**BAB III**

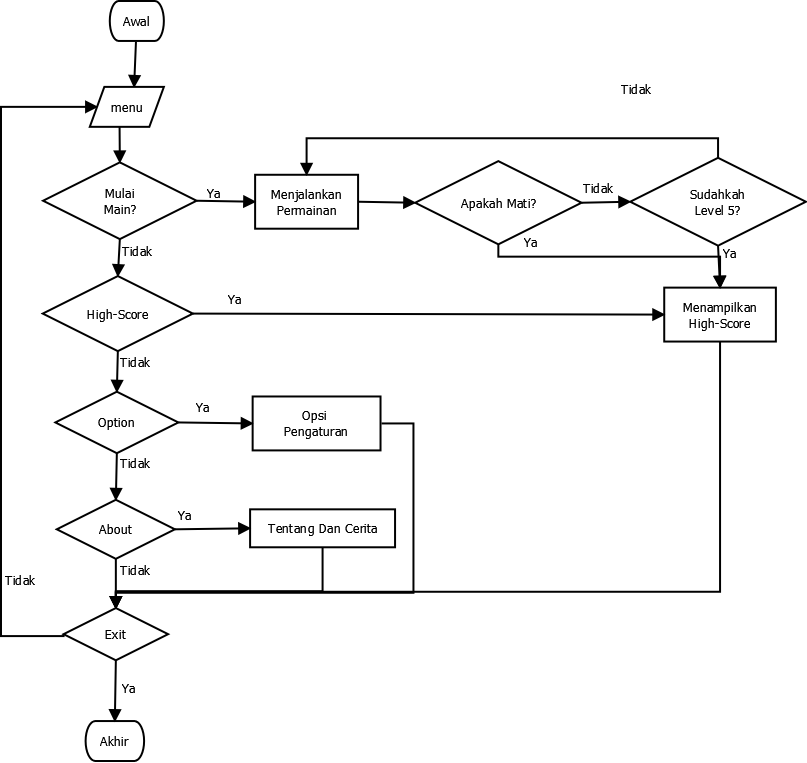
**ANALISA DAN PERANCANGAN SISTEM**

* 1. **Perancangan Aplikasi**

Perancangan aplikasi adalah suatu perancangan yang mendukung dan mengarahkan alur program yang dibuat. Pada permainan aritmatika ini, penulis menggunakan beberapa bagan untuk menjelaskan dan mengarahkan kemana permainan tersebut berjalan dan apa saja kelebihan dari permainan ini.

* + 1. **Perancangan flowchart**

Bagan alir sistem atau flowchart menggambarkan bagaimana urutan secara logika analisa sistem memecahkan suatu masalah tertentu. Berikut ini adalah flowchart sistem dari permainan aritmatika berbasis *arcade*.

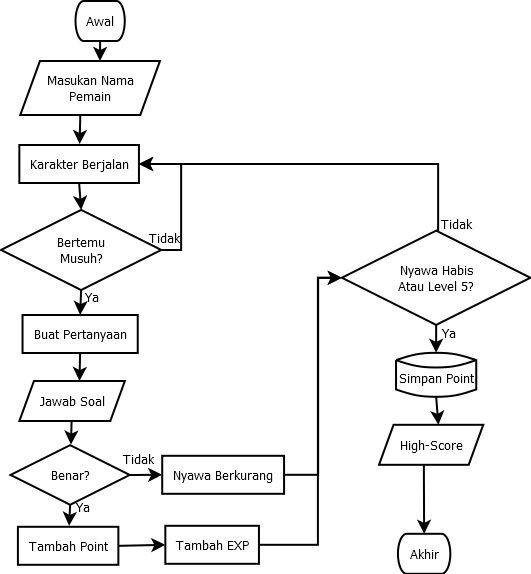


Gambar 3.1. Flowchart Aplikasi

Pada gambar 3.1 menjelaskan bahwa permainan akan selalu dijalankan jika *player* sudah mati atau *player* mencapai level 5.

1. ***Flowchart* Permainan**

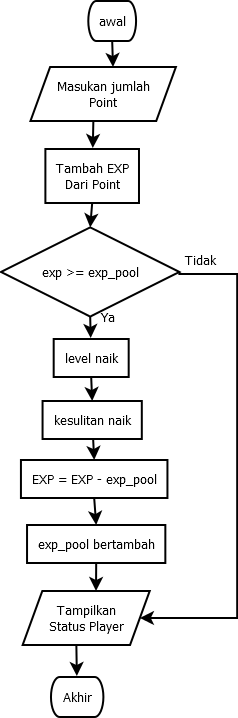
Pada *flowchart* akan menceritakan jalannya permainan saja. Yang mana permainan ini ber-genre *arcade*. Yang bercerita tentang seorang shogun (player) yang harus berperang dengan pasukan barbar (enemy). Disetiap soal yang tidak dijawab atau salah, maka anda mendapat kerusakan (damage) yang mengurangi nyawa secara perlahan.



