagram

1. Actor Definiton

Actor Definiton berfungsi untuk menjelaskan *actor* yang terdapat pada *Use Case Diagram* dan menggambarkan pihak-pihak yang berperan dalam sistem. Pemain berperan sebagai *actor* dengan aktivitas memainkan *game*.

2. Use Case Definition

Use Case definition berfungsi untuk menjelaskan fungsi *Use Case* yang terdapat pada *Use Case Diagram* dan menjelaskan aktivitas yang disiapkan oleh sistem. *Use Case Definition* diterangkan pada tabel

Tabel 3.3. Use Case Definition

No	Use Case	Deskripsi
1	Memulai permainan	Proses untuk memainkan *game*
2	Menampilkan bantuan	Proses untuk menampilkan menu bantuan
3	Menampilkan *credits*	Proses untuk menampilkan menu *credits*
4	Keluar dari aplikasi	Proses untuk keluar dari aplikasi
5	Memainkan level 1	Proses untuk menampilkan dan memainkan level 1

6	Memainkan level 2	Proses untuk menampilkan dan memainkan level 2
7	Memainkan level 3	Proses untuk menampilkan dan memainkan level 3

3. Use Case Scenario

Use Case Scenario mendeskripsikan urutan langkah-langkah dalam proses bisnis baik yang dilakukan aktor terhadap sistem maupun yang dilakukan oleh sistem terhadap aktor. Berdasarkan *use case diagram* pada gambar di atas maka *use case scenario* untuk aplikasi yang dibangun akan dijelaskan sebagai berikut.

a. Use Case Scenario Memulai permainan

Use Case Scenario dari *Use Case* Memulai permainan dijelaskan pada tabel di bawah ini.

Tabel 3.4. Use Case Scenario Memulai permainan

Identifikasi	
Nomor	1
Nama	Memulai permainan
Tujuan	Proses yang dilakukan pemain untuk mulai memainkan permainan
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pemain berada di menu utama
Kondisi Aktor	**Reaksi Sistem**
1. Memilih tombol *play menu*	
	2. Sistem menampilkan menu *play menu*
Kondisi Akhir	Menampilkan halaman *play menu*

b. Use Case Scenario menampilkan bantuan

Use Case Scenario untuk *Use Case* menampilkan bantuan dijelaskan pada tabel di bawah ini.

Tabel 3.5. Use Case Scenario menampilkan bantuan

Identifikasi	
Nomor	3
Nama	Menampilkan menu bantuan
Tujuan	Proses yang dilakukan pemain untuk menampilkan menu bantuan
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pemain berada di menu utama
Kondisi Aktor	**Reaksi Sistem**
1. Memilih tombol bantuan	
	2. Sistem menampilkan menu bantuan
3. Memilih tombol kembali	
	4. Sistem menampilkan menu utama
Kondisi Akhir	Menampilkan halaman menu bantuan dan apabila pemain memilih tombol kembali maka sistem akan menampilkan menu utama

c. Use Case Scenario menampilkan credits

Use Case Scenario untuk *Use Case* menampilkan *credits* dijelaskan pada tabel di bawah ini.

Tabel 3.6. Use Case Scenario menampilkan credits

Identifikasi	
Nomor	3
Nama	Menampilkan menu *credits*
Tujuan	Proses yang dilakukan pemain untuk menampilkan menu *credits*
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pemain berada di menu utama
Kondisi Aktor	**Reaksi Sistem**
1. Memilih tombol *credits*	
	2. Sistem menampilkan menu *credits*
3. Memilih tombol kembali	
	4. Sistem menampilkan menu utama
Kondisi Akhir	Menampilkan halaman menu *credits* dan apabila pemain memilih tombol kembali maka sistem akan menampilkan menu utama

d. Use Case Scenario keluar dari aplikasi

Use Case Scenario untuk *Use Case* keluar dari aplikasi dijelaskan pada tabel di bawah ini.

Tabel 3.7. Use Case Scenario keluar dari aplikasi

Identifikasi	
Nomor	4
Nama	Keluar dari aplikasi
Tujuan	Proses yang dilakukan pemain untuk keluar dari aplikasi
Aktor	Pemain

Skenario Utama	
Kondisi Awal	Pemain berada di menu utama
Kondisi Aktor	**Reaksi Sistem**
1. Memilih tombol keluar	
	2. Sistem menutup aplikasi
Kondisi Akhir	Sistem menutup aplikasi

e. Use Case Scenario menampilkan level 1

Use Case Scenario untuk *Use Case* menampilkan level 1 dijelaskan pada tabel di bawah ini.

Tabel 3.8. Use Case Scenario menampilkan level 1

Identifikasi	
Nomor	4
Nama	Memainkan level 1
Tujuan	Proses yang dilakukan pemain untuk menampilkan level 1
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pemain berada di halaman menu level
Kondisi Aktor	**Reaksi Sistem**
1. Memilih tombol main	
	2. Sistem menampilkan *scene* 1
	3. Sistem menampilkan tombol main dan lanjut
4. Memilih tombol lanjut	
	5. Sistem menampilkan *scene* 2
	6. Sistem menampilkan tombol main dan lanjut
7. Memilih tombol lanjut	
	8. Sistem menampilkan *scene* 3

	9. Sistem menampilkan tombol main dan lanjut
10. Memilih tombol lanjut	
	11. Sistem menampilkan *scene* 4
	12. Sistem menampilkan tombol main dan lanjut
13. Memilih tombol lanjut	
	14. Sistem menampilkan *scene* 5
	15. Sistem menampilkan tombol main
16. Memilih tombol main	
	17. Sistem menampilkan level 1
18. Mengendalikan karakter	
	19. Karakter bergerak
	20. Cek *power* dan darah
	21. Jika peluru musuh mengenai pemain maka darah pemain berkurang
	22. Jika musuh menabrak pemain maka darah pemain berkurang
	23. Jika darah pemain habis maka sistem menampilkan menu *game over* dimana pemain bisa memilih ulangi
	24. Jika pemain mendapatkan objek berbentuk kilat maka *power* bertambah
	25. Jika *power* sudah penuh maka akan muncul bantuan
26. Menyerang musuh	
	27. Perhitungan *score* saat berhasil menghancurkan musuh
	28. Target menghancurkan musuh akan bertambah saat pemain dapat menghancurkan musuh

29. Menyelesaikan level permainan	
	30. Jika target menghancurkan musuh sudah tercapai maka pemain dapat lanjut ke level berikutnya
Kondisi Akhir	Pemain dapat melanjutkan permainan ke level berikutnya

f. Use Case Scenario menampilkan level 2

Use Case Scenario untuk *Use Case* menampilkan level 2 dijelaskan pada tabel di bawah ini.

Tabel 3.9. Use Case Scenario menampilkan level 2

Identifikasi	
Nomor	5
Nama	Memainkan level 2
Tujuan	Proses yang dilakukan pemain untuk menampilkan level 2
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pemain berada di halaman menu level
Kondisi Aktor	**Reaksi Sistem**
	1. Sistem menampilkan level 2
2. Mengendalikan karakter	
	3. Karakter bergerak
	4. Cek *power* dan darah
	5. Jika peluru musuh mengenai pemain maka darah pemain berkurang
	6. Jika musuh menabrak pemain maka darah pemain berkurang

	7. Jika darah pemain habis maka sistem menampilkan menu *game over* dimana pemain bisa memilih ulangi
	8. Jika pemain mendapatkan objek berbentuk kilat maka *power* bertambah
	9. Jika *power* sudah penuh maka akan muncul bantuan
10. Menyerang musuh	
	11. Perhitungan *score* saat berhasil menghancurkan musuh
	12. Target menghancurkan musuh akan bertambah saat pemain dapat menghancurkan musuh
13. Menyelesaikan level permainan	
	14. Jika target menghancurkan musuh sudah tercapai maka pemain dapat lanjut ke level berikutnya
Kondisi Akhir	Pemain dapat melanjutkan permainan ke level berikutnya

g. Use Case Scenario menampilkan level 3

Use Case Scenario untuk *Use Case* menampilkan level 3 dijelaskan pada tabel di bawah ini.

Tabel 3.10. Use Case Scenario menampilkan level 3

Identifikasi	
Nomor	6
Nama	Memainkan level 3
Tujuan	Proses yang dilakukan pemain untuk menampilkan level 3
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pemain berada di halaman menu level
Kondisi Aktor	**Reaksi Sistem**
	1. Sistem menampilkan level 3
2. Mengendalikan karakter	
	3. Karakter bergerak
	4. Cek *power* dan darah
	5. Jika peluru musuh mengenai pemain maka darah pemain berkurang
	6. Jika musuh menabrak pemain maka darah pemain berkurang
	7. Jika darah pemain habis maka sistem menampilkan menu *game over* dimana pemain bisa memilih ulangi
	8. Jika pemain mendapatkan objek berbentuk kilat maka *power* bertambah
	9. Jika *power* sudah penuh maka akan muncul bantuan
10. Menyerang musuh	
	11. Perhitungan *score* saat berhasil menghancurkan musuh

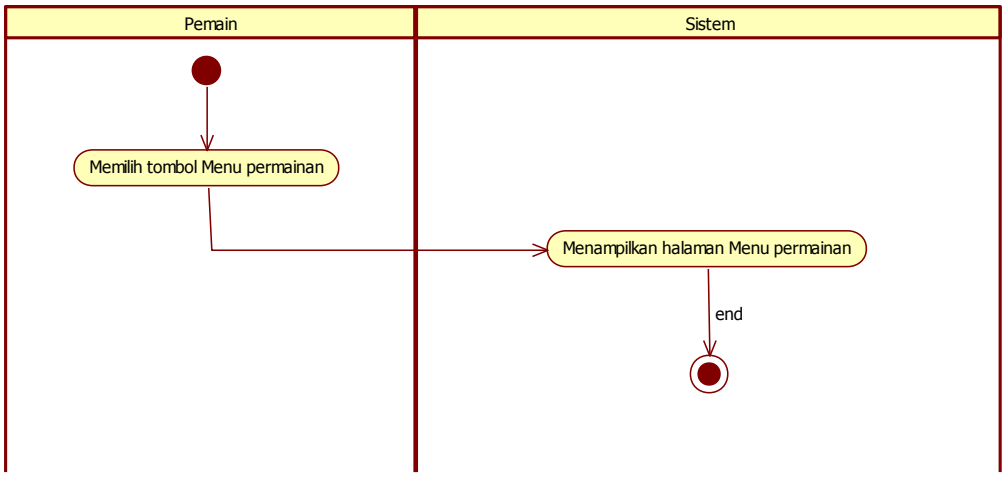
	12. Target menghancurkan musuh akan bertambah saat pemain dapat menghancurkan musuh
	13. Jika target menghancurkan musuh sudah tercapai maka sistem akan menampilkan sosok Raja atau musuh terkuat.
14. Menyelesaikan level permainan	
	15. Sistem menampilkan *ending scene*
	16. Sistem menampilkan tombol main dan keluar
17. Memilih tombol main	
	18. Sistem menampilkan level 1
19. Memilih tombol keluar	
	20. Sistem menutup aplikasi
Kondisi Akhir	Sistem menampilkan *ending scene* dan menampilkan tombol main dan keluar

3.1.5.2 Activity Diagram

Activity Diagram lebih memfokuskan diri pada eksekusi dan alur sistem dari pada bagaimana sistem itu dirakit. Diagram ini tidak hanya memodelkan *software* melainkan memodelkan model bisnis juga. *Activity Diagram* menunjukan aktivitas sistem dalam bentuk kumpulan aksi-aksi. Berikut ini *Activity Diagram* pada game Gatotkaca *Evolution*.

