# BAB III

# ANALISA DAN PERANCANGAN

# Analisa Masalah

Pesatnya kemajuan teknologi di zaman yang serba canggih ini, juga berpengaruh terhadap perkembangan *game*. Banyak sekali *game* yang beredar di kalangan masyarakat dan jenisnya pun bermacam-macam, ada yang berasal dari dalam negeri dan ada pula yang berasal dari luar negeri. Saat ini *game* yang terdapat di Indonesia sebagian besar berasal dari luar negeri sehingga memiliki peminat yang cukup banyak dari pada *game* yang berasal dari dalam negeri.

Ada banyak *Game*s yang ada sekarang dan ada banyak *game* yang merupakan *game board* yang dibuat versi digital-nya seperti catur, ular tangga, dan tik-tok Dan *game*-*game* yang diadaptasi bisa masih memiliki *gameplay* yang sama seperti *game board*-nya atau menggunakan *gameplay* lain. Salah satu *game* board yang di adaptasi ke digital adalah *game* *Battleships*, yang dimana *game* *Battleships* adalah *game* yang memiliki *gameplay* *Turn-Base Strategy.* Salah satu *Device* yang menjadi tempat adaptasi *game* *Battleships* adalah *Android*.

Setelah dilakukanya observasi *game* *Battleships offline* untuk *Android* yang ada pada Play Store didapatkan data dari 33 *game* *Battleships* yaitu sebagai berikut.

Table 3.1 hasil pengamatan game battleships di play store

|  |  |  |
| --- | --- | --- |
| No. | Gamplay | Jumlah |
| 1 | Action | 12 *Game* |
| 2 | Turn-Base Strategy | 18 *Game* |
| 3 | Shooter | 3 *Game* |
|  | Total *Game* | 33 *Game* |

## Identifikasi Masalah

Berdasarkan analisa masalah diatas, terdapat permasalahan yang dihadapi yaitu:

1. Masalah
   1. Adaptasi *game* masih banyak yang menggunakan *gameplay Turn-base Strategy.*
   2. Beberapa *gameplay* yang ada belum memiliki adaptasi *game* *Battleships* contohnya *Real-Time Strategy* dan *simulator.*
2. Penyebab
   1. *Game* *Battleships* yang ada masih banyak yang mengikuti dasar dari *game* *board-*nya.
   2. Kurangnya inovasi *game* *Battleships* dengan *gameplay* yang berbeda.
3. Dampak
4. Pemain hanya dapat bermain *game* *Battleships* dengan *gameplay* yang sama.
5. Banyak *game* *Battleships* namun perbedaannya hanya pada grafisnya karena *gameplay*-nya sama.

## Usulan Pemecahan Masalah

Berdasarkan hasil identifikasi masalah, maka didapatkan usulan yaitu mengadaptasi game *battleships* dengan menggunakan *gameplay* yang belum ditemukan pada *google play* yang dimana adalah *real-time strategy*

## Kelebihan Aplikasi

Dengan mengadaptasi *game battleship* dan mengubah *gameplay*-nya menjadi *real-time strategy game battleship* bukan hanya memberikan pemain game *battleship* dengan nuansa baru tetapi juga memberikan tantangan bertarung *secara real-time* sehingga pemain harus berusaha menyerang dan menjaga unit agar tidak hancur tidak seperti *turn-base battleships* yang dimana pemain hanya dapat menyerang secara bergantian.

# Perencanaan Game

pada tahap perencanaan game penulis merencanakan user story, objek 3D yang diperlukan, dan aplikasi apa saja yang dibutukan dalam pembuatan.

## Use story

perencanaan story adalah dimana pemain adalah komandan battalion dan diberikan misi untuk menghancurkan 3 kapal utama musuh(Boss). pemain akan diberikan 3 kapal yang berbeda dalam misi.

## Objek 3D

objek 3d yang akan di butuhkan dalam pembuatan adalah sebagai berikut.

Tabel 3.1 3D objek yang dibutuhkan

|  |  |  |
| --- | --- | --- |
|  | Kebutuhan | Objek |
| 1 | Arena | 1. Terrain  2. Water |
| 2 | Unit Pemain | 1. Unit 1 (Crusier)  2. Unit 2 (Battleships)  3. Unit3 (Destroyer) |
| 3 | Unit Musuh | 1. Kapal Musuh  2. Kapal Boss |
| 4 | Heal | 1. Tower  2. Tanda Heal |

## Aplikasi yang dibutuhkan

aplikasi yang akan dibutukan dalam pembuatan game ini ada 2 yaitu Unity dan Visual Studio. Aplikasi Unity akan digunakan dalam pembuatan objek dan pembuatan game sedangkan Visual Studio akan berungsi dalam pembuatan scrips atau Coding dalam pembuatan game.

# Perancangan *Game*

Pada tahap perancangan *game*, dipaparkan proses perancangan sistem sesuai dengan metode atau teknik yang digunakan berdasarkan hasil analisa pada tahapan sebelumnya**.**

## Gambaran Umum *Game*

*Game* ini menggunakan konsep *game* *Real-Time Strategy* dimana pemain akan diberikan 3 unit yang setiap unit memiliki *Health* dan *Damage* yang berbeda dan menggunakannya untuk mencari boss dan unit musuh lalu menghancurkannya.

## Rancangan *Game*

Rancangan game berisi Tujuan utama yang adalah tujuan pemain dalam game, rancangan *gameplay* yang dimana adalah rancangan cara bermain, stage yang menjadi tempat bermain, unit yang bisa dikendalikan, sistem game dan *win/lose condition*.

1. Tujuan Utama

Tujuan utama dari *game* adalah untuk mencari dan menghancurkan boss unit yang ada pada stage agar dapat menyelesaikan stage pada *game*. dalam perjalanan atau explorasi pemain dapat bertemu unit musuh yang ada di sekitar *environment* dan akan diserang jika posisi unit pemain dekat dengan unit musuh.

