BAB 3

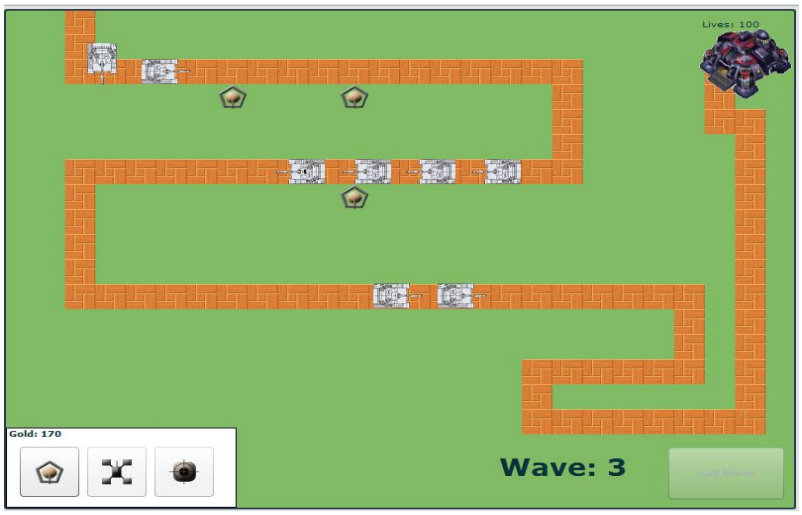
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Pada bagian analisis sistem ini, akan dilakukan berbagai macam analisis yang berhubungan dengan *game* yang akan dibangun, mulai dari analisis terhadap game sejenis analisis berbagai kebutuhan baik fungsional maupun non-fungsional.

3.1.1 Analisis Aplikasi Permainan Sejenis

Dalam hal ini diperlukan sebuah analisis yang dilakukan terhadap aplikasi permainan lain yang sejenis untuk menentukan dimana letak kekurangan aplikasi permainan tersebut, sehingga dapat kita kembangkan dengan membuat aplikasi permainan baru agar semakin menarik dibandingkan dengan aplikasi permainan yang dianalisis. Aplikasi permainan sejenis yang akan dianalisis adalah aplikasi permainan yang berjudul *Simple Tower Defense*, seperti pada gambar 3.1.



Gambar 3.1 *Screenshot simple tower defense*[18]

Simple Tower Defense menawarkan sebuah permainan game ber-*genre tower defense*. Pada game tersebut, musuh terdiri dari 3 jenis dengan kekuatan yang berbeda-beda, musuh sendiri dibedakan dari warnanya. Musuh diharuskan untuk terus maju sampai *goal line* yang berupa bangunan, ketika musuh tersebut sampai pada *goal line* maka nyawa yang tertera di atas bangunan akan berkurang seiring banyaknya musuh yang berhasil masuk ke dalam bangunan.

Pada game ini disediakan tiga jenis tower yang berbeda-beda, baik itu dalam hal kekuatan serang maupun dalam jarak jangkauan serang terhadap musuh. Setiap tower mempunyai harga yang berbeda-beda untuk dapat dipakai oleh *player*, untuk memperoleh uang yang digunakan membeli tower, maka *player* harus membunuh setiap musuh yang ada, dan setiap musuh datang secara bergelombang (*wave*).

3.1.2 Analisis Masalah

Berdasarkan analisis yang dilakukan terhadap *game* sejenis yang berjudul "*Simple Tower Defense"* dapat disimpulkan bahwa dalam *game* tersebut, dalam hal cara bermain masih terlalu monoton, hal ini dibuktikan pada alur permainannya yang masih sederhana, mulai dari tingkatan kesulitan yang hanya dibedakan dalam bentuk gelombang dan misi dari permainan ini yang masih belum jelas akhirnya.

3.1.3 Analisis Pembangunan Aplikasi Permainan *Pirates of Money*

3.1.3.1 Pengenalan

Pirates of Money merupakan nama aplikasi permainan ber-*genre tower defense* yang akan dibuat. Kata *Pirates* dalam aplikasi permainan ini berarti pembajak, dan kata *Money* yang berarti uang. Dalam pemberian nama pada aplikasi permainan ini bertujuan selain memberikan hiburan kepada para pemain, juga memberikan sebuah pengajaran bahwa para pemain dapat menerapkan apa yang ada di dalam aplikasi permainan ini ke dalam dunia nyata, yaitu dengan memerangi setiap pembajak uang yang berusaha menguras habis keuangan negara. Aplikasi permainan ini dibangun dalam bentuk dua dimensi (2D), adapun kelebihan-kelebihan yang terdapat di dalam aplikasi permainan *pirates of money* terdapat pada tabel 3.1.