1. Activity Diagram memulai permainan

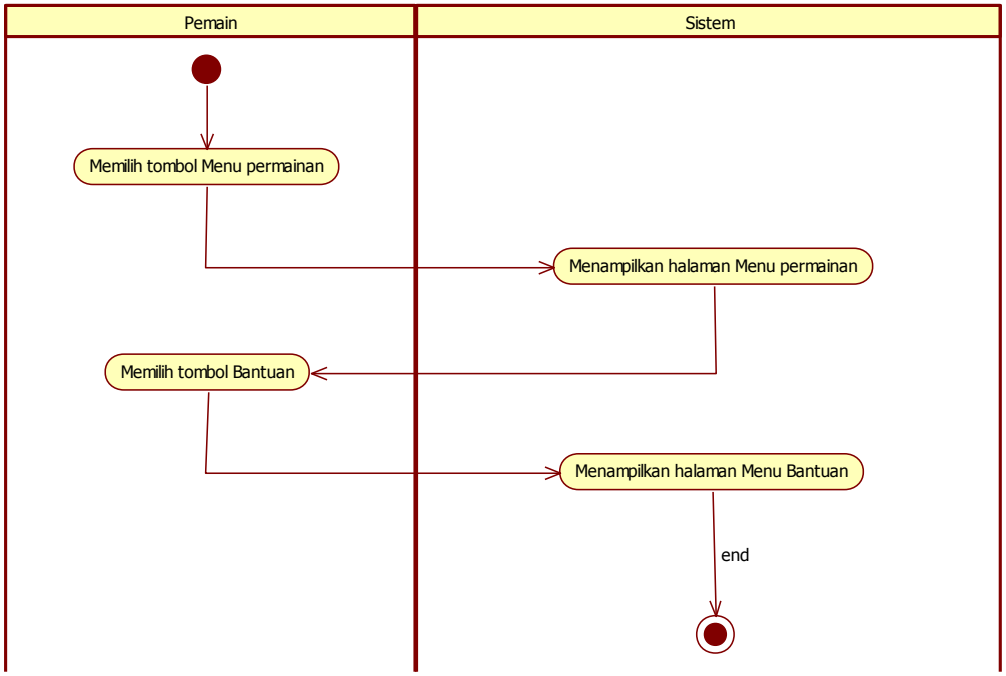
Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas memulai permainan yang dapat dilihat pada gambar berikut ini.



Gambar 3.5. Activity Diagram memulai permainan

2. Activity Diagram menampilkan menu bantuan

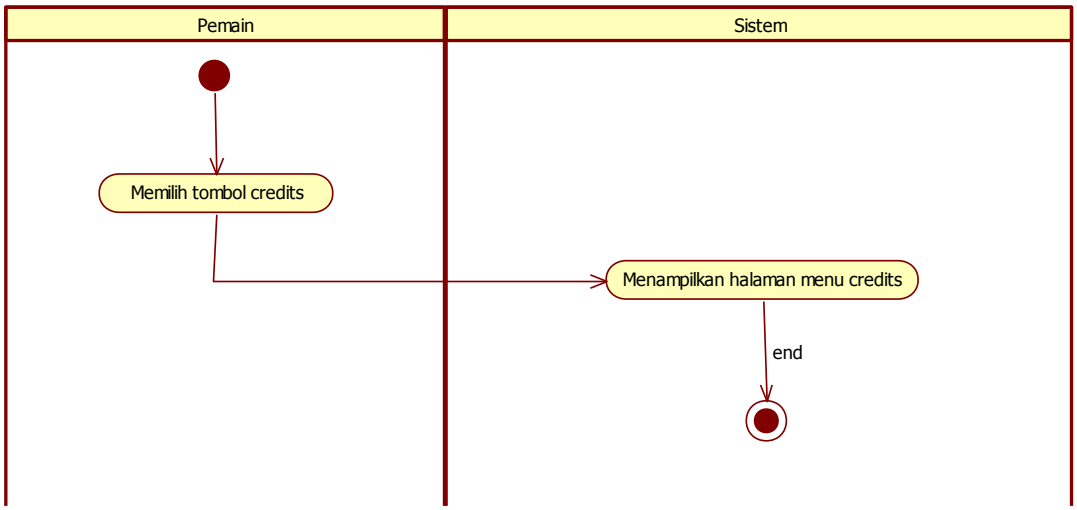
Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas menampilkan menu bantuan yang dapat dilihat pada gambar berikut ini.



Gambar 3.6. Activity Diagram menampilkan bantuan

3. Activity Diagram menampilkan menu credits

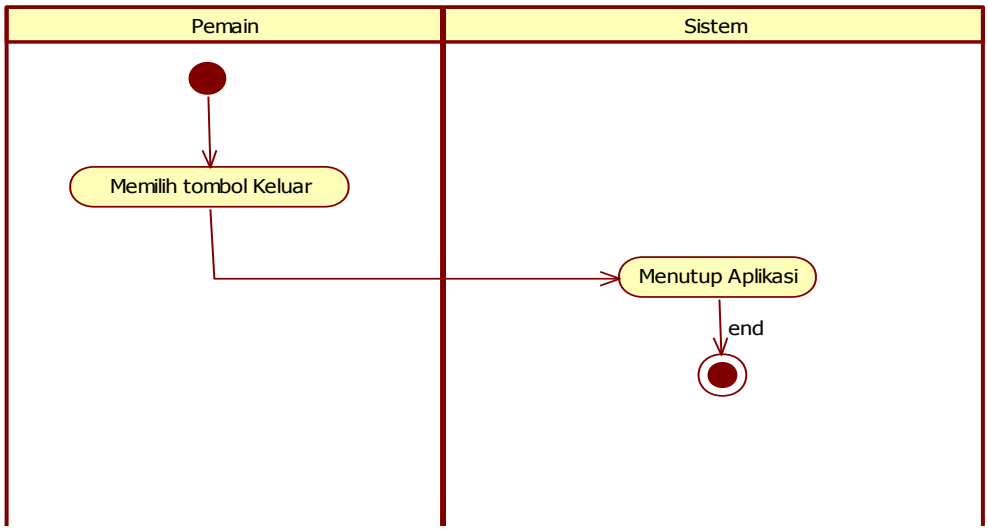
Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas menampilkan menu credits yang dapat dilihat pada gambar berikut ini.



Gambar 3.7. Activity Diagram menampilkan credits

4. Activity Diagram keluar dari permainan

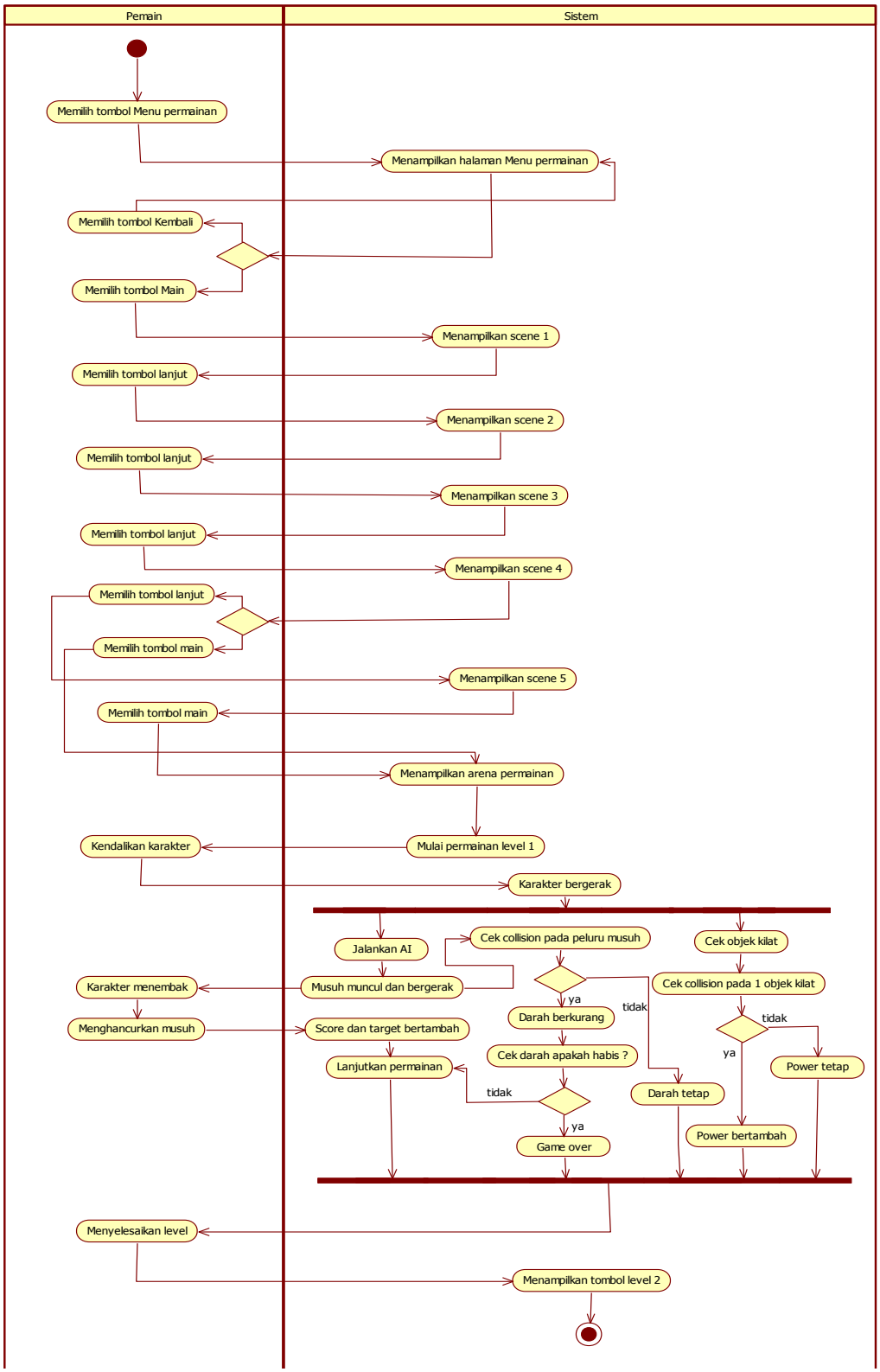
Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas keluar dari permainan yang dapat dilihat pada gambar berikut ini.



Gambar 3.8. Activity Diagram Keluar dari permainan

5. Activity Diagram memainkan level 1

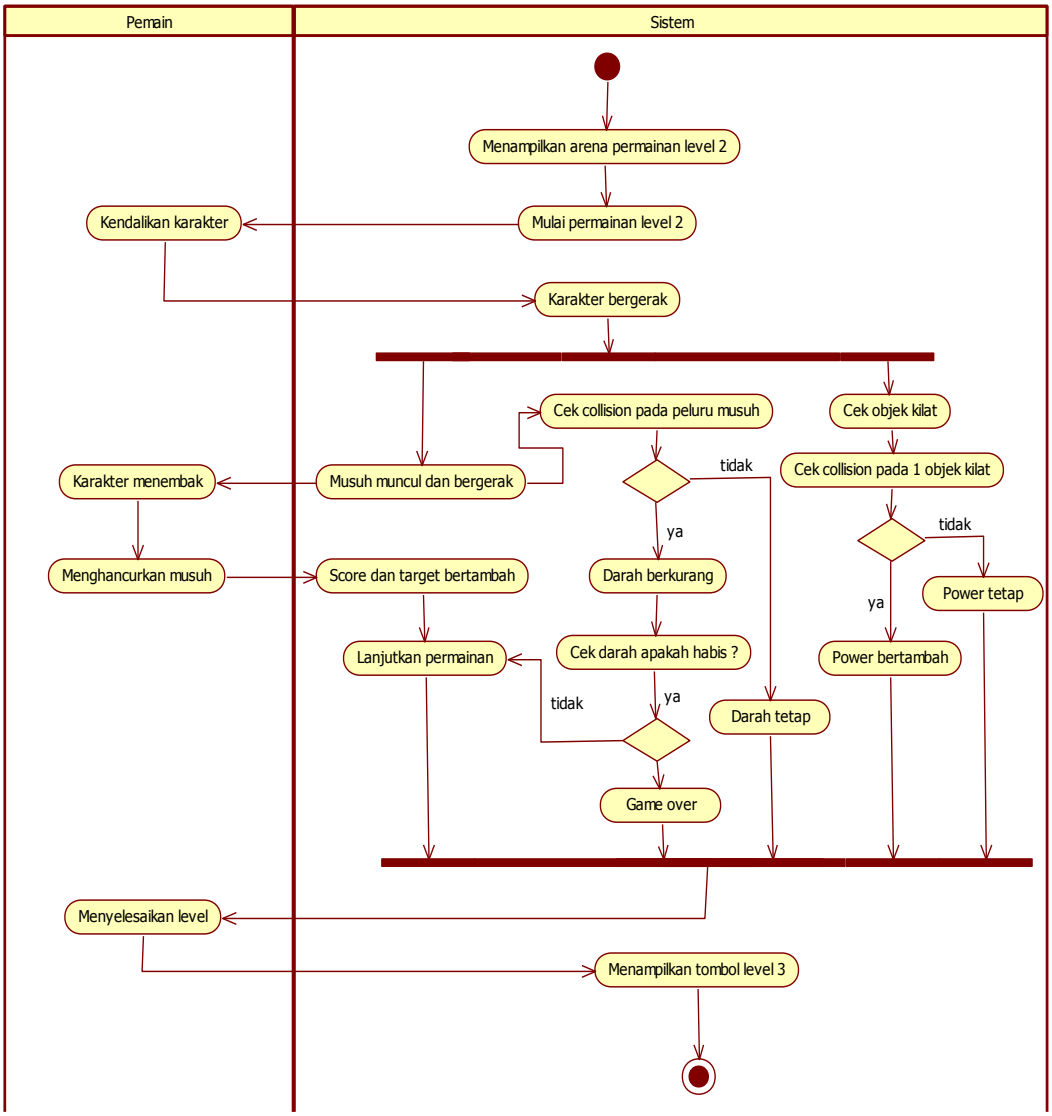
Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas memainkan level 1 yang dapat dilihat pada gambar berikut ini.