Gambar 3.2 *Flowchart* Permainan

Pada gambar 3.2 menjelaskan bahwa permainan ini berjalan seperti *flowchart* diatas. *Flowchart* yang ini memberitahukan bahwa permainan akan terus berjalan hingga shogun kehilangan seluruh nyawanya. Pada *flowchart* ini menyebutkan bahwa batasan level adalah 5. Dan gambar 3.2 hanya menggambarkan jalannya permainan saja. Seperti jika bertemu musuh, maka pemain akan disuruh menjawab soal.

1. ***Flowchart* Naik Level Dan Pengalaman**

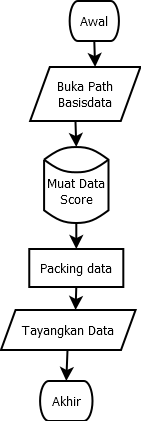


Gambar 3.3 *Flowchart* Naik Level Dan Pengalaman

Pada gambar 3.3 dijelaskan bahwa pemain akan naik level jika pengalaman yang didapat melebihi batas\_exp dan batas\_exp akan bertambah sesuai bertambahnya level.

1. ***Flowchart High-Score***

*Flowchart high-score* adalah *flowchart* yang menunjukkan permainan terbaik yang pernah dilakukan oleh pengguna.

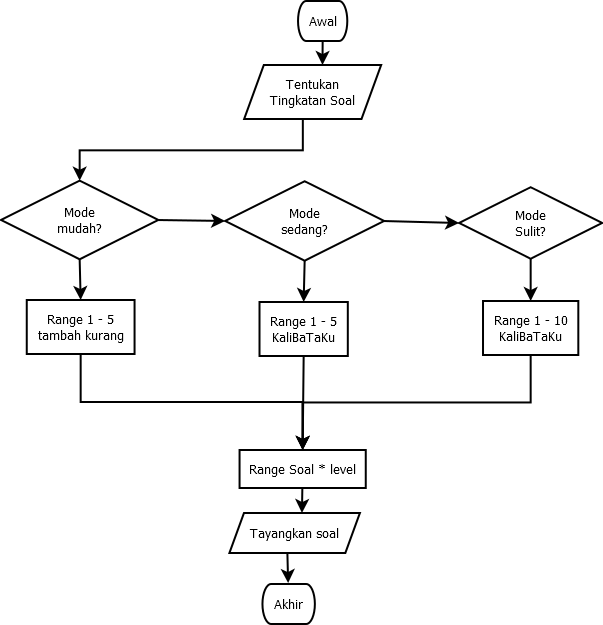


Gambar 3.4 *Flowchart High-Score*

Dari gambar 3.4 digambarkan bahwa setiap kali pengguna meminta daftar *high-score*. Maka sistem akan mengurutkan data tersebut dari permainan terbaik hingga terburuk.

1. ***Flowchart* Kesulitan**

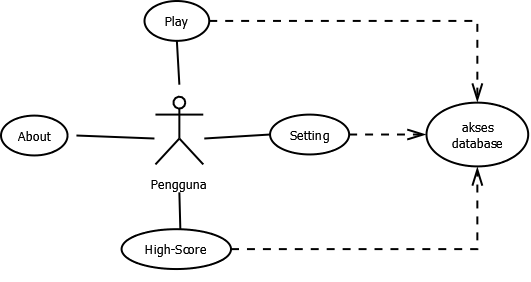
*Flowchart* kesulitan adalah *flowchart* yang menggambarkan tentang kesulitan tiap pengaturan, level dan kerusakan.



Gambar 3.5 *Flowchart* Tingkat Kesulitan

Di gambar 3.5 dijelaskan bahwa mode mudah tidak akan menemui operasi kali dan bagi. Sedangkan mode sedang dan sulit menemui kendala yang hampir sama di setiap pertarungannya. Dan setiap kali level karakter naik, maka range soal akan dikali 2. Jadi, range soal akan semakin panjang.