1. Rancangan *Gameplay*

Saat memulai *game* pemain akan diberikan 3 unit yang dimana tiap unit memiliki kelebihan dan kekurangan masing-masing untuk melakukan explorasi dan menyerang unit musuh. pemain akan memilih diatara 3 unit yang ada dan mengendalikan unit yang dipilih dan jika pemain ingin mengunakan unit yang lain pemain perlu memiliih unit tersebut dan mengendalikannya. Saat Pemain melakukan explorasi pemain akan bertemu unit musuh namun pemain dapat memilih untuk menyerang unit musuh atau tidak di karenakan unit musuh tidak akan menyerang selama unit pemain tidak ada dalam jarak deteksi unit musuh namun jika unit musuh tersebut adalah boss maka pemain harus melawan dan menghancurkan unit tersebut agar bisa melanjutkan atau menyelesaikan atau memenangkan stage.

1. *Stage*

Pada *game* ini akan terdapat 2 (dua) stage. Stage 1 adalah Dimana pemain akan memulai *game* dan Agar pemain dapat melanjutkan *game* ke Stage 2 pemain diharuskan memenangkan stage 1.

1. *Stage* 1
   1. Misi utama menemukan dan menghancurkan 3 unit boss.
   2. Selama perjalanan menemukan boss ada unit musuh yang akan menyerang jika unit pemain dekat atau masuk jarak deteksi unit.
2. *Stage* 2
3. Misi utama menghancurkan 1 boss.
4. Unit musuh memiliki *Damage* dan *Health* lebih besar.
5. Root/Status

Root/Status adalah inti dan indikator yang ada pada tiap unit dan berisi informasi *damage*, *health,* *attack speed,* unit yang terdeteksi dan status *movement* unit.

1. *Damage*

*Damage* adalah kerusakan yang dapat di berikan oleh unit dan terbagi menjadi *Damage* terkecil, *Damage* normal dan *Damage* terbesar.

1. *Health*

*Health* Atau HP adalah daya tahan dari unit yang jika habis maka unit akan hancur dan terbagi menjadi *Health* Paling Sedikit, *Health* normal dan *Health* Terbanyak.

1. *Attack Speed*

*Attack Speed* adalah kecepatan unit untuk melakukan damage pada unit lain. *Attack Speeed* tiap unit berbeda ada yang cepat, normal dan lambat.

1. Unit yang Terdeteksi

Unit yang Terdeteksi berfungsi untuk mengidentifikasi unit yang ada pada jarak deteksi sehingga sistem dapat memberikan keputusan untuk menyerang unit tersebut atau tidak tergatung apakah unit tersebut musuh atau bukan.

1. status *movement* unit

status *movement* unit adalah indikator yang ada pada sistem yang berguna untuk mengetahui apakah unit sedang berjalan(move), diam(idle), mengikuti unit lain (follow) atau menyerang (combat).

1. *Movement* Sistem

*movement* sistem berfungsi mengatur fungsi bergerak pada unit pemain dan enemy unit

1. *Follow* Sistem

*Follow* sistem berfungsi agar unit pemain dapat menggikuti unit lain nya seingga pemain hanya perlu menggerakan 1 unit dan unit lain akan menggikuti unit tersebut.

1. Unit

Setiap unit memiliki perbedaan dan ada 3 jenis perbedaan yang terdapat pada Root dan perbedaannya yaitu pada *Damage*, *Health* dan *Attack Speed*:

1. Unit 1

Unit pertama yang dapat digunakan pemain.

1. Memiliki *Health* terkecil
2. Memiliki *Damage* terkecil
3. Memiliki *Attack Speeed* yang cepat
4. Unit 2

Unit kedua yang dapat digunakan pemain.

1. Memiliki *Health* yang normal
2. Memiliki *Damage* yang normal
3. Memiliki *Attack Speeed* yang Normal
4. Unit 3

Unit ketiga yang dapat digunakan pemain.

1. Memiliki *Health* terbesar dari antara unit pemain
2. Memiliki *Damage* terbesar dari antara unit pemain
3. Memiliki *Attack Speeed* yang lambat
4. Enemy Unit 1

unit musuh yang ada ada level 1.

1. Memiliki *Health* kecil
2. Memiliki *Damage* normal
3. Memiliki *Attack Speeed* yang lambat
4. Enemy Unit 2

Unit musuh yang ada pada level 2.

1. memiliki *Health* Normal
2. memiliki *Damage* normal
3. Memiliki *Attack Speeed* yang normal
4. Boss

Unit musuh yang harus di hancurkan untuk menyelesaikan Stage.

1. Memiliki *Health* Tebanyak dari semua unit
2. Memiliki *Damage* Terbesar dari semua unit
3. dapat mengeluarkan *enemy* unit 1.
4. *Pointer*

*Pointer* Adalah Indikator unit yang menandakan kemana unit tersebut berjalan atau tujuan dari unit yang bergerak dan indikator unit mana yang di ikuti jika pemain memberikan perintah untuk mengikuti unit lain. Dimana *Pointer* unit akan diletakan ditentukan oleh pemain dan unit yang tehubung dengan *Pointer* tersebut akan bergerak menuju *Pointer* tersebut. setiap unit milik pemain memiliki *Pointer* yang berbeda warna sehingga pemain bisa mengetauhi unit mana yang sedang bergerak. *Pointer* contohnya unit 1 berwarna merah dan unit 2 hijau dan saat pemain sedang mengerakan unit 1 *Pointer* yang muncul berwarna merah dan saat pemain menggerakn unit 2 *Pointer* yang muncul berwarna hijau.