Gambar 3.9. Activity Diagram memainkan level 1

6. Activity Diagram memainkan level 2

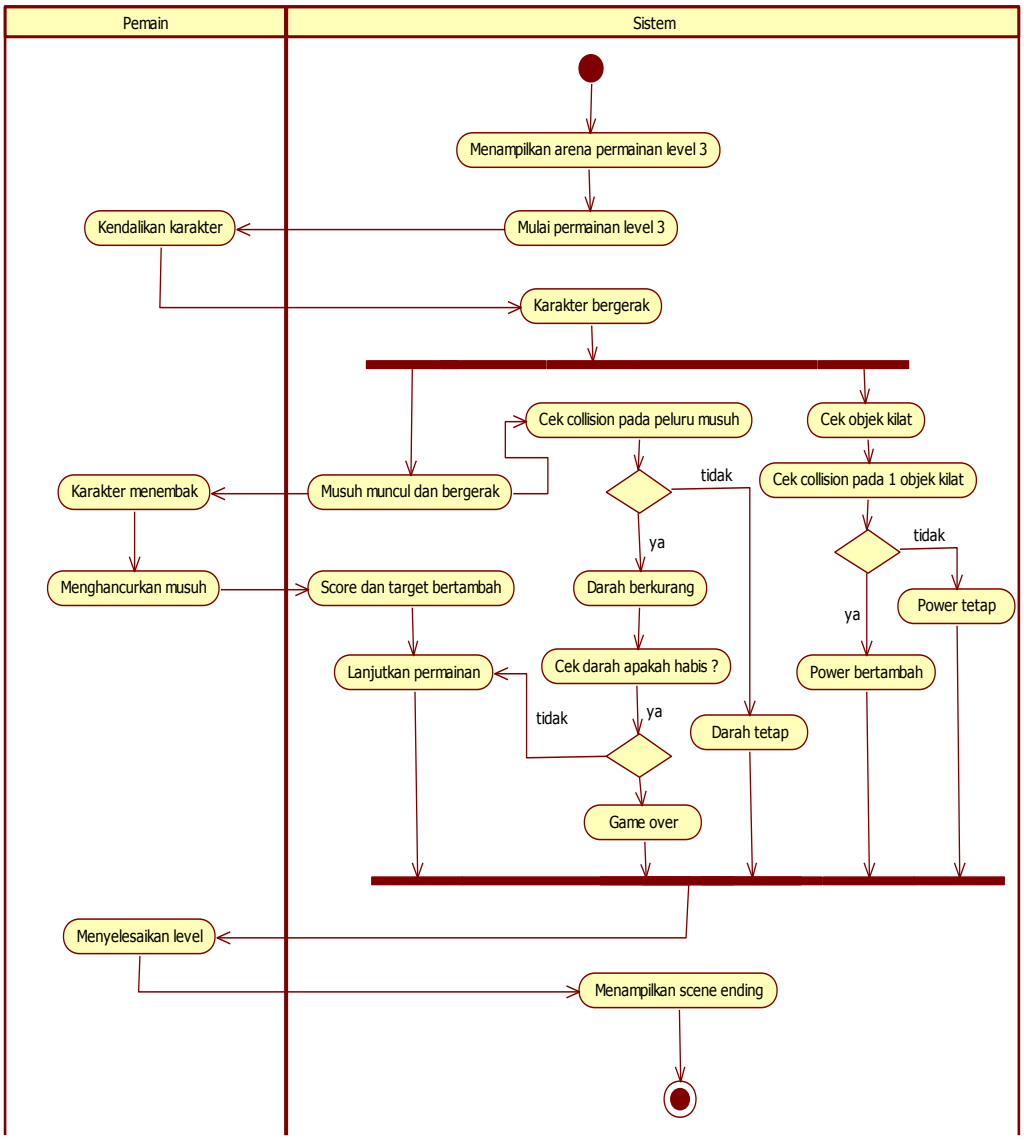
Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas memainkan level 2 yang dapat dilihat pada gambar berikut ini.



Gambar 3.10. Activity Diagram memainkan level 2

7. Activity Diagram memainkan level 3

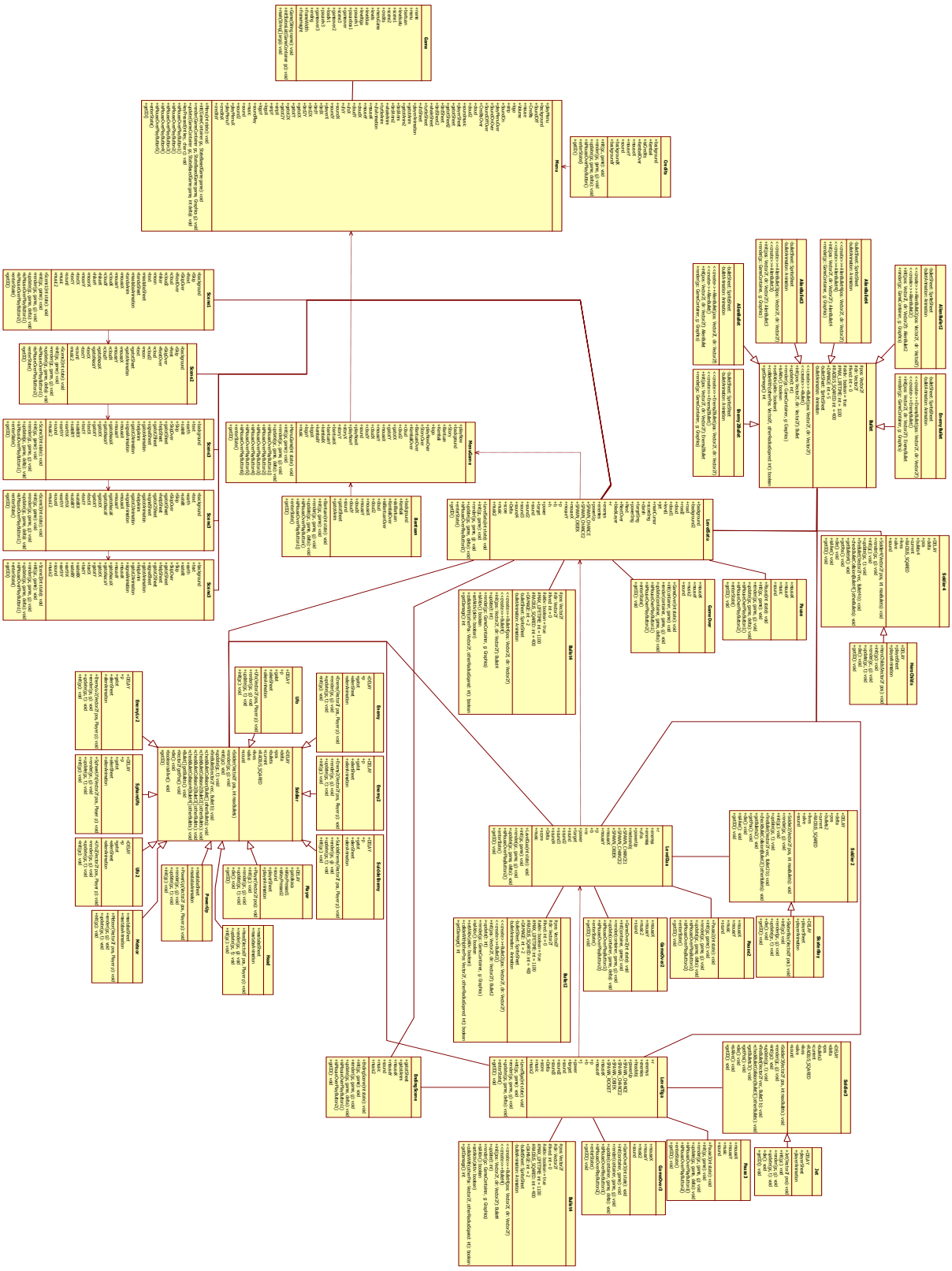
Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas memainkan level 3 yang dapat dilihat pada gambar berikut ini.



Gambar 3.11. Activity Diagram memainkan level 3

3.1.5.3 Class Diagram

Class Diagram adalah inti dari proses pemodelan objek. *Class Diagram* menggambarkan struktur dan hubungan antar objek-objek yang ada pada sistem. Struktur itu meliputi atribut dan *method* yang ada pada masing-masing *class*. Di bawah ini merupakan *Class Diagram* game Gatotkaca *Evolution*.



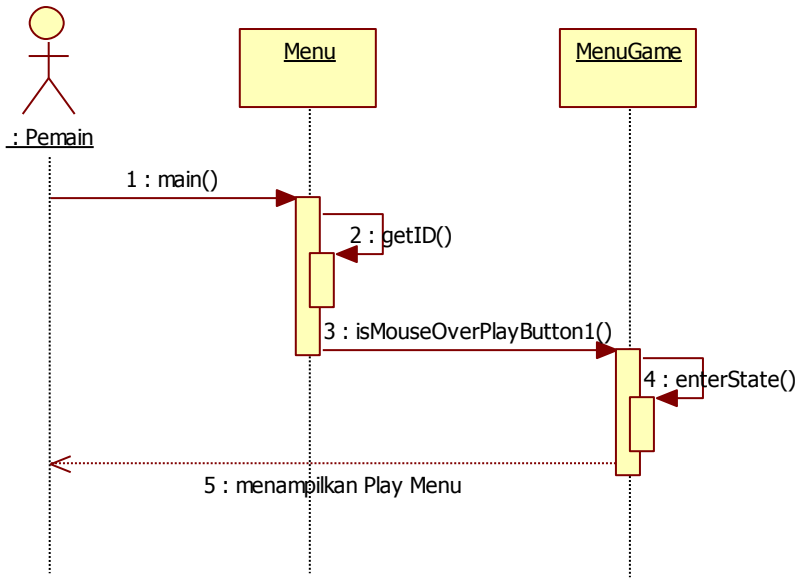
Gambar 3.12. Diagram Kelas

3.1.5.4 Sequence Diagram

Sequence Diagram menggambarkan interaksi antar masing-masing objek pada setiap *use case* dalam urutan waktu. Interaksi ini berupa pengiriman serangkaian data antar objek-objek yang saling berinteraksi. *Sequence Diagram* pada game Gatotkaca *Evolution* terdiri dari *sequence diagram* memainkan permainan, *sequence diagram* menampilkan *credits*, *sequence diagram* menghentikan aplikasi, *sequence diagram* menampilkan menu level, *sequence diagram* menampilkan *story*, *sequence diagram* menampilkan menu bantuan, *sequence diagram* menampilkan level 1, *sequence diagram* menampilkan level 2 dan *sequence diagram* menampilkan level 3. Berikut penjelasan dari masing-masing *sequence diagram* :

1. **Sequence Diagram memulai permainan**

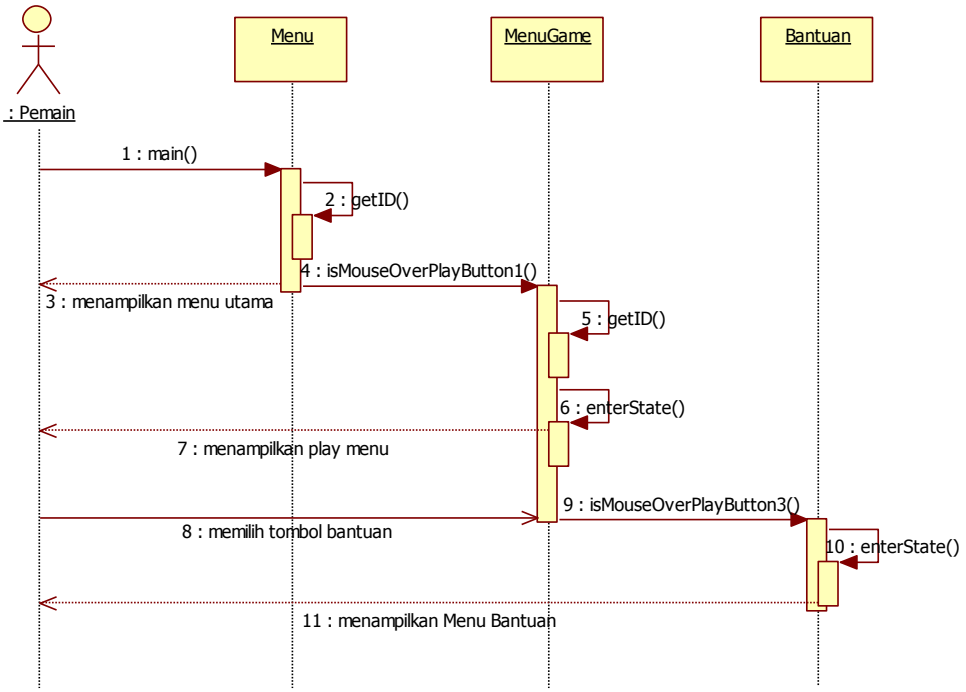
   Diagram ini menjelaskan interaksi antar partisipan di dalam aktivitas memainkan permainan dalam urutan waktu. Untuk lebih jelasnya dapat dilihat pada gambar di bawah ini.