Tabel 3.1 Kelebihan Aplikasi Permainan

Kelebihan Aplikasi Permainan *Pirates of Money*
1. Terdapat bermacam-macam musuh dan *hero* 2. Penerapan konsep AI pada musuh di aplikasi permainan tersebut pada tingkatan tertentu 3. Terdapat tambahan uang jika *hero* yang ditempatkan oleh pemain berhasil membunuh musuh dan dapat membeli *hero* yang telah disediakan di *shop* sesuai dengan jumlah uang yang dimiliki

3.1.4 Story Line Game

Negara mengalami krisis uang yang berkepanjangan, semua kekayaan lenyap hanya dalam waktu beberapa tahun, dan saat ini yang tersisa hanya sejumlah uang di dalam tempat penyimpanan uang yang disimpan pada brankas besi besar. Namun, disaat keadaan seperti itu muncul penyerangan besar-besaran dari sekelompok musuh, yaitu manusia tikus. Manusia tikus ini sangat menyukai uang dan tujuan utamanya adalah menghabiskan seluruh cadangan kekayaan yang masih ada di negara tersebut. Oleh karena itu, pihak keamanan yang terdiri dari pemuda dan para aparat negara berusaha mempertahankan satu-satunya kekayaan yang tersisa dari negara tersebut agar tidak di ambil oleh para manusia tikus tersebut.

Alur dan misi dari aplikasi permainan *Pirates of Money*, yaitu :

A. Alur Aplikasi Permainan *Pirates of Money*

Alur permainan merupakan ketentuan-ketentuan yang terdapat pada sebuah aplikasi permainan agar setiap pemain dapat mengetahui bagaimana cara untuk memainkan permainan ini dan untuk menyelesaikan misi pada aplikasi permaianan tersebut.

Alur permainan dari aplikasi permainan *Pirates of Money* terdapat pada tabel 3.2.

Tabel 3.2 Alur Aplikasi Permainan

Alur Permainan
1. Musuh bergerak dari *point* A(markas musuh) menuju *point* B(tempat penyimpanan uang) sesuai dengan jalur yang telah disediakan

2. Pada *level* tertentu musuh bergerak dengan menentukan rute terpendek untuk mencapai tujuannya
3. *Hero* yang sudah ditempatkan pada petak otomatis akan menembak musuh yang berada di dekatnya
4. Jika *hero* berhasil membunuh musuh, maka uang permainan akan bertambah dan musuh akan lenyap
5. Jika musuh mencapai tempat penyimpanan uang (brankas), maka darah pemain akan berkurang sesuai dengan kekuatan musuh yang masuk
6. Jika darah pemain menjadi 0 (nol) maka permainan dinyatakan berakhir.
7. Jika musuh berhasil dibunuh sesuai dengan banyaknya musuh yang harus dibunuh sebelum darah pemain menjadi 0 (nol) maka permainan dapat dilanjutkan ke *level* berikutnya

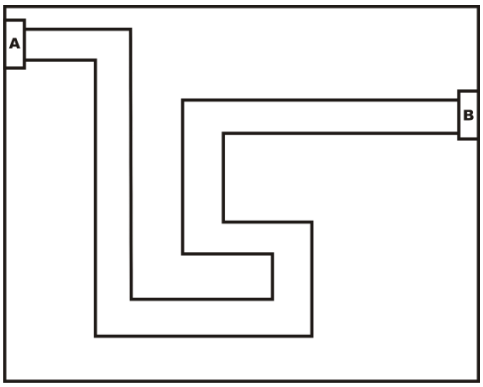
B. Misi Aplikasi Permainan *Pirates of Money*

Misi yang harus diselesaikan pada aplikasi permainan *Pirates of Money* ini yaitu dengan melakukan pertahanan selama waktu yang telah ditentukan, pertahanan yang dilakukan agar musuh yang menyerang tidak mencapai tempat penyimpanan uang (*goal point*).

3.1.5 *Storyboard*

Storyboard merupakan sketsa gambar yang disusun berurutan sesuai naskah yang ada, dan dengan storyboard kita dapat menyampaikan ide cerita kita kepada orang lain dengan mudah.

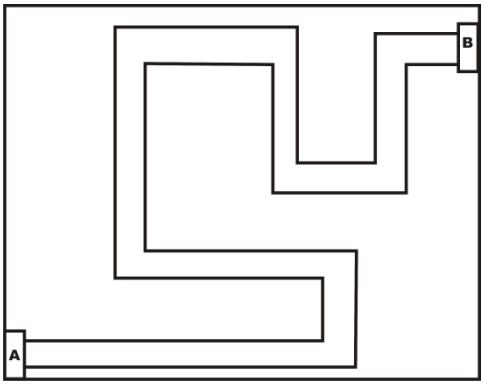
Storyboard level 1 dari aplikasi *game Pirates of Money*, terdapat pada gambar 3.2.