1. Detector

Detector Berfungsi mendeteksi unit yang ada di sekitarnya. detector memiliki jarak dan jika ada unit yang terdeteksi makan unit tersebut akan dimasukan pada unit yang terdeteksi di *root.*

1. *Winning/Lose Condition*

Dalam *game* ada condisi yang menandakan jika pemain menang atau kalah yaitu sebagai berikut.

1. *Winning Condition*

*Winning condition* adalah kondisi yang di dapatkan pemain saat pemain berhasil menghancurkan boss pada stage

1. *Lose Condition*

*Lose condition* adalah kondisi yang terjadi jika semua unit milik pemain hancur.

1. *Enemy AI*

unit musuh akan mempunyai AI yang akan digunakan dalam *game*. AI tersebut dibagi menjadi 3 yaitu:

1. *Enemy Alert(Object Detection)*

*Enemy* Alert adalah AI milik *enemy* unit yang berfungsi jika ada salah satu *enemy* unit yang sedang diserang oleh unit pemain ada dalam detector milik unit tersebut maka unit tersebut akan membantu dan ikut dalam *combat* dengan unit pemain.

1. *Enemy Movement* dan *waypoint (Pathfinding)*

*Enemy* *Movemen*t memiliki beberapa perbedaan dengan unit movement biasa di karenakan *enemy* unit mengikuti waypoint yang ada dan jika dalam perjalanan *enemy* unit mendeteksi ada unit pemain maka *enemy* unit akan melakukan combat karena combat state pada *enemy* unit lebih di prioritaskan dari pada moving state. *Enemy* *Movemen*t menggunakan logika *pathfinding* agar saat berjalan unit tidak bertabrakan jika ada objek di depan unit

1. *Enemy Respawn*

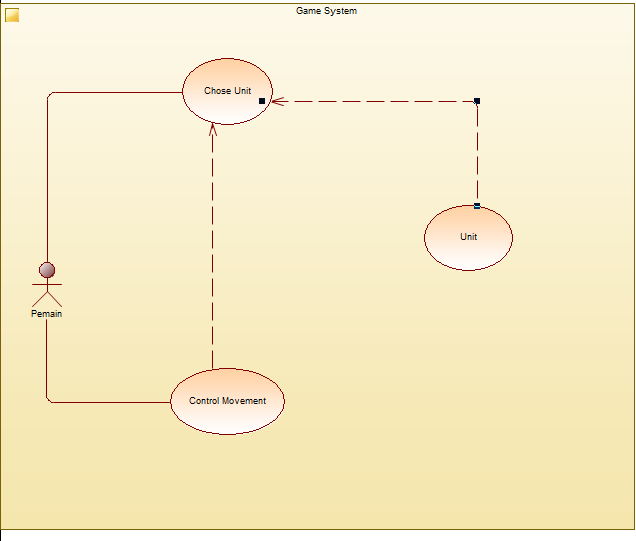
*Enemy* *respawn* adalah AI untuk boss yang dimana boss akan mengeluarkan *enemy* unit dengan waktu tertentu.

## Desain Perancangan UML *Game*

Sebelum ke tahap *coding* dilakukan perancangan atau pemodelan dalam bentuk diagram UML sebagai berikut.

1. **Use Case Diagram**
   1. **Use Case Pemain**

Di dalam *game* ini pemain bisa memilih dan mengendalikan movement *movement* unit.



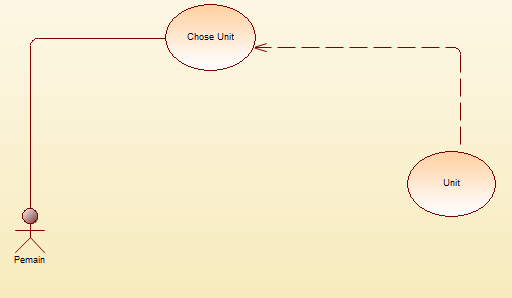
**gambar 3.1** use case pemain

pada control movement pemain dapat mengendalikan pergerakan unit dan jika unit ada pada jarak deteksi unit musuh maka sistem akan mengaktifkan combat sistem.

1. *Use Case Scenario* Pemain

Dalam *use case scenario* penulis memberikan penjelasan mengenai runtutan langkah yang harus dipenuhi dalam sebuah alur kejadian, *flow of events.* Serta ada kondisi-kondisi yang harus dipenuhi untuk melakukan suatu *flow of events*. Kondisi yang maksud yaitu : *precondition*, kondisi sebelum berjalannya aktivitas; *postcondition,* kondisi setelah berjalannya *events*.

1. Use Case Chosen Unit



**gambar 3.2** use case chosen Unit

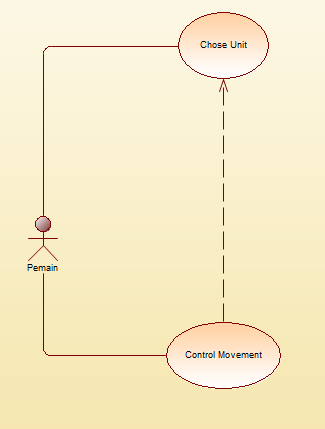
pemain dapat memilih unit yang akan dikendalikan dan setelah pemain memilih maka sistem akan menunjukan unit yang telah dipilih dan sedang di kendalikan.

Tabel 3.2 tabel use case chosen Unit

|  |  |
| --- | --- |
| Precondition | player belum memilih unit  sistem memilih unit 1 secara default |
| Flow of events | pemain memilih unit  sistem memilih unit |
| Postcondition | if unit chosen> camera follow, |

1. Use Case Control Movement

pemain memberikan posisi baru pada unit yang dipilih lalu sistem akan menampilkan sprite pada posisi yang dipilih pemain dan unit yang dipilih akan menuju ke posisi sprite atau posisi yang di pilih pemain.