Gambar 3.13. Sequence Diagram memulai permainan

2. Sequence Diagram menampilkan bantuan

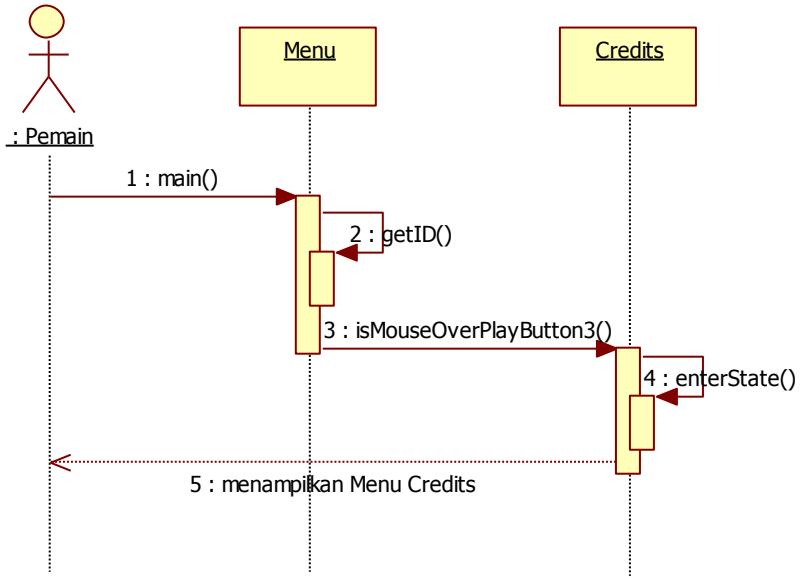
Diagram ini menjelaskan interaksi antar partisipan di dalam aktivitas menampilkan bantuan dalam urutan waktu. Untuk lebih jelasnya dapat dilihat pada gambar di bawah ini.



Gambar 3.14. Sequence Diagram menampilkan bantuan

3. Sequence Diagram menampilkan credits

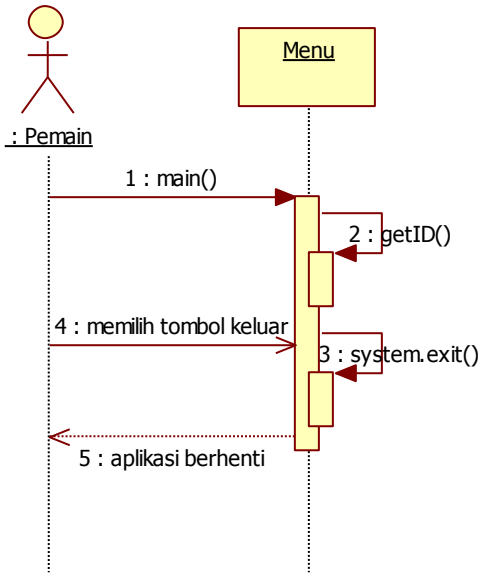
Diagram ini menjelaskan interaksi antar partisipan di dalam aktivitas menampilkan *credits* dalam urutan waktu. Untuk lebih jelasnya dapat dilihat pada gambar di bawah ini.



Gambar 3.15. Sequence Diagram menampilkan credits

4. Sequence Diagram keluar dari aplikasi

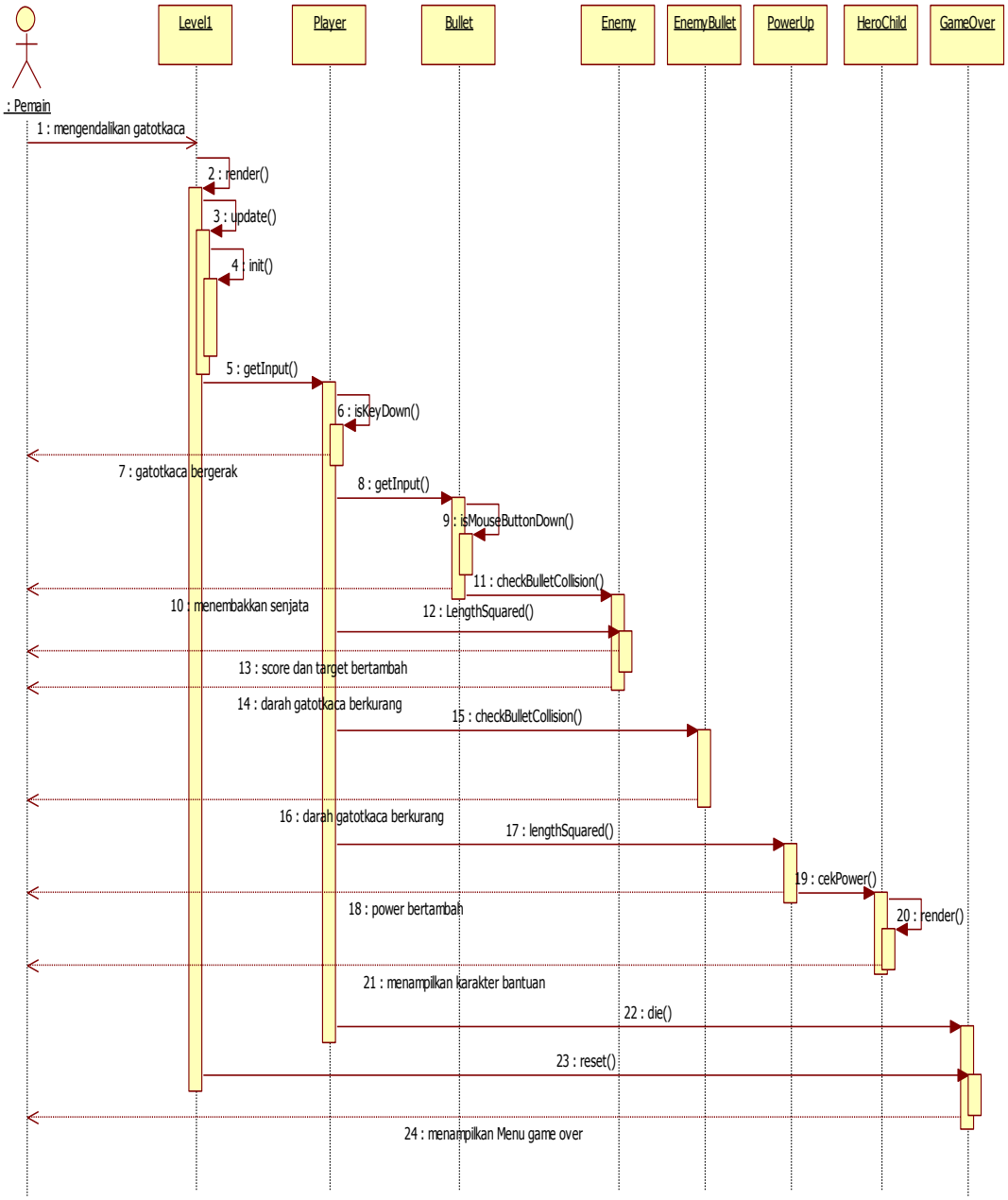
Diagram ini menjelaskan interaksi antar partisipan di dalam aktivitas keluar dari aplikasi dalam urutan waktu. Untuk lebih jelasnya dapat dilihat pada gambar di bawah ini.



Gambar 3.16. Sequence Diagram keluar dari aplikasi

5. Sequence Diagram memainkan level 1

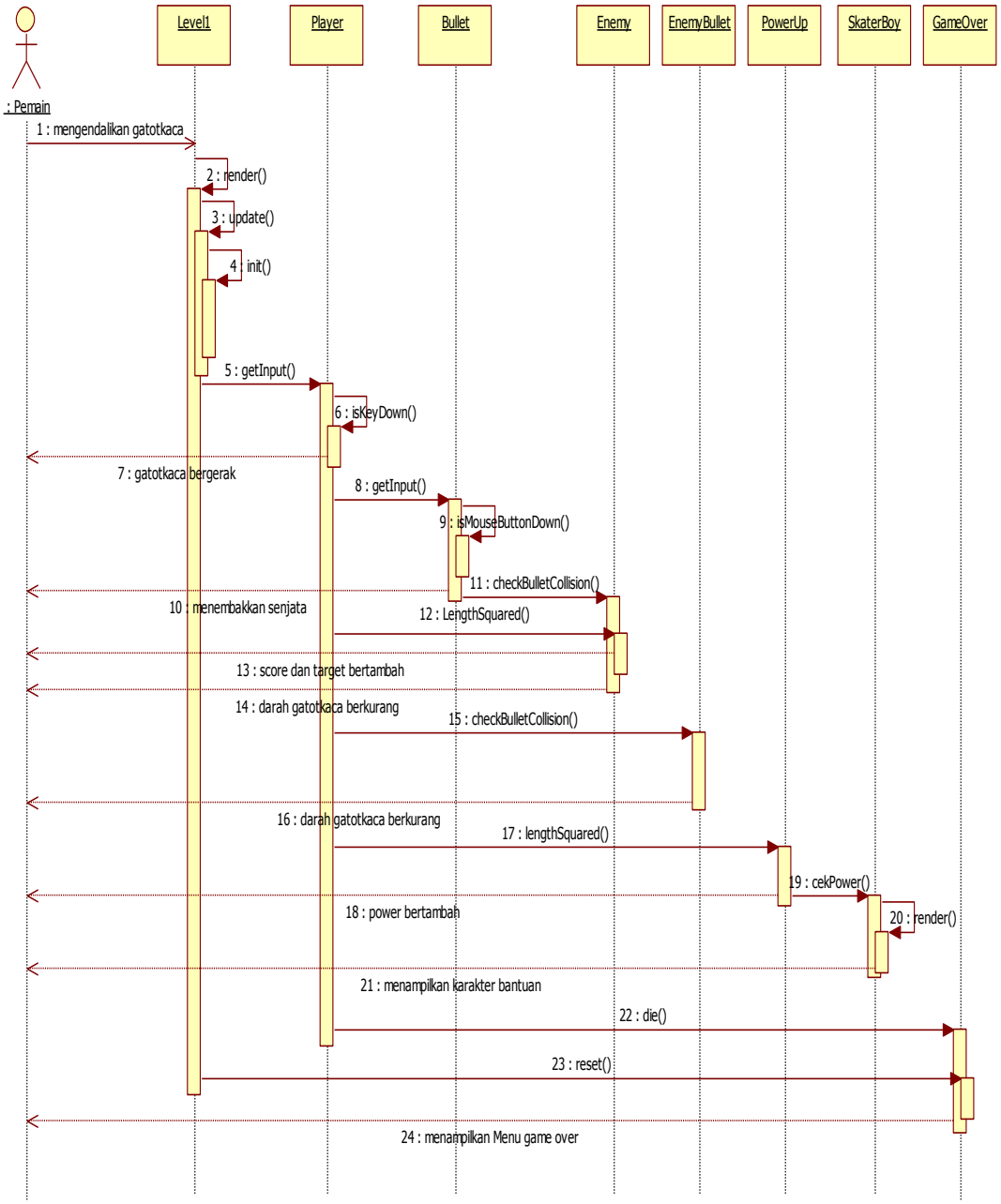
Diagram ini menjelaskan interaksi antar partisipan di dalam aktivitas memainkan level 1 dalam urutan waktu. Untuk lebih jelasnya dapat dilihat pada gambar di bawah ini.