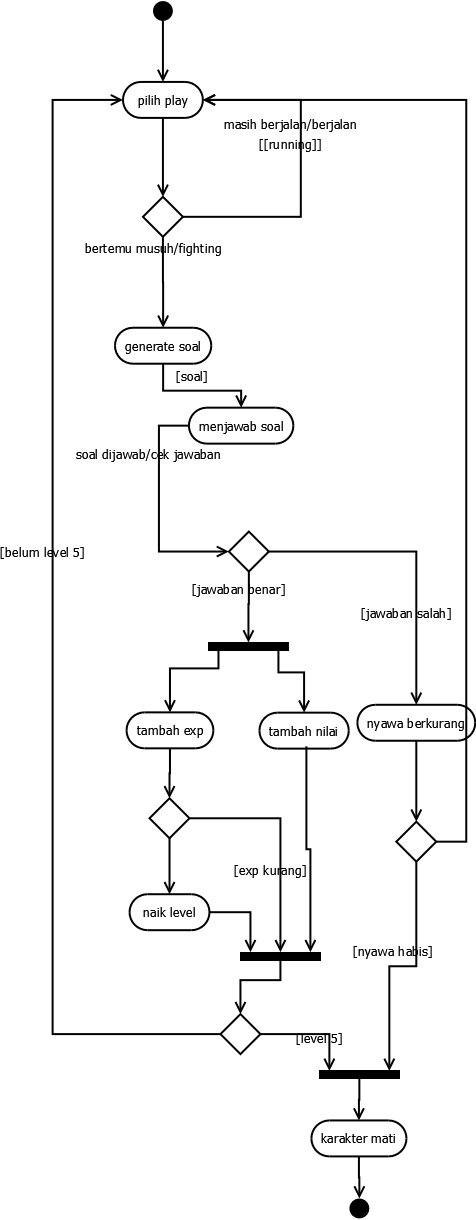
1. ***Use Case Diagram***



Gambar 3.5 Use Case

Pada gambar 3.5, kasus penggunaan dari permainan ini hanya terdiri dari 5 kasus. Akan tetapi, pengguna hanya melakukan 4 kasus saja yaitu kasus about, kasus play, kasus high-score dan kasus setting. Kasus akses database bersifat tertutup.