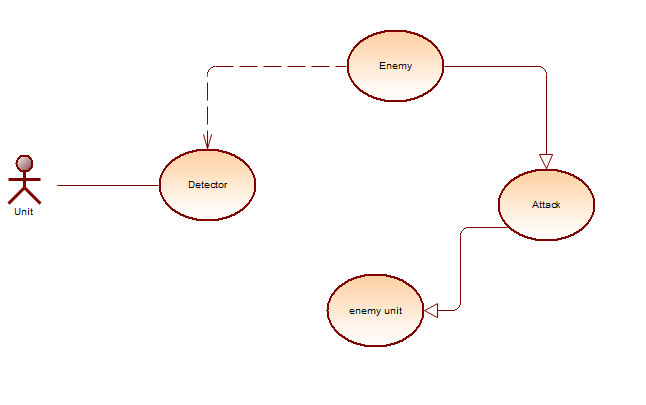
**gambar 3.3** use case control movement

Tabel 3.3 table use case control movement

|  |  |
| --- | --- |
| *Precondition* | unit sudah di pilih |
| *Flow of events* | menampilkan sprite pada posisi yang dipilih  pemain memberikan posisi baru untuk unit |
| *Postcondition* | unit chose > move = new position |

* 1. **Use Case Combat**

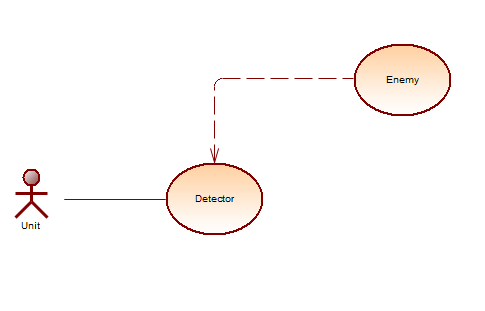
Sistem pada combat yang mengatur deteksi unit dan penyerangan unit.



**gambar 3.4** use case combat

1. *Use Case Scenario* Combat
2. Use Case Detector

sistem yang berfungsi dalam melakukan identifikasi dalam jarak tertentu.

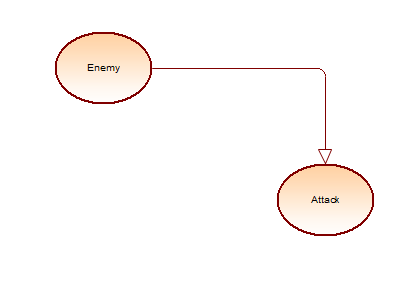
 **gambar 3.5** use case detector

Tabel 3.4 tabel use case detector

|  |  |
| --- | --- |
| *Precondition* | belum ada unit yang terdeteksi |
| *Flow of events* | ada unit yang terdeteksi |
| *Postcondition* | mengidentifikasi unit yang terdeteksi enemy atau bukan |

1. *Use Case detect Enemy*

sistem yang digunakan untuk membedakan hasil identifikasi unit yang ada pada jarak deteksi.



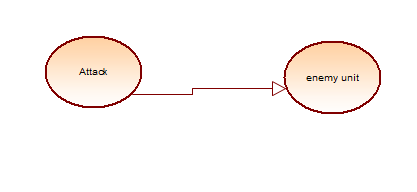
**gambar 3.6** use case enemy

tabel 3.5 tabel use case enemy

|  |  |
| --- | --- |
| *Precondition* | ada unit yang terdeteksi |
| *Flow of events* | unit yang terdeteksi adalah unit musuh(*enemy* unit)  memasukan hasil identifikasi ke sistem |
| *Postcondition* | unit mendekati musuh sampai pada jarak serang |

1. Use Case Attack

Attack adalah sistem yang berfungsi untuk memberikan perintah menyerang unit yang ada dalam jarak serang jika hasil dari identifikasi unit tersebut adalah unit musuh.



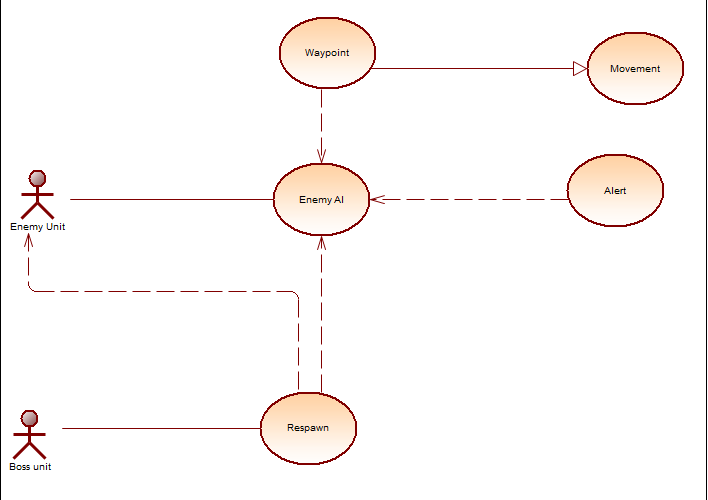
**gambar 3.7** use case attack

tabel 3.6 tabel use case attack

|  |  |
| --- | --- |
| *Precondition* | unit musuh terdeteksi |
| *Flow of events* | mendekati musuh sampai pada jarak serang |
| *Postcondition* | menyerang musuh/*enemy* unit |

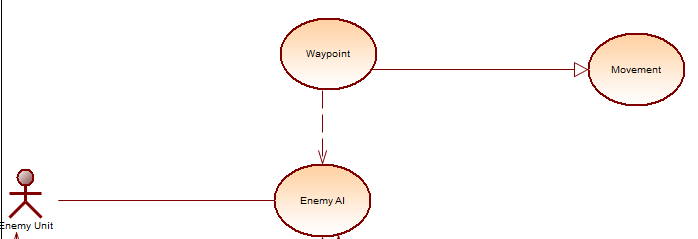
* 1. **Use Case Enemy AI**

use case sistem yang mengatur waypoint, respawn dan alert AI pada unit musuh.