Gambar 3.17. Sequence Diagram memainkan Level 1

6. Sequence Diagram memainkan level 2

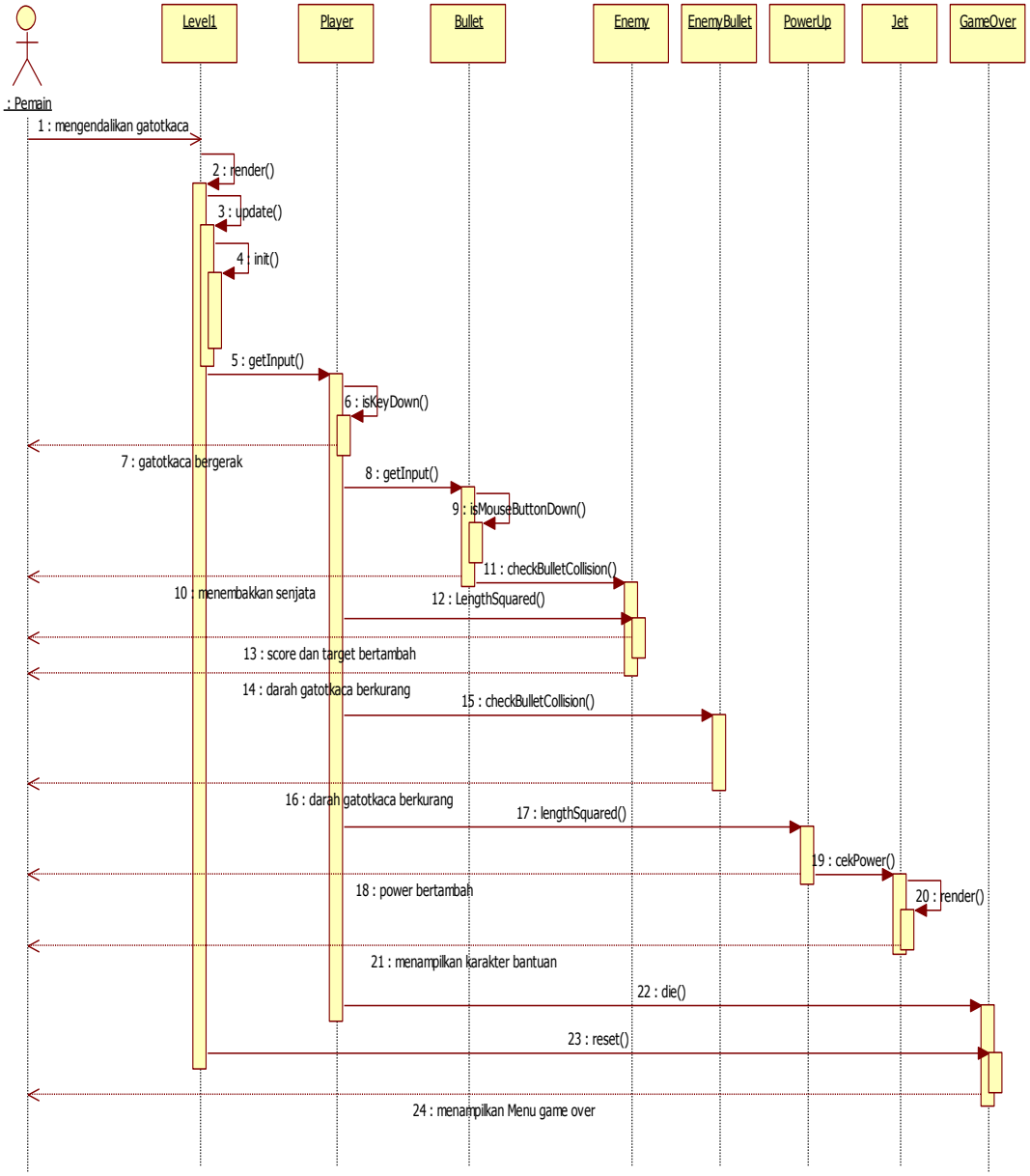
Diagram ini menjelaskan interaksi antar partisipan di dalam aktivitas memainkan level 2 dalam urutan waktu. Untuk lebih jelasnya dapat dilihat pada gambar di bawah ini.



Gambar 3.18. Sequence Diagram memainkan Level 2

7. Sequence Diagram memainkan level 3

Diagram ini menjelaskan interaksi antar partisipan di dalam aktivitas memainkan level 2 dalam urutan waktu. Untuk lebih jelasnya dapat dilihat pada gambar di bawah ini.



Gambar 3.19. Sequence Diagram memainkan Level 3

3.2 Perancangan Sistem

Perancangan sistem merupakan tahapan dari siklus pengembangan sistem didefinisikan dari kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi yang menggambarkan bagaimana suatu sistem dibentuk, yang dapat berupa penggambaran, perancangan, dan pembuatan sketsa. Di dalam bagian ini akan dijelaskan perancangan komponen permainan, perancangan antarmuka, perancangan pesan dan jaringan semantik dari *game* Gatotkaca *Evolution*.

3.2.1 Perancangan Komponen Permainan

Di dalam bagian ini akan dijelaskan karakter dan jenis senjata yang ditembakkan baik oleh karakter pemain dan karakter musuh yang terdapat pada *game* Gatotkaca *Evolution*.

3.2.1.1 Karakter *Game*

Di dalam bagian ini akan dijelaskan karakter-karakter yang terdapat pada *game* Gatotkaca *Evolution* seperti yang bisa dilihat pada tabel berikut ini.

Tabel 3.11. Tabel Karakter pada Game

No	Karakter	Keterangan
1		Gatotkaca merupakan karakter utama yang dapat terbang dan menembakkan bola api, untuk menambah power maka pemain harus mengambil *item*
2		Karakter ini merupakan karakter bantuan yang akan keluar di level 1 apabila *power* sudah penuh

3		Karakter ini merupakan karakter bantuan yang akan keluar di level 2 apabila *power* sudah penuh
4		Karakter ini merupakan karakter bantuan yang akan keluar di level 3 apabila *power* sudah penuh
5		Karakter ini merupakan karakter musuh Robokaca yang akan keluar di level 1, memiliki *health point* sebesar 20, apabila hancur menambah *score* 20
6		Karakter ini merupakan karakter musuh RoboCakil yang akan keluar di level 1, memiliki *health point* sebesar 20, apabila hancur menambah *score* 10
7		Karakter ini merupakan karakter musuh Rahwana yang akan keluar di level 2, memiliki *health point* sebesar 30, apabila hancur menambah *score* 20

8		Karakter ini merupakan karakter musuh Kurawa yang akan keluar di level 2, memiliki *health point* sebesar 30, apabila hancur menambah *score* 10
9		Karakter ini merupakan karakter musuh yang akan keluar di level 2, karakter musuh ini tidak menembak tetapi menghancurkan diri ke arah Gatotkaca, apabila hancur menambah *score* 5
10		Karakter ini merupakan karakter musuh ButoCakil yang akan keluar di level 3, memiliki *health point* sebesar 30, apabila hancur menambah *score* 40
11		Karakter ini merupakan karakter musuh raksasa yang akan keluar di level 3, memiliki *health point* sebesar 30, apabila hancur menambah *score* 30

12		Karakter ini merupakan karakter musuh terkuat yang akan keluar di level 3, memiliki *health point* sebesar 300

3.2.1.2 Jenis Senjata

Jenis senjata yang digunakan pada game Gatotkaca *Evolution* meliputi jenis senjata yang digunakan oleh Gatotkaca, karakter bantuan dan juga musuh. Setiap jenis senjata yang dikeluarkan oleh musuh, memiliki kekuatan yang berbeda-beda. Berikut ini tabel jenis senjata dan keterangannya.

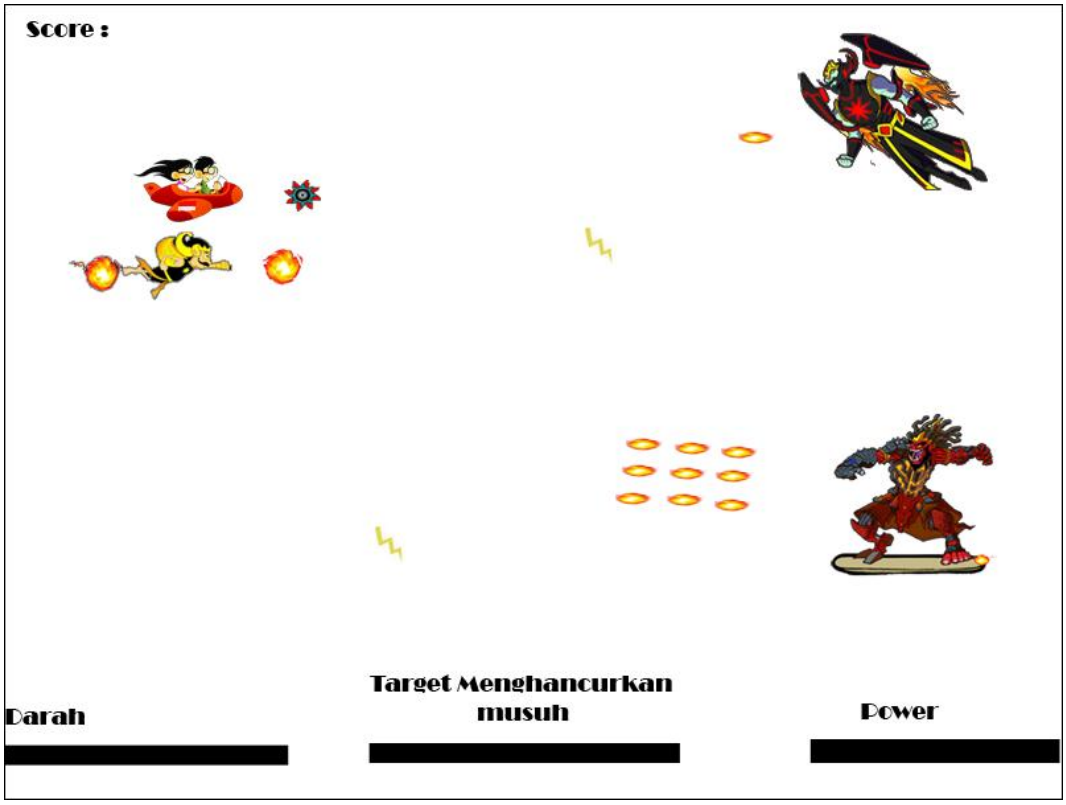
Tabel 3.12. Tabel Jenis Senjata

No	Jenis Senjata	Keterangan
1		Jenis senjata yang dimiliki oleh Gatotkaca apabila mengenai musuh akan mengurangi *health point* musuh sebesar 5
2		Jenis senjata yang dimiliki oleh karakter bantuan di level 1, apabila mengenai musuh akan mengurangi *health point* musuh sebesar 2
3		Jenis senjata yang dimiliki oleh karakter bantuan di level 1, apabila mengenai musuh akan mengurangi *health point* musuh sebesar 2
4		Jenis senjata yang dimiliki oleh karakter bantuan di level 1, apabila mengenai musuh akan mengurangi *health point* musuh sebesar 2
5		Jenis senjata yang dimiliki oleh karakter musuh Robokaca dan

		RoboCakil di level 1, apabila mengenai pemain akan mengurangi *health point* pemain sebesar 1
6		Jenis senjata yang dimiliki oleh karakter musuh Kurawa di level 2, apabila mengenai pemain akan mengurangi *health point* pemain sebesar 3
7		Jenis senjata yang dimiliki oleh karakter musuh Rahwana di level 2, apabila mengenai pemain akan mengurangi *health point* pemain sebesar 2
8		Jenis senjata yang dimiliki oleh karakter musuh ButoCakil di level 3, apabila mengenai pemain akan mengurangi *health point* pemain sebesar 3
9		Jenis senjata yang dimiliki oleh karakter musuh Raksasa di level 3, apabila mengenai pemain akan mengurangi *health point* pemain sebesar 3
10		Jenis senjata yang dimilki oleh karakter musuh terkuat di level 3, apabila mengenai pemain akan mengurangi *health point* pemain sebesar 7