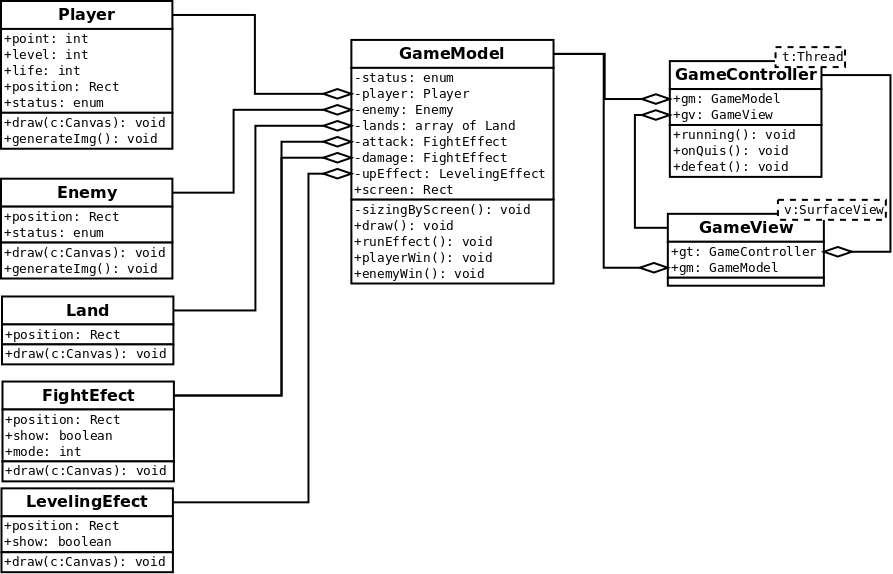
1. **Diagram Aktifitas Permainan**



Gambar 3.6 Activity Diagram Permainan

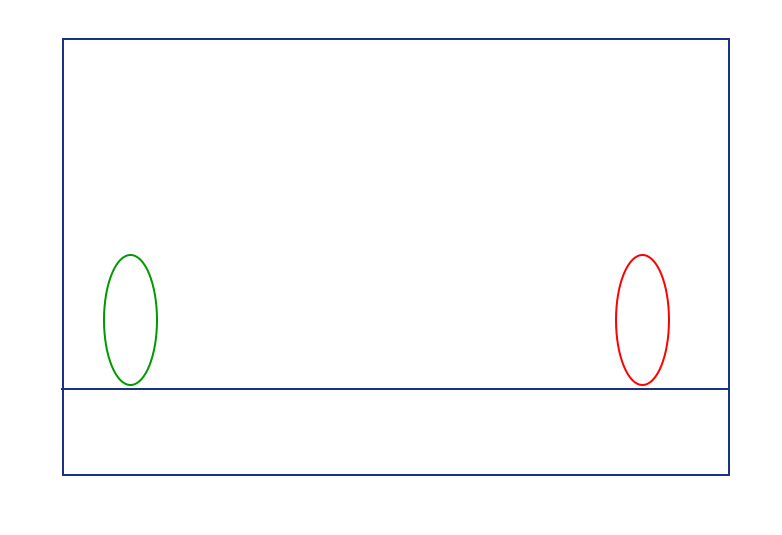
Pada gambar 3.6, diagram aktifitas permainan digambarkan bahwa diagram ini hanya menjelaskan tentang alur aktifitas permainan. Permainan dimulai dengan karakter yang berlari, kemudian bertemu musuh. Akan tetapi, nyawa karakter tidak langsung dikurangi. Melainkan pengguna harus menjawab pertanyaan dulu. Jika jawaban salah, maka nyawa karakter dikurangi. Hingga nyawa karakter habis atau karakter mencapai level 5.

1. **Class Diagram**

Gambar 3.7 Class Diagram

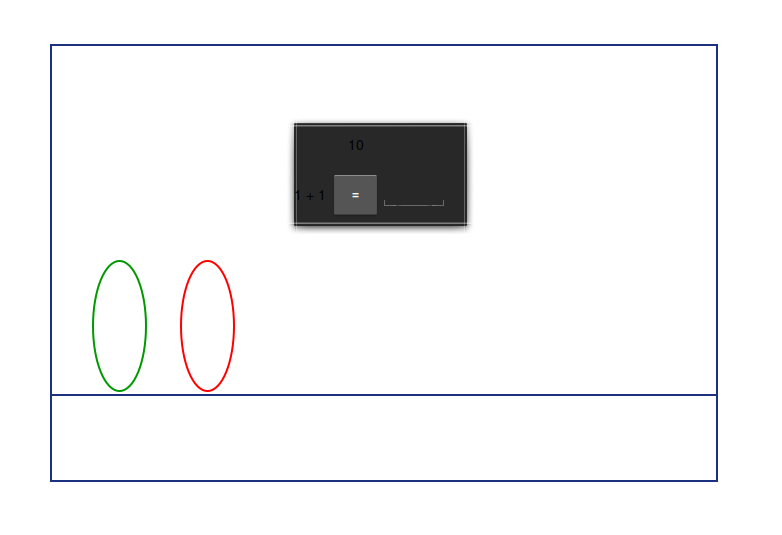
Pada gambar 3.7, digambarkan bahwa *class* GameModel mengatur dan menyimpan semua kegiatan permainan serta menayangkannya. Class yang terdapat pada gambar 3.7 terdiri dari: Player, Enemy, Land, FightEffect, LavelingEffect, GameModel, GameController, dan GameView. Player adalah tokoh utama dari permainan ini. Enemy adalah class yang bertugas untuk memberikan soal pada player. Land adalah class yang menggambarkan struktur tanah pada permainan. FightEffect adalah class yang akan muncul dan memberikan kerusakan pada player atau enemy. LevelingEffect adalah class yang menunjukkan kenaikan level darai player.

**3.1.8 Rancangan Antar Muka Game Ketika Berjalan**

Gambar 3.8 Rancangan Antar Muka Ketika Berjalan

Pada gambar 3.8, diterangkan bahwa oval merah (Enemy) akan bergerak mendekati oval hijau (Player). Hingga mencapai titik temu kemudian terjadilah pertarungan.

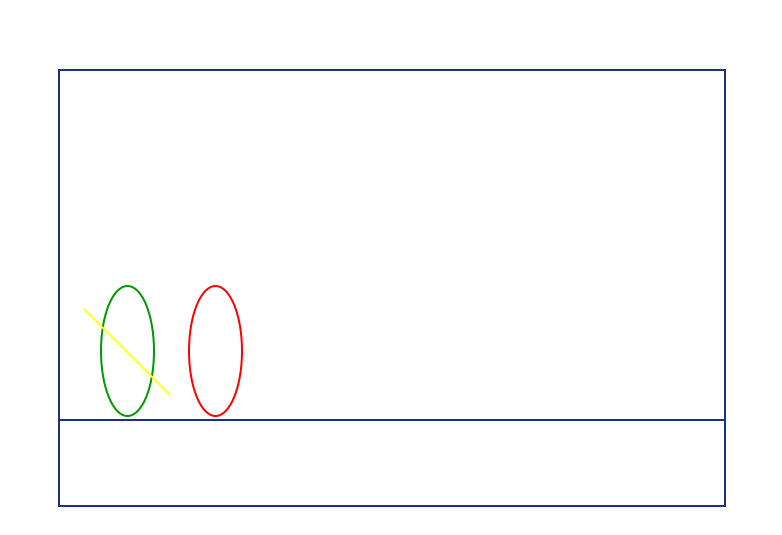
**3.1.9 Rancangan Antar Muka Game Ketika Bertarung**