**gambar 3.8 use case enemy unit AI**

1. *Use Case Scenario enemy unit AI*
2. *Use Case Scenario Waypoint*

****

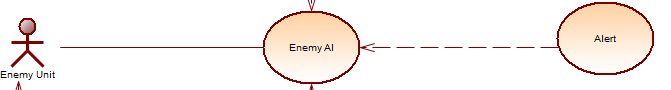
**gambar 3.9** use case waypoint

enemy unit akan memiliki waypoint sebagai tujuan dan saat enemy unit sampai pada waypoint unit akan berhennti sejenak dan melanjutkan ke waypoint berikutnya. unit akan terus bergerak ke waypoint sampai unit tersebut mencapai waypoint terakhir.

**tabel 3.7** tabel use case waypoint

|  |  |
| --- | --- |
| *Precondition* | unit memiliki waypoint tujuan |
| *Flow of events* | memberikan perintah bergerak ke waypoint  menentukan waypoint berikutnya |
| *Postcondition* | if it first waypoint decide next waypoint  if it last waypoint decide stop |

1. *Use Case Scenario Alert*

****

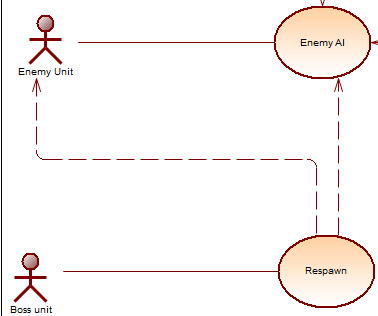
**gambar 3.10** use case alert

sistem alert pada unit musuh berfungsi jika unit tersebut sedang melakukan combat dengan unit pemain maka unit tersebut akan memberikan peringatan kepada unit musuh lainnya yang ada pada jarak deteksi unit tersebut untuk ikut dalam combat dengan unit pemain.

**Tabel 3.8** table use case alert

|  |  |
| --- | --- |
| *Precondition* | unit teman ada dalam jarak deteksi |
| *Flow of events* | mendeteksi ada unit teman di jarak deteksi  mengaktifkan sistem alert |
| *Postcondition* | if enemy on detector alert friend unit on detector  enemy unit position > alert friend |

1. *Use Case Scenario Respawn*

****

**gambar 3.11** use case respawn

sistem respawn akan digunakan oleh 1 unit musuh yaitu boss yang dimana boss dapat melakukan respawn pada unit musuh selain boss dalam waktu yang ditentukan.

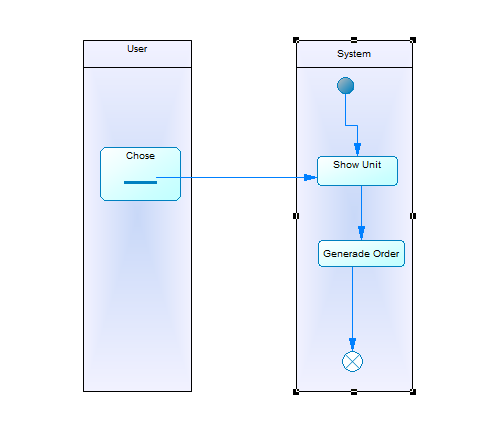
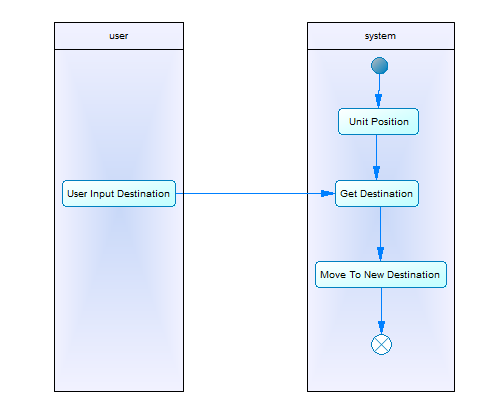
**tabel 3.9** tabel use case respawn

|  |  |
| --- | --- |
| *Precondition* | sudah di berikan waktu respawn and jarak antara respawn |
| *Flow of events* | respawn unit setelah waktu respawn yang ditentukan  respawn unit setelah jarak respawn terpenuhi |
| *Postcondition* | respawn time > respawn unit |

1. **Activity Diagram**

Berikut alur *Activity* pada *game* ini

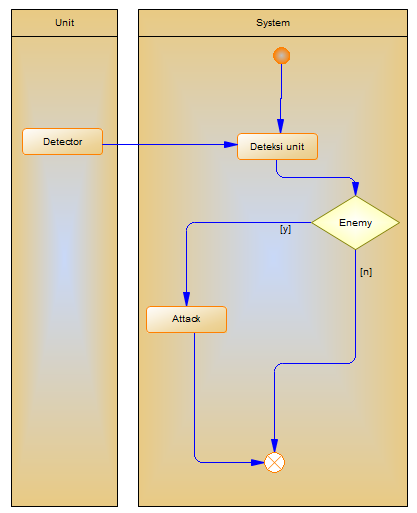
1. *Activity Chosen Unit dan Control Movement*

** **

**gambar 3.12** *Activity Chosen Unit dan Control Movement*

pemain memilih unit dan setelah dipilih sistem akan menampilkan unit tersebut dan pemain dapat mengendalikan unit yang telah dipilih tersebut dan menggerakan unit tersebut ke posisi baru.