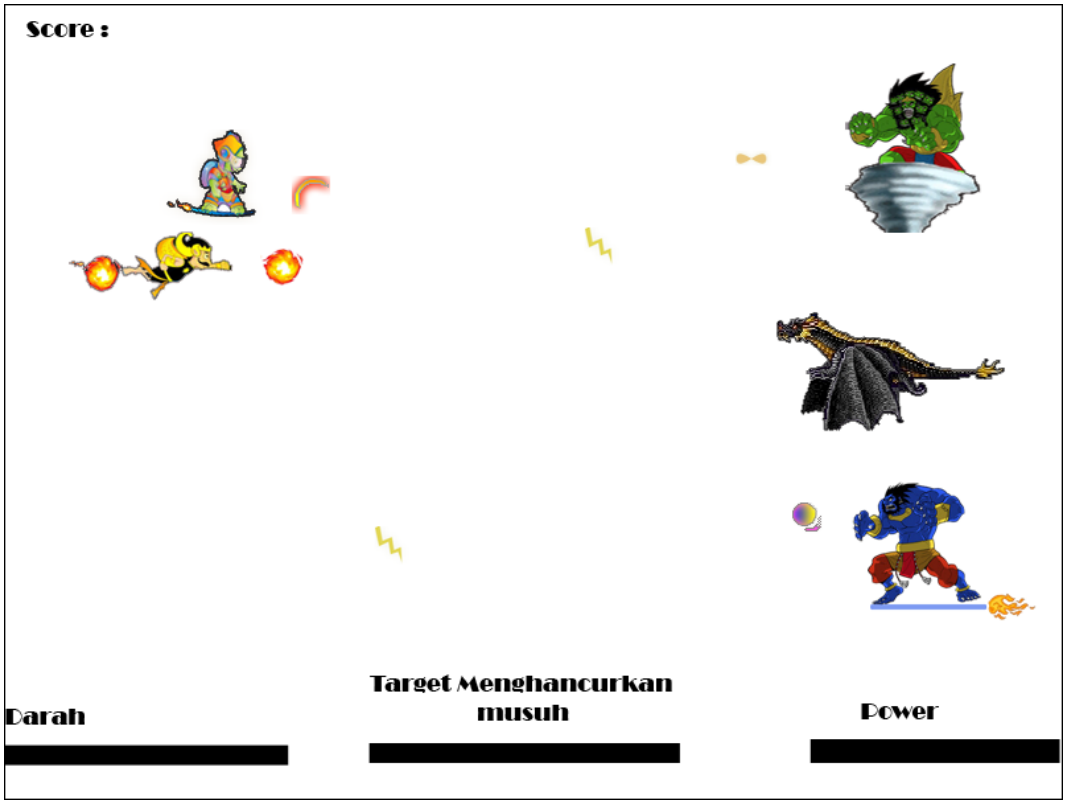
3.2.1.3 Storyboard

Storyboard adalah sketsa gambar yang disusun berurutan sesuai dengan naskah, dengan *storyboard* ide cerita dapat disampaikan kepada orang lain dengan lebih mudah, karena dapat menggiring khayalan seseorang mengikuti gambar-gambar yang tersaji sehingga menghasilkan persepsi yang sama. Pada bagian ini akan dijelaskan *storyboard* dari *Game* Gatotkaca *Evolution*.



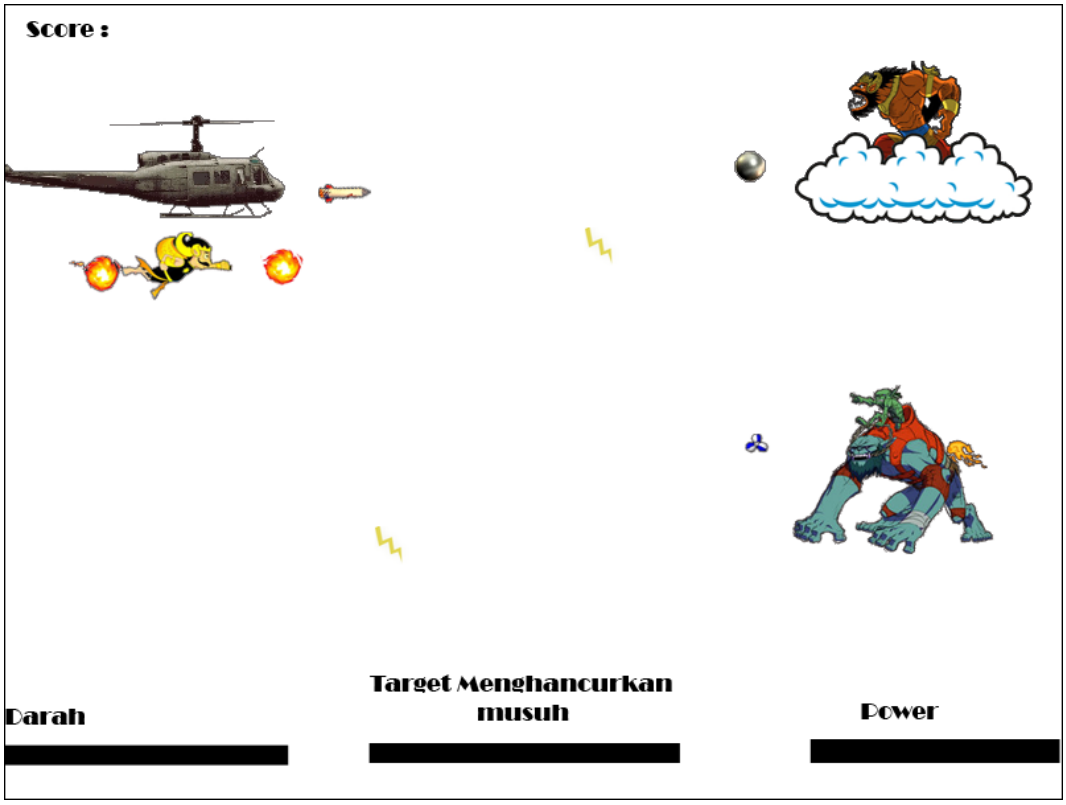
Gambar 3.20. Storyboard Level 1

Pada level 1 pemain harus menghancurkan musuh sesuai target yang ada. Terdapat 2 jenis musuh pada level ini dengan kekuatan yang berbeda-beda pula. Untuk memanggil bantuan, pemain harus mengambil objek berbentuk kilat agar power bertambah dan apabila power sudah penuh maka bantuan akan otomatis keluar. Latar pada level ini adalah pantai Ancol (Jakarta).

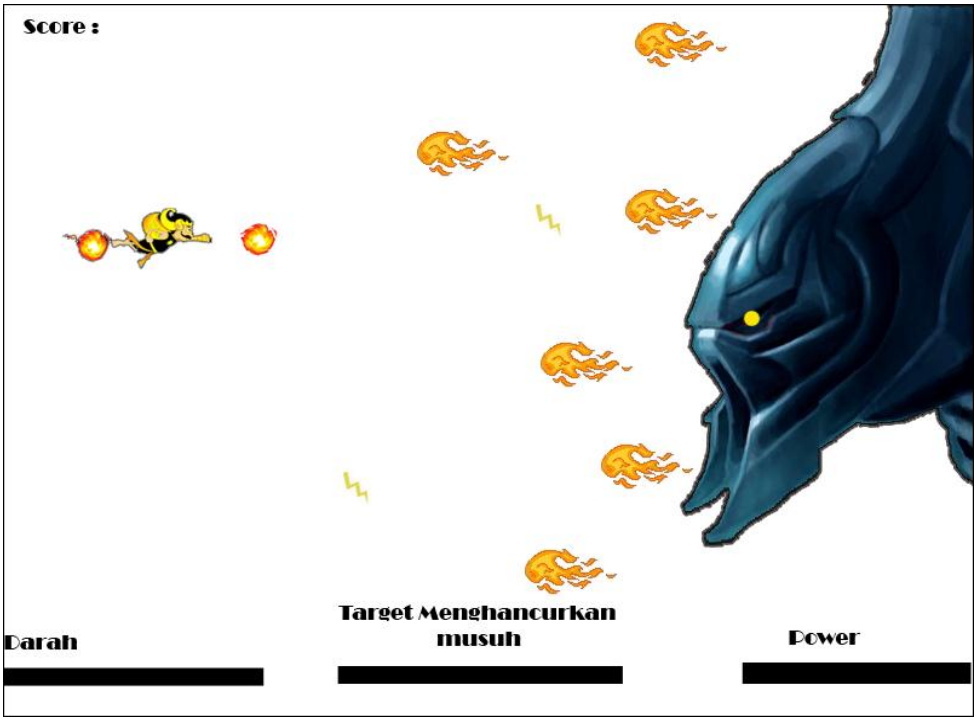


Gambar 3.21. Storyboard Level 2

Pada Level 2 misi pemain sama seperti pada Level 2 yaitu menghancurkan musuh sesuai target. Namun pada Level ini terdapat 3 jenis musuh yang memiliki kekuatan berbeda-beda, salah satunya adalah naga yang memiliki kecepatan yang lebih dari musuh lainnya. Selain itu, pada level ini bantuannya pun berbeda jenisnya dari level 1. Latar pada level 2 ini berada di Candi Borobudur (Yogyakarta).



Gambar 3.22. Storyboard Level 3

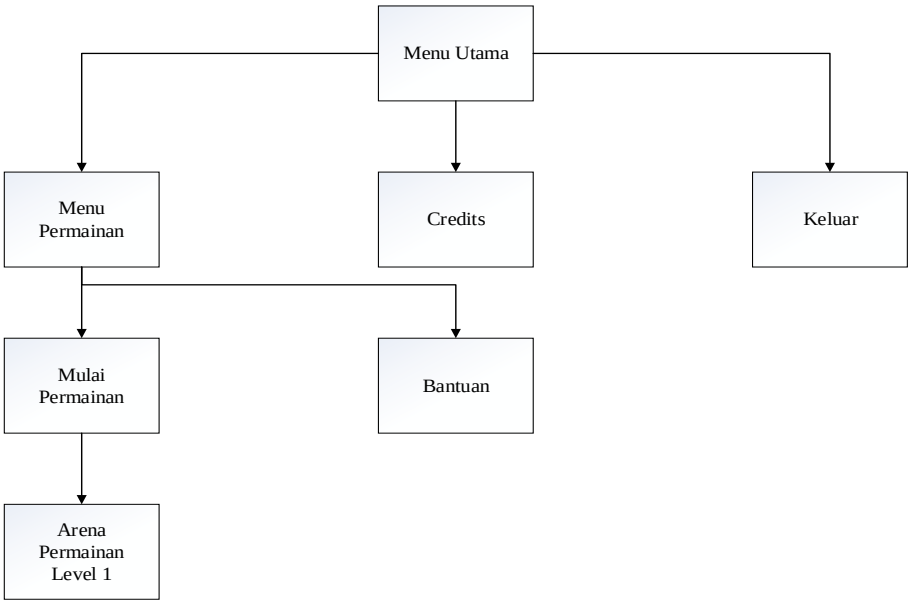


Gambar 3.23. Storyboard Level 3 Raja

Pada Level 3 misi pemain sama seperti pada Level 1 dan Level 2 yaitu menghancurkan musuh sesuai target. Level 3 adalah level terakhir pada game Gatotkaca *Evolution*. Pada Level ini terdapat 2 jenis musuh yang memiliki kekuatan berbeda-beda dan di akan terdapat karakter musuh terkuat yang harus dihancurkan. Selain itu, pada level ini bantuannya pun berbeda jenisnya dari level 1 dan Level 2. Latar pada level 3 ini berada di Gedung Sate (Bandung).