Gambar 3.2 *Storyboard level* 1

Pada *level* 1 ini, musuh akan keluar secara secara berurutan dari *point* A(markas musuh) menuju *point* B(tempat menyimpan uang) sesuai dengan jalur yang telah disediakan, musuh yang muncul pada *level* ini yaitu musuh *level* 1. Setiap pemain pada *level* ini dapat menggunakan semua jenis *hero* untuk melakukan pertahanan dengan cara membeli *hero* yang akan digunakan pada *shop* yang telah tersedia sesuai dengan jumlah uang yang dimiliki. *Hero* yang digunakan hanya bisa ditempatkan pada jalur selain dari jalur yang dilewati musuh.

Storyboard level 2 dari aplikasi *game Pirates of Money*, terdapat pada gambar 3.3.



Gambar 3.3 *Storyboard level* 2

Pada *level* 2 ini, musuh keluar dari *point* A, Musuh yang keluar pada *level* 2 ini adalah musuh dengan *level* 2 yang memiliki kekuatan dan kecepatan lebih dibanding dengan musuh *level* 1. Setiap pemain pada *level* ini dapat menggunakan semua jenis *hero* untuk melakukan pertahanan dengan cara membeli *hero* yang akan digunakan pada *shop* yang telah tersedia sesuai dengan jumlah uang yang dimiliki. *Hero* yang digunakan hanya bisa ditempatkan pada jalur selain dari jalur yang dilewati musuh.

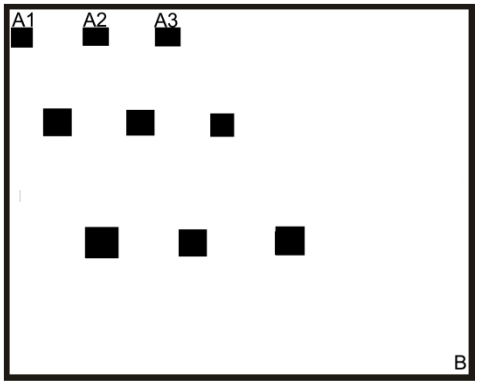
Storyboard level 3 dari aplikasi *game Pirates of Money*, terdapat pada gambar 3.4.



Gambar 3.4 *Storyboard level* 3

Pada *level* 3 ini, musuh sudah diterapkan algoritma A* (A *star*) untuk mencari rute terpendek dari *point* A(markas musuh) untuk menuju ke *point* B(tempat penyimpanan uang) dengan cara menghitung nilai F yang berupa jumlah kotak dari posisi brankas dikurangi dengan jumlah kotak dari posisi musuh berdiri. Setiap pemain pada *level* ini dapat menggunakan semua jenis *hero* untuk melakukan pertahanan dengan cara membeli *hero* yang akan digunakan pada *shop* yang telah tersedia sesuai dengan jumlah uang yang dimiliki. *Hero* yang digunakan bisa ditempatkan disetiap jalur yang ada dengan tujuan untuk menghalangi musuh yang lewat dan ketika jalur yang akan dilewati musuh selanjutnya sudah terdapat penghalang berupa *hero*, maka musuh akan memilih jalur lain dengan nilai F yang terkecil.

Storyboard level 4 dari aplikasi *game Pirates of Money*, terdapat pada gambar 3.5.



Gambar 3.5 *Storyboard level* 4

Pada *level* 4 ini, musuh sudah diterapkan algoritma A* (A *star*) untuk mencari rute terpendek dari *point* A(markas musuh) untuk menuju ke *point* B(tempat penyimpanan uang) dengan cara menghitung nilai F yang berupa jumlah

kotak dari posisi brankas dikurangi dengan jumlah kotak dari posisi musuh berdiri. Setiap pemain pada *level* ini dapat menggunakan semua jenis *hero* untuk melakukan pertahanan dengan cara membeli *hero* yang akan digunakan pada *shop* yang telah tersedia sesuai dengan jumlah uang yang dimiliki. *Hero* yang digunakan bisa ditempatkan disetiap jalur yang ada dengan tujuan untuk menghalangi musuh yang lewat dan ketika jalur yang akan dilewati musuh selanjutnya sudah terdapat penghalang berupa *hero*, maka musuh akan memilih jalur lain dengan nilai F yang terkecil.

3.1.6 Entitas *Game*

Entitas yang terdapat di dalam *game* terbagi menjadi dua, yaitu karakter *hero* dan NPC yang berupa musuh, selain itu terdapat juga item yang digunakan untuk membantu *hero* dalam pertahanan.

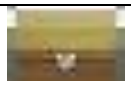
Entitas karakter dan item yang terdapat di dalam game *pirates of money* ditunjukkan pada tabel 3.3.