1. *Activity Detector And Combat/Attack*

****

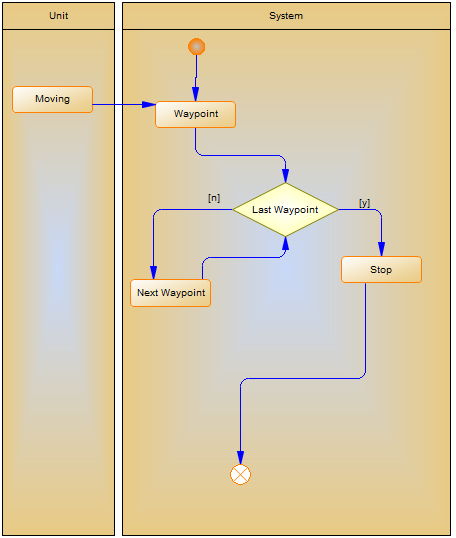
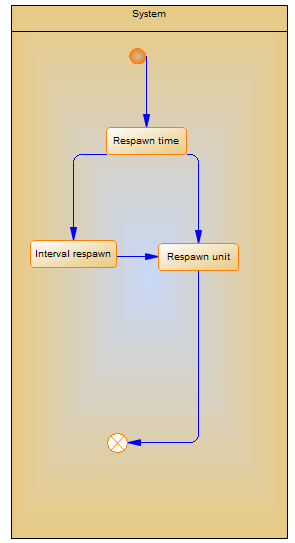
**gambar 3.13** *Activity Detector And Combat/Attack*

unit memiliki jarak deteksi dan saat ada unit yang masuk pada jarak deteksi maka sistem akan mengidentifikasi apakah unit tersebut musuh atau bukan dan jika musuh maka unit akan melakukan penyerangan pada unit yang di deteksi tersebut.

1. *Activity Enemy AI*

berikut *Activity* yang ada pada AI untuk unit musuh yang ada dalam game.

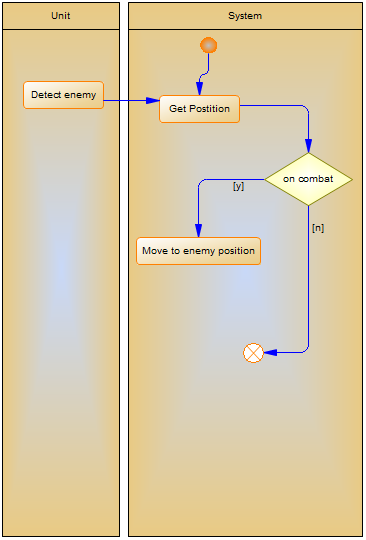
1. *Activity enemy Waypoint dan Respawn AI*

**** ****

**gambar 3.14** *Activity enemy Waypoint dan Respawn AI*

unit musuh akan melakukan respawn sesuai dengan respawn time dan setelah unit pertama respawn maka unit kedua akan respawn dengan jarak waktu yang ditentukan . jika unit tersebut memiliki tujuan waypoint maka unit tersebut akan lansung bergerak ke waypoint tujuan.

1. *Activity Alert AI*

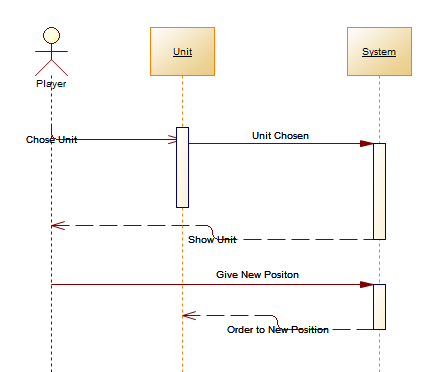
****

**gambar 3.15** *Activity Alert AI*

jika ada unit musuh yang mendeteksi unit pemain dan melakukan combat maka unit musuh tersebut akan memberikan posisi dari unit pemain dan memberikan alert pada unit musuh lain yang ada pada jarak deteksinya untuk datang dan melakukan combat dengan unit pemain.

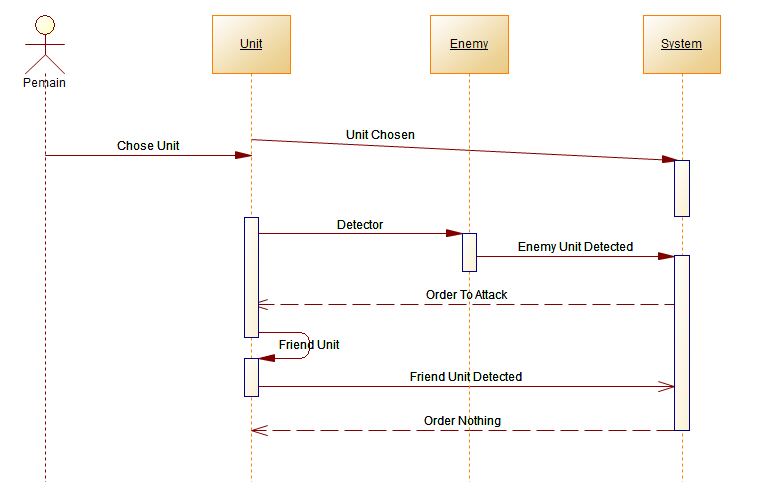
1. **Sequence Diagram**
2. Unit Sequence Diagram

Berikut penulis menjelaskan tentang interaksi tiap *object* dalam sebuah urutan waktu.



**gambar 3.16** Sequence Diagram Unit (1)

*Sequence diagram* pada gambar menunjukan interaksi antara unit yang dipilih dengan sistem saat pemain memilih unit pada proses chosen unit dan control movement.