3.2.2 Perancangan Struktur Menu

Untuk mempermudah dalam pembuatan sistem, maka diperlukan perancangan struktur menu program yang akan dibangun. Perancangan struktur menu program ini membantu dalam merancang bagian-bagian dari sistem yang sebenarnya dan untuk mengetahui bagian mana yang nantinya akan diakses terlebih dahulu setelah program selesai dibuat. Perancangan arsitektur menu pada aplikasi permainan ini dapat dilihat pada gambar

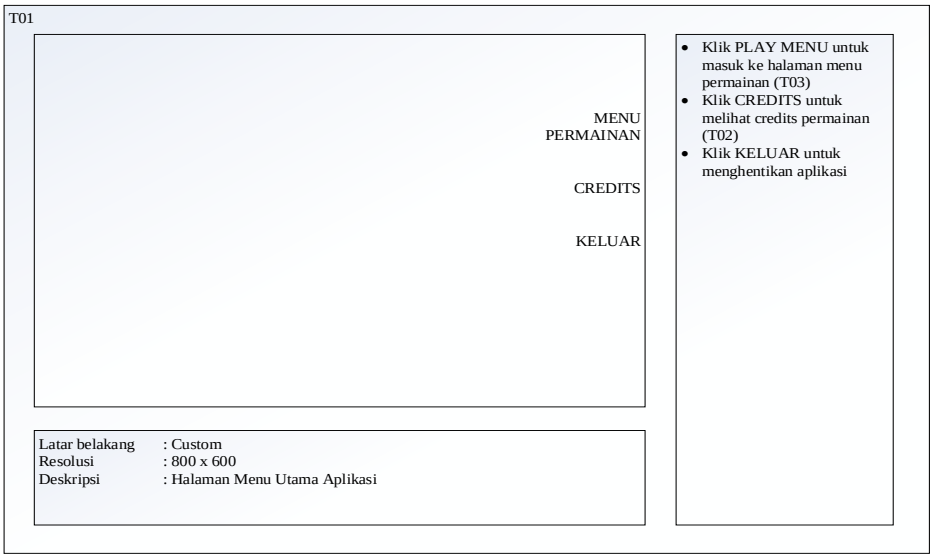


Gambar 3.24. Struktur Menu

3.2.3 Perancangan Antar Muka

1. Perancangan Antarmuka Menu Utama

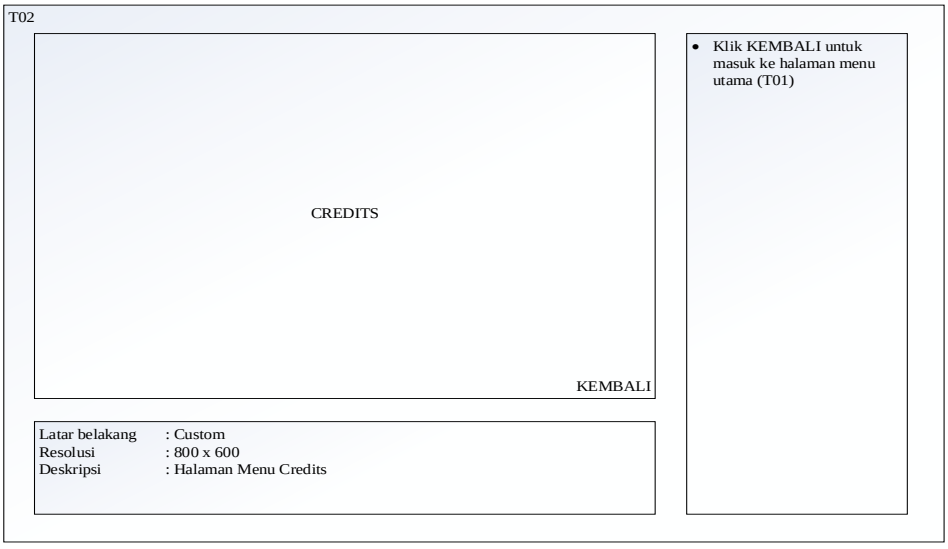
Perancangan antarmuka menu utama pada aplikasi permainan dapat dilihat pada gambar



Gambar 3.25. Antarmuka Menu Utama

2. Perancangan Antarmuka Credits

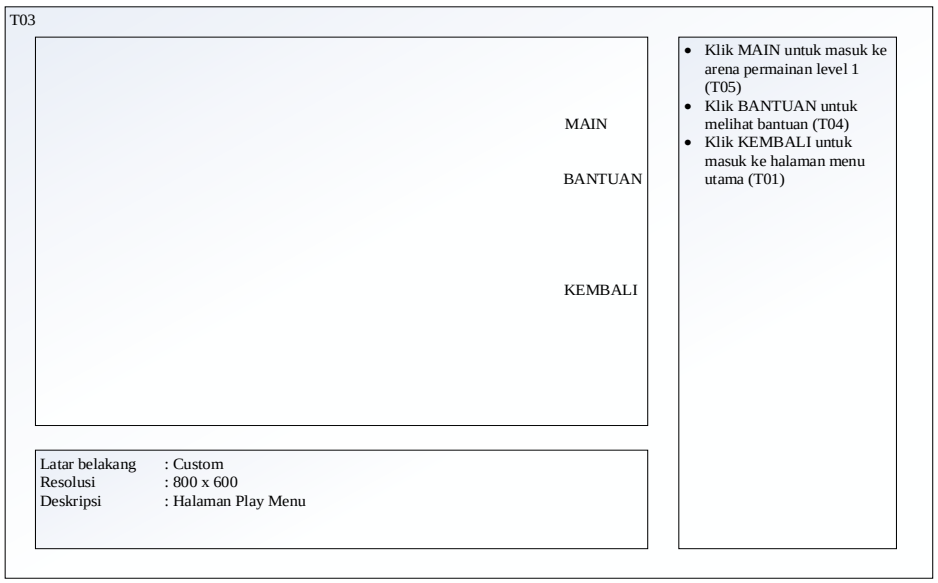
Perancangan antarmuka Credits pada aplikasi permainan ini dapat dilihat pada gambar



Gambar 3.26. Antarmuka Menu Credits

3. Perancangan Antarmuka Play Menu

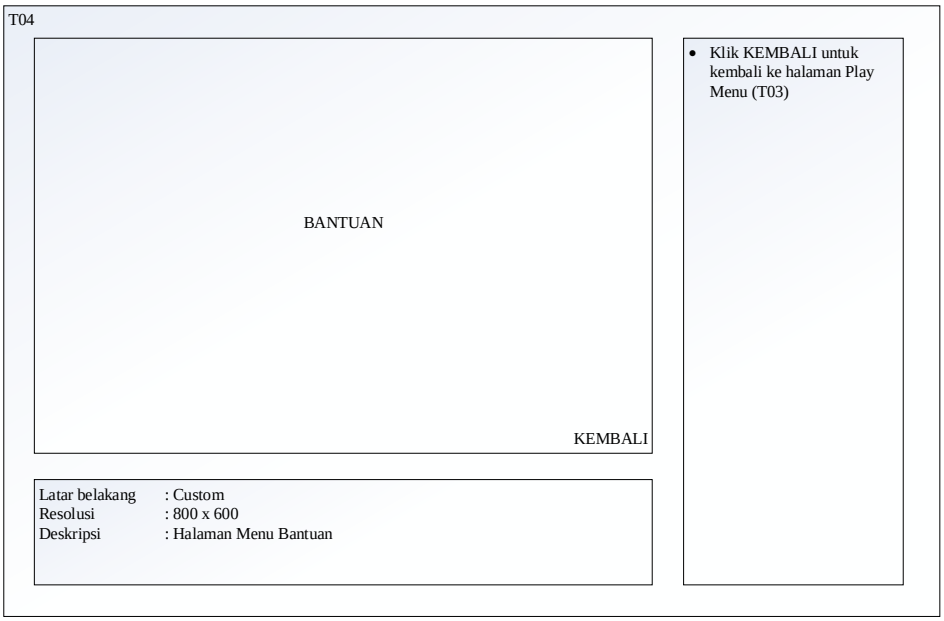
Perancangan antarmuka Play Menu pada aplikasi permainan ini dapat dilihat pada gambar



Gambar 3.27. Antarmuka Play Menu

4. Perancangan Antarmuka Bantuan

Perancangan antarmuka Bantuan pada aplikasi permainan ini dapat dilihat pada gambar

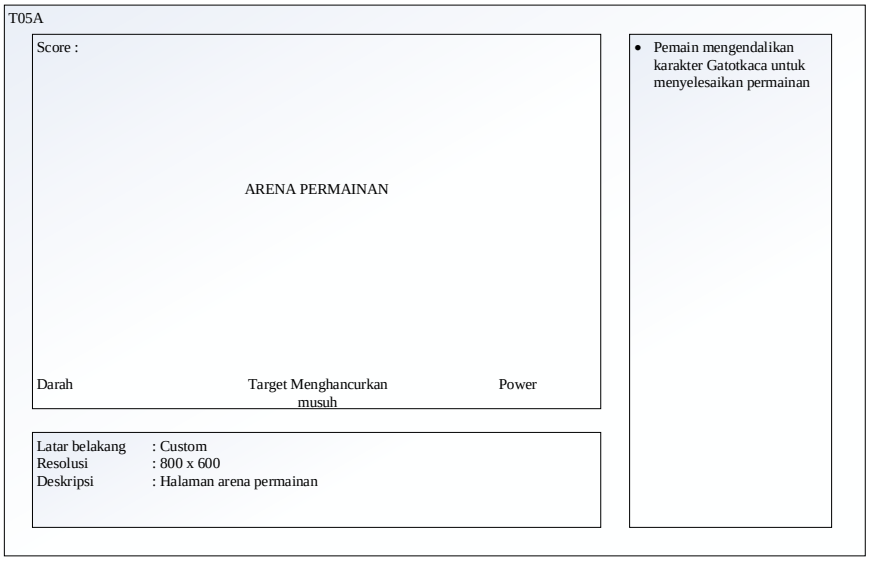


Gambar 3.28. Antarmuka Menu Bantuan

5. Perancangan Antarmuka Arena Permainan

a. Antarmuka Arena Permainan

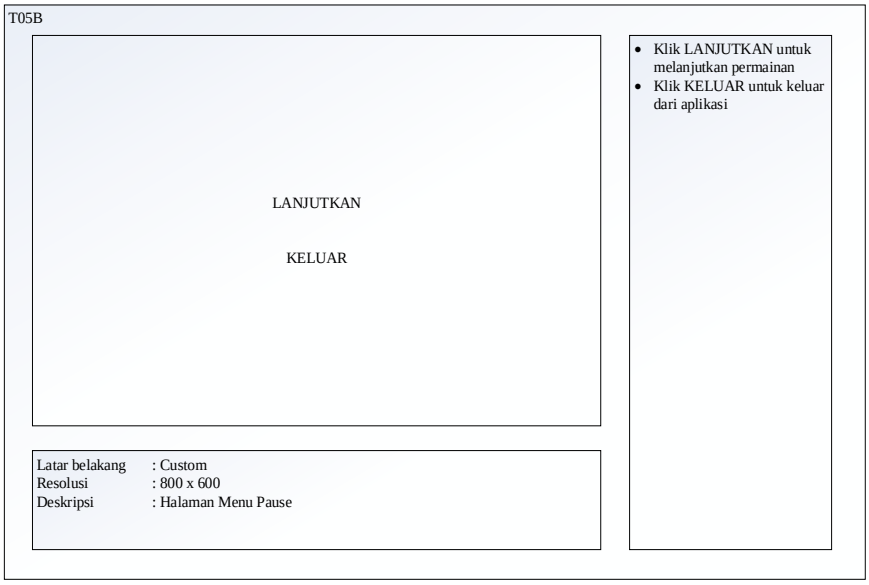
Perancangan antarmuka Arena Permainan pada aplikasi permainan ini dapat dilihat pada gambar



Gambar 3.29. Antarmuka Arena Permainan

b. Antarmuka Menu Pause

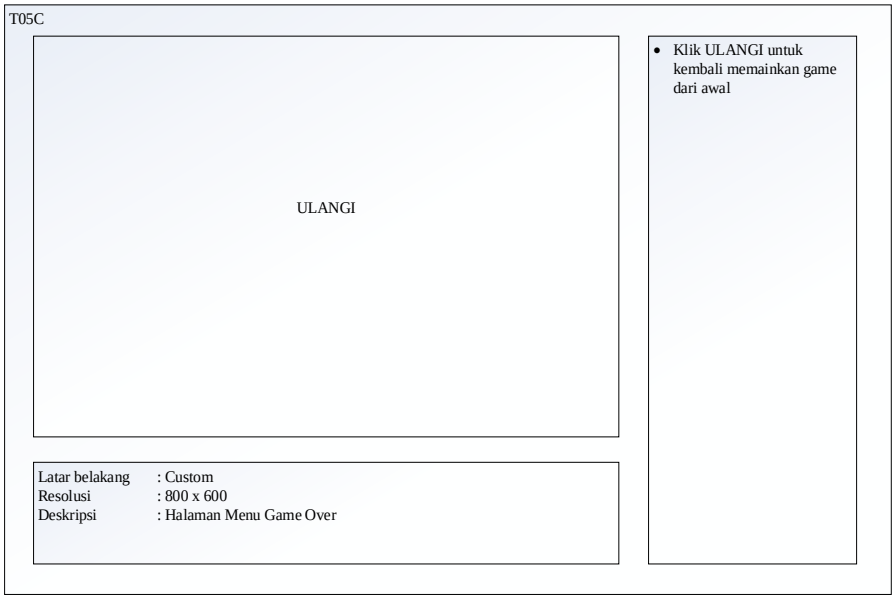
Perancangan antarmuka Menu Pause pada aplikasi permainan ini dapat dilihat pada gambar