Gambar 3.9 Rancangan Antar Muka Ketika Bertarung

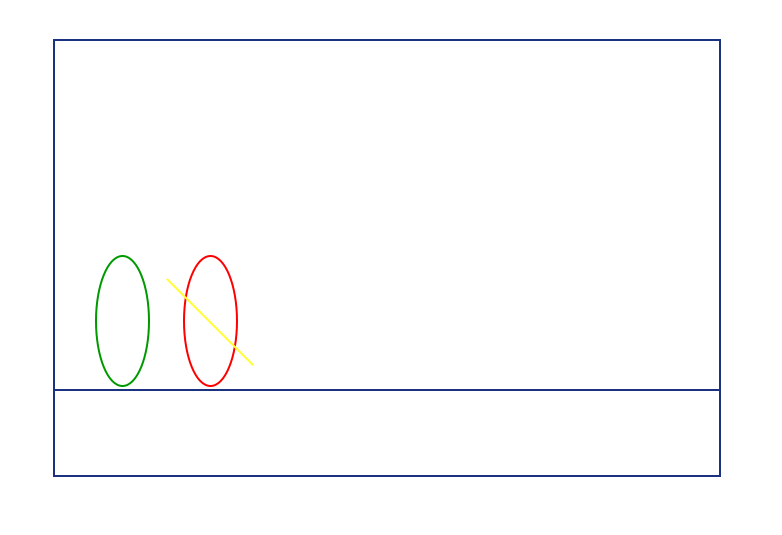
Gambar 3.9 menunjukan aksi ketika enemy dan player bertemu. Dan player harus menjawab dengan benar soal diatas. Jika jawaban salah, maka enemy akan menanyakan soal selanjutnya. Jika jawaban benar, maka enemy tidak akan menayangan soal lagi.

**3.1.10 Rancangan Antar Muka Game Ketika Terluka**

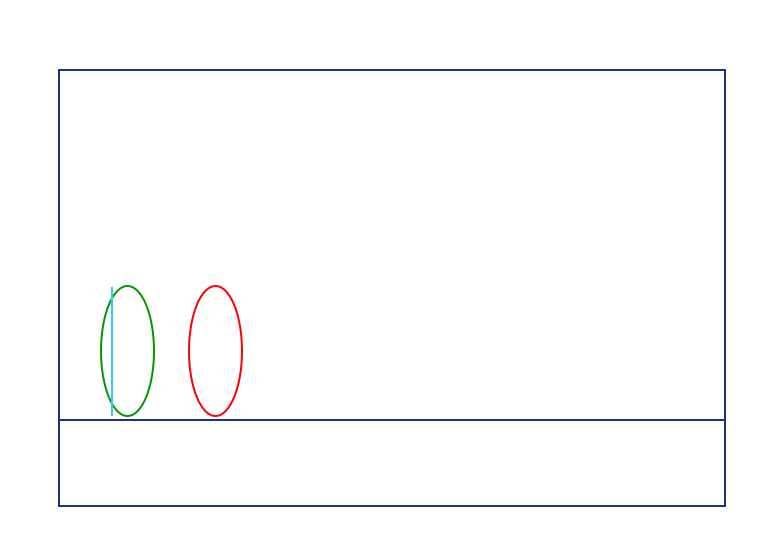
Gambar 3.10 Rancangan Antar Muka Ketika Terluka

Gambar 3.10 menerangkan bahwa ketika jawaban salah maka player menerima kerusakan (damage) yang digambarakan dengan garis kuning. Kemudian player berjalan sedikit dan akan muncul soal baru seperti pada gambar 3.9.

**3.1.11 Rancangan Antar Muka Game Ketika Menyerang**

Gambar 3.11 Rancangan Antar Muka Ketika Menyerang

Gambar 3.11 menerangkan bahwa player yang menjawab soal dengan benar. Akan melakukan penyerangan kemudian enemy tersebut tidak bisa menayangkan soal kembali. Jika player berhasil naik level, maka rancangan antar muka seperti gambar 3.12 akan ditayangkan.

**3.1.12 Rancangan Antar Muka Game Ketika Naik Level**

Gambar 3.12 Rancangan Antar Muka Ketika Naik Level

Gambar 3.12 menerangkan bahwa akan ada efek tertentu ketika player naik level.