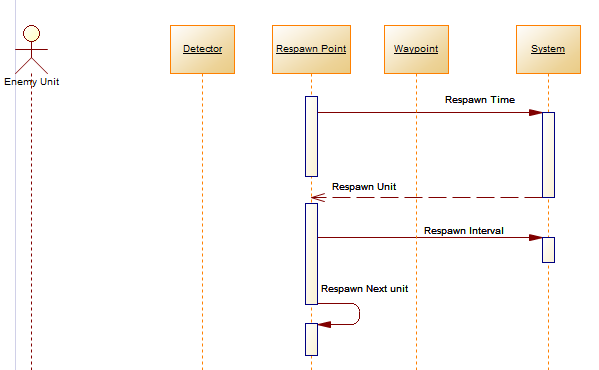
****

**gambar 3.17** Sequence Diagram Unit (2)

*Sequence diagram* pada gambar menunjukan interaksi antar objek dalam proses deteksi unit dan attack. pada *sequence diagram* dapat dilihat jika detector mendeteksi unit dan unit tersebut adalah musuh maka sistem akan langsung memerintahkan unit untuk menyerang unit tersebut namun jika unit yang terdeteksi adalah unit teman maka sistem tidak akan memerintahkan penyerangan pada unit tersebut.

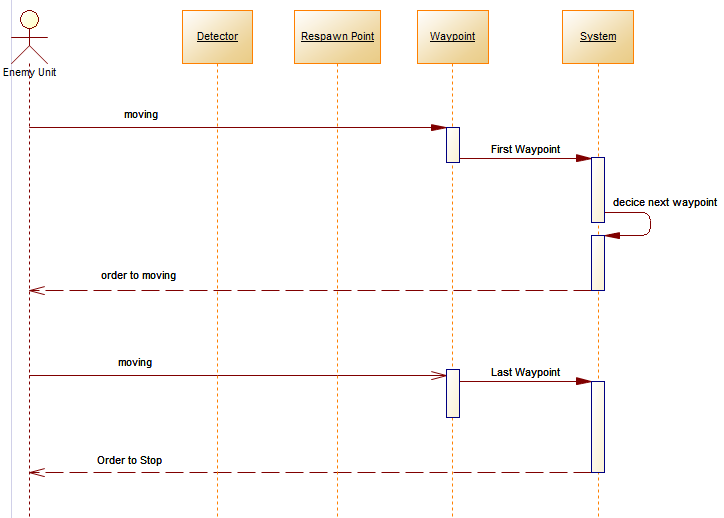
1. Enemy AI Sequence Diagram

Berikut penulis menjelaskan tentang interaksi AI dalam mengatur Unit .



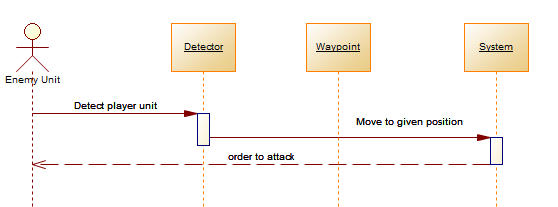
**gambar 3.18** *respawn sequence diagram*

*Sequence diagram* pada gambar menunjukan bahwa respawn mengatur waktu respawn unit dan akan melakukan respawn lagi setelah interval.



**gambar 3.19** *waypoint sequence diagram*

*Sequence diagram* pada gambar menunjukan bagaimana unit musuh bergerak ke waypoint pertama dan jika waypoint tesebut bukan last waypoint maka sistem akan menentukan waypoint selanjutnya dan memberikan perintah untuk unit musuh bergerak ke waypoint selanjutnya.



**gambar 3.20** *Alert sequence diagram*

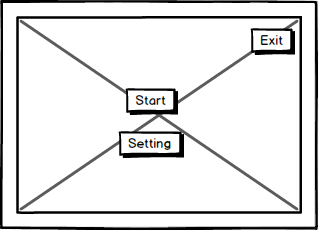
*Sequence diagram* pada gambar menunjukan saat unit musuh mendeteksi unit pemain dan melakukan combat. unit musuh akan memberikan posisi unit pemain dan memberitahukan unit musuh yang ada disekitar jarak deteksi agar unit tersebut mendekat dan melakukan combat denga unit pemain juga

## Rancangan User Interface

*User Interface* atau juga  tampilan visual sebuah produk yang menjembatani sistem dengan pengguna (user) dalam *Game* ini dirancang dengan tampilan *gameplay* sesederhana mungkin sehingga diharapkan untuk mempermudah pemain dalam bermain.

1. Menu Utama

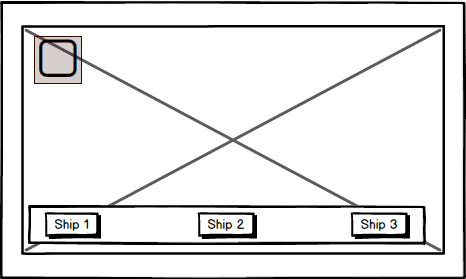
Menu utama merupakan menu pembuka dari *game* dimana akan tersedia beberapa tombol pilihan *menu* seperti *Start*,*Setting*, dan Keluar.

 Gambar 3.21 Rancangan Menu Utama

Keterangan:

1. *Start Button* untuk memulai *game*.
2. *Setting Button* untuk melakukan pengaturan pada aplikasi.
3. *Exit Button* untuk keluar dari aplikasi.
4. User Interface dalam *game*

Interface yang ada saat *game* di mulai dimana ada tombol untuk memilih unit



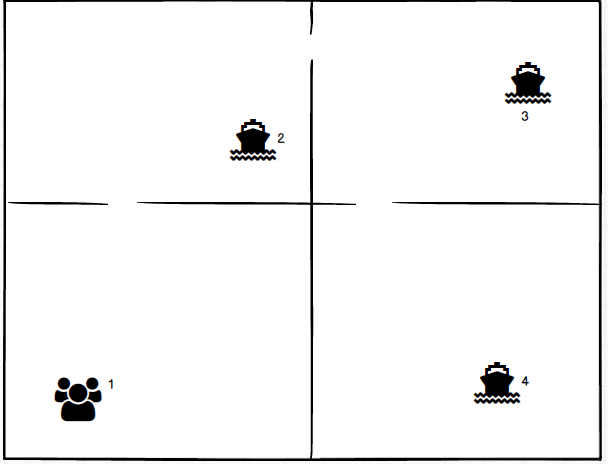
**Gambar 3.22** User Interface dalam *Game*