Tabel 3.3 Entitas karakter dan item

NO	NAMA	GAMBAR	PENJELASAN
1	Pemuda		Pada karakter *hero* ini, merupakan karakter pertama yang dapat dimainkan dengan harga Rp.20. Kekuatan karakter ini termasuk ke dalam *level* biasa saja, dengan menggunakan karakter ini musuh

			yang ditembak akan berkurang darahnya sebanyak satu untuk setiap tembakan.
2	Polisi		Pada karakter *hero* ini, merupakan karakter pertama yang dapat dimainkan dengan harga Rp.30. Kekuatan karakter ini termasuk ke dalam *level* 2, dengan menggunakan karakter ini musuh yang ditembak akan berkurang darahnya sebanyak tiga untuk setiap tembakan.
3	Tentara		Pada karakter *hero* ini, merupakan karakter pertama yang dapat dimainkan dengan harga Rp.40. Kekuatan karakter ini termasuk ke dalam *level* 3, dengan menggunakan karakter ini musuh yang ditembak akan berkurang darahnya sebanyak lima untuk setiap tembakan.
4	Taito		Pada NPC (*Non-Player Characters*) ini, merupakan

			karakter musuh pertama yang akan menyerang brankas, ketika musuh ini mencapai brankas maka darah *player* akan berkurang 10 point.
5	Otto		Pada NPC (*Non-Player Characters*) ini, merupakan karakter musuh dengan *level* kesulitan 2 karena lebih kuat dibanding musuh pertama, ketika musuh ini mencapai bank, maka darah *player* akan berkurang 15 point.
6	Braket		Pada NPC (*Non-Player Characters*) ini, merupakan karakter musuh dengan *level* kesulitan 3 karena lebih kuat dibanding musuh pertama, ketika musuh ini mencapai bank, maka darah *player* akan berkurang 20 point.
7	Bigtrob		Pada NPC (*Non-Player Characters*) ini, merupakan karakter musuh dengan *level*

			kesulitan 3 karena merupakan karakter penutup *level* (*boss level*), *ketika* musuh ini mencapai bank, maka darah *player* akan berkurang 30 point.
8	Meriam		Item ini merupakan pilihan pertahanan yang lumayan membantu, yaitu dengan harga Rp. 10, adapun untuk kekuatan serang sama dengan karakter *hero* pemuda.
9	Peluru Laser		Jenis peluru ini adalah peluru yang ditembakkan oleh setiap *hero* yang pemain pasang.
10	Brankas		Brankas merupakan tempat penyimpanan uang terakhir dari suatu negara pada aplikasi permainan ini yang harus dilindungi oleh para *hero* dan merupakan tujuan akhir dari musuh

3.1.7 Analisis Metode dan Algoritma *Pirates of Money*

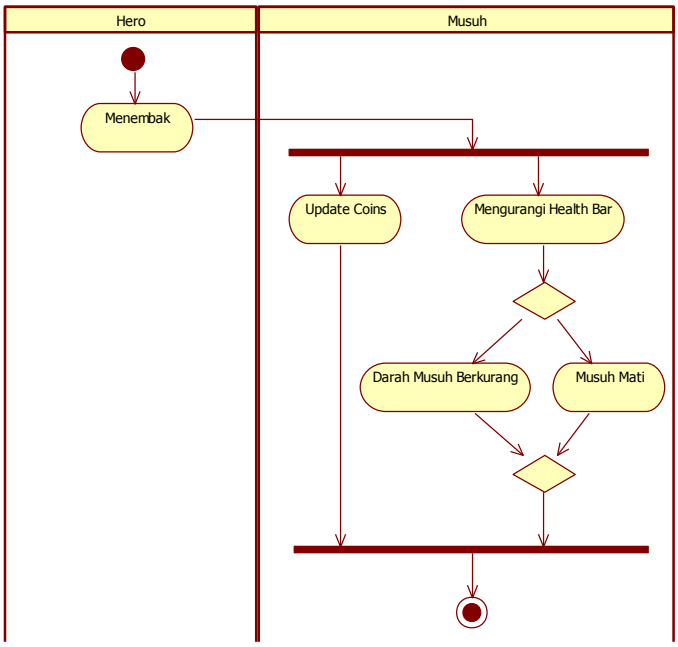
Pada aplikasi permainan *pirates of money* ini, diterapkan metode dan algoritma yang berguna untuk membuat aplikasi permainan ini menjadi lebih menarik dan menantang.

3.1.7.1 Penerapan Metode *Collision Detection Pirates of Money*

Agar aplikasi permainan *Pirates of Money* ini menjadi lebih menarik, maka akan diterapkan metode *collision detection* yang berupa metode dimana yang bersinggungan merupakan objek secara langsung, baik itu *sprite* dengan *sprite* maupun *sprite* dengan objek lain