Gambar 3.30. Antarmuka Menu Pause

c. Antarmuka Menu Game Over

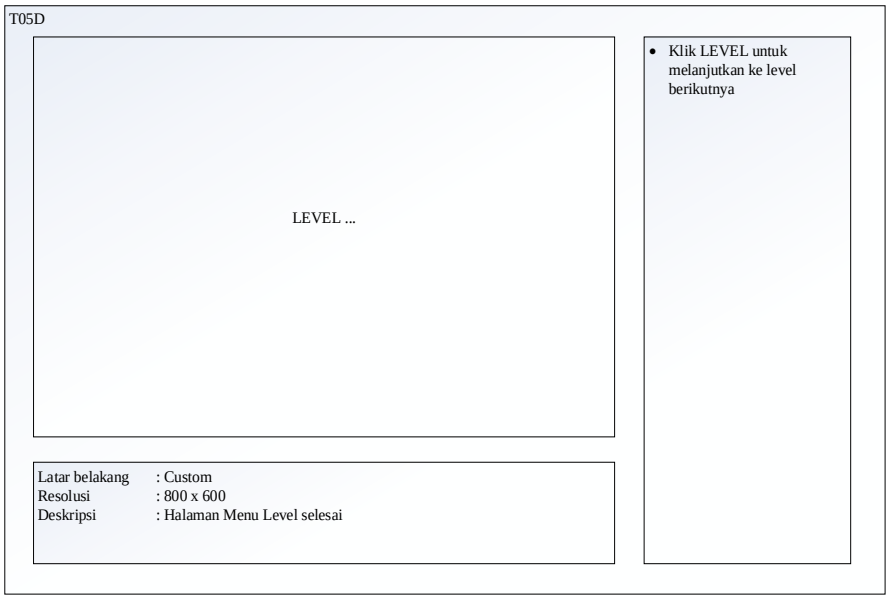
Perancangan antarmuka Menu Game Over pada aplikasi permainan ini dapat dilihat pada gambar



Gambar 3.31. Antarmuka Menu Game Over

d. Antarmuka Menu Level Selesai

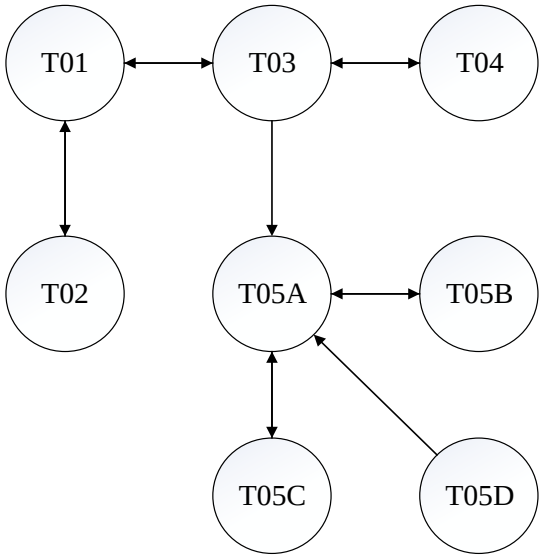
Perancangan antarmuka Menu Level Selesai pada aplikasi permainan ini dapat dilihat pada gambar



Gambar 3.32. Antarmuka Menu Level Selesai

3.2.4 Jaringan Semantik

Gambaran pengetahuan grafis yang menunjukan hubungan antar berbagai objek, terdiri dari lingkaran-lingkaran yang dihubungkan dengan anak panah yang menunjukan objek dan informasi tentang objek-objek tersebut disebut jaringan semantik. Jaringan semantik dari *game* Gatotkaca *Evolution* bisa dilihat pada gambar



Gambar 3.33. Jaringan Semantik

Keterangan :

T01 : Antarmuka Menu Utama

T02 : Antarmuka Menu *Credits*

T03 : Antarmuka *Play* Menu

T04 : Antarmuka Menu Bantuan

T05A : Antarmuka Arena Permainan

T05B : Antarmuka Menu *Pause*

T05C : Antarmuka Menu *Game Over*

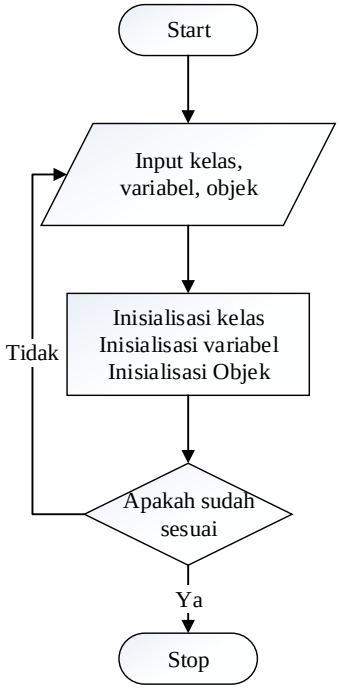
T05D : Antarmuka Menu Level Selesai

3.2.5 Perancangan Method

Perancangan komponen *method* merupakan perancangan yang dibuat setelah perancangan arsitektur menu, antarmuka dan jaringan semantik. Perancangan ini berfungsi untuk mendeskripsikan *method-method* yang berada di dalam aplikasi permainan. *Method* dapat dipanggil dengan menyertakan variabel, baik hanya satu variabel, banyak variabel atau bahkan tidak ada sama sekali. Adapun *method-method* yang terdapat dalam *game* Gatotkaca *Evolution* yang akan dibangun dijelaskan melalui *flowchart* sebagai berikut.

3.2.5.1 Method Init()

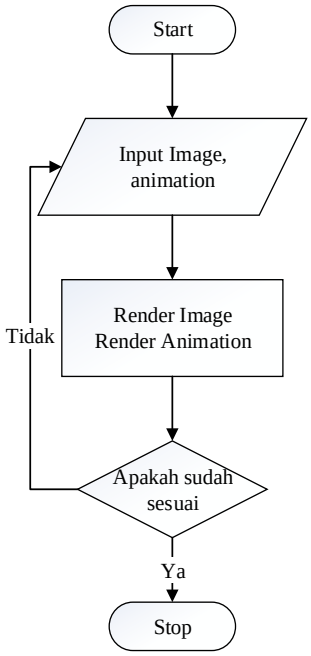
Method Init digunakan untuk melakukan inisialisasi terhadap setiap variabel, kelas dan objek-objek seperti gambar, animasi, musik dan lainnya. *Method* bisa dilihat pada gambar berikut.



Gambar 3.34. Flowchart Method Init()

3.2.5.2 Method Render()

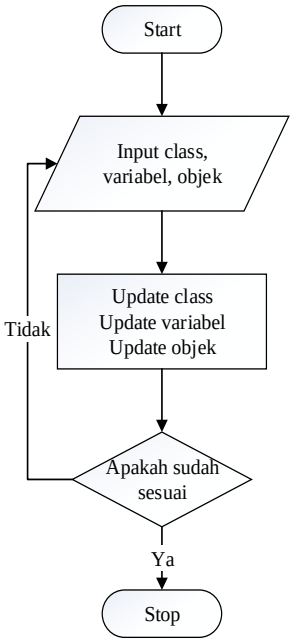
Method Render digunakan untuk melakukan penggambaran terhadap setiap objek gambar atau animasi yang akan digunakan. *Method* bisa dilihat pada gambar berikut.



Gambar 3.35. Flowchart Method Render()

3.2.5.3 Method Update()

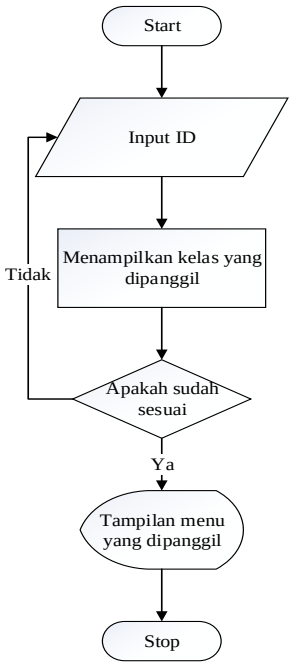
Method Update digunakan untuk melakukan pengupdatean terhadap setiap kelas, variabel dan objek yang akan digunakan. *Method* bisa dilihat pada gambar berikut.



Gambar 3.36. Flowchart Method Update()

3.2.5.4 Method getID()

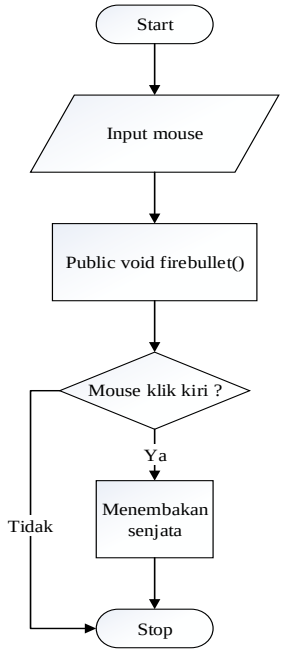
Method getID digunakan untuk melakukan pemanggilan terhadap setiap kelas yang telah atau akan dipanggil dalam bentuk menu. *Method* bisa dilihat pada gambar berikut.



Gambar 3.37. Flowchart Method getID()

3.2.5.5 Method fireBullet()

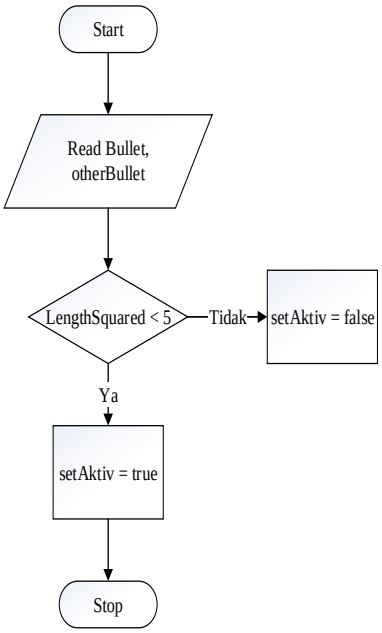
Method fireBullet digunakan agar karakter Gatotkaca dan musuh dapat menembakan senjata. Pada karakter Gatotkaca, *method* ini bekerja saat pemain menekan mouse bagian kiri. *Method* bisa dilihat pada gambar berikut.



Gambar 3.38. Flowchart Method fireBullet()

3.2.5.6 Method checkBulletCollision()

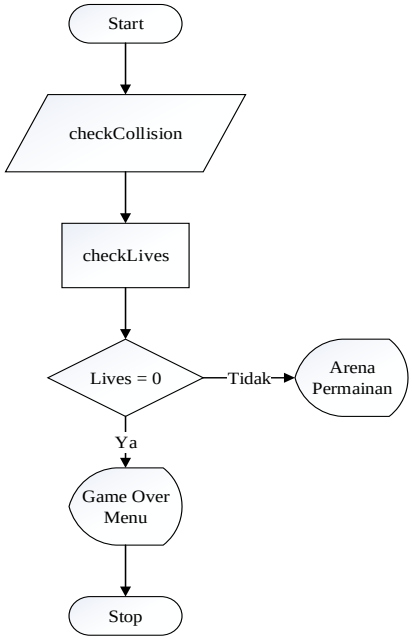
Method checkBulletCollision digunakan untuk melakukan pengecekan terhadap senjata musuh yang mengenai pemain maupun senjata pemain yang mengenai musuh. *Method* bisa dilihat pada gambar berikut.



Gambar 3.39. Flowchart Method checkBulletCollision()

3.2.5.7 Method die()

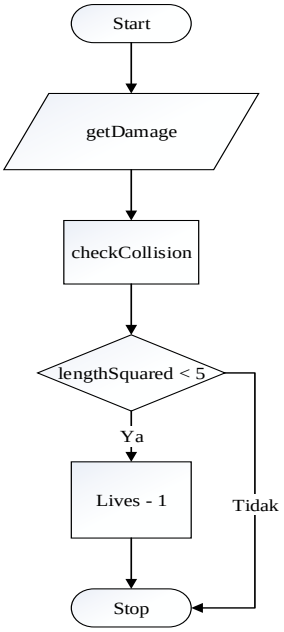
Method die digunakan untuk melakukan pengecekan terhadap darah pemain apakah habis atau belum. Pengecekan dilakukan saat senjata musuh mengenai pemain dan musuh menabrak pemain. *Method* bisa dilihat pada gambar berikut.



Gambar 3.40. Flowchart Method die()

3.2.5.8 Method getDamage()

Method getDamage digunakan untuk melakukan pengecekan terhadap tubrukan antara senjata musuh dengan pemain, senjata pemain dengan musuh dan tubrukan musuh dengan pemain. *Method* bisa dilihat pada gambar berikut.



Gambar 3.41. Flowchart Method getDamage()