Keterangan

1. Tombol ship 1 untuk memilih unit pertama.
2. Tombol ship 2 untuk memilih unit kedua
3. Tombol ship 3 untuk memilih unit ketiga.
4. Desain Arena/ *environment*  *Game*

Rancangan Arena Pada *Game*

1. Stage 1



Gambar 3.23 Rancangan Arena/ *environment*  *game*

Keterangan

1. Unit pemain

Posisi unit yang pemain gunankan dalam arena *game*

1. Boss pertama

Posisi boss pertama dalam arena *game*

1. Boss kedua

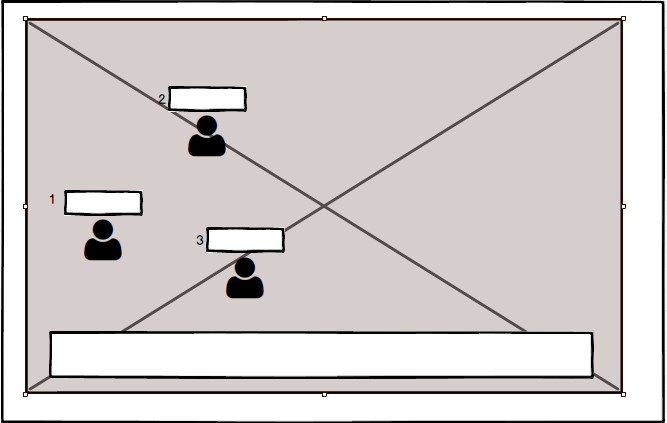
Posisi boss kedua dalam arena *game*

1. Boss ketiga

Posisi boss ketiga dalam arena *game*

1. Rancangan HUD

HUD atau Head Up Display adalah metode yang digunakan untuk menyampaikan informasi secara visual kepada pemain sebagai bagian dari interface pemain *game* contohnya menampilkan *Health Bar* unit.



Gambar 3.24 Rancangan HUD *game*

Keterangan

1. HUD Health Unit 1.
2. HUD Health Unit 2.
3. HUD Health Unit 3.
4. Desain Unit

Desain dari setiap Unit yang ada dalam *game*

1. Unit 1

Desain unit pertama pemain

Tabel 3.10 Model tiga dimensi unit

|  |  |  |
| --- | --- | --- |
| Model tiga dimensi Unit | Keterangan | |
| C:\Users\Antorias\Pictures\Skripso\ship1.PNG | Nama | Unit 1(Crusier) |
| Unit | Pemain |
| Karakteristik | 1.kecil  2. Health sedikit  3. Attack Speed Cepat |

1. Unit 2

Desain unit kedua pemain

Tabel 3.11 Model tiga dimensi unit 2

|  |  |  |
| --- | --- | --- |
| **Model tiga dimensi Unit** | **Keterangan** | |
| **C:\Users\Antorias\Pictures\Skripso\Ship2.PNG** | **Nama** | Unit 2 |
| **Unit** | Pemain |
| **Karakteristik** | 1.sedang  2. Health normal  3. Attack Speed normal |

1. Unit 3

Desain Pada unit ketiga yang akan digunkan oleh pemain.

Tabel 3.12 Model tiga dimensi unit 3

|  |  |  |
| --- | --- | --- |
| **Model tiga dimensi Unit** | **Keterangan** | |
| **C:\Users\Antorias\Pictures\Skripso\Ship3.PNG** | **Nama** | Unit 3(destroyer) |
| **Unit** | Pemain |
| **Karakteristik** | 1.besar  2. Health banyak  3. Attack Speed lambat |

1. Enemy

Desain unit Musuh yang akan di temukan dan dihadapi pada stage 1 dan 2.

Tabel 3.13 Model tiga dimensi unit musuh

|  |  |  |
| --- | --- | --- |
| **Model tiga dimensi Unit** | **Keterangan** | |
| **C:\Users\Antorias\Pictures\Skripso\enemy.PNG** | **Nama** | Enemy ship |
| **Unit** | enemy |
| **Karakteristik** | 1.kecil  2. Health sedikit  3. Attack Speed normal |

1. Boss

Desain unit Musuh yang akan di temukan dan dihadapi pada setiap stage dan unit yang perlu dihancurkan untuk mendapatkan *winning condition*.

Tabel 3.14 Model tiga dimensi boss

|  |  |  |
| --- | --- | --- |
| **Model tiga dimensi Unit** | **Keterangan** | |
| **C:\Users\Antorias\Pictures\Skripso\boss1.PNG** | **Nama** | Boss |
| **Unit** | Enemy |
| **Karakteristik** | 1.besar  2. Health banyak  3. Attack Speed lambat |

## Musik dan Audio

Musik dan audio yang digunakan dalam *game* terdiri dari dua macam, yaitu musik latar (*backsound*) dan musik efek (*sound effect*). Ada beberapa musik efek yang digunakan dalam *game* seperti suara tombol dan ledakan.

## Mekanisme Pengujian(Testing)

Setelah proses pembuatan game selesai dilakukan maka tahap selanjutnya yang dilakukan adalah menguji coba program. Dimana dalam tahap ini dilakukan pengujian terhadap aplikasi atau program yang telah dibuat dengan cara menjalankan aplikasi atau program tersebut dan dilihat apakah terdapat kesalahan atau tidak. pengujian akan terbagi 2 yaitu pengujian fungsional yang dimana akan menguji system dan AI dalam game dan pengujian fitur yang dimana akan menguji fitur-fitur yang akan digunakan dalam game seperti tombol pause dan pengaturan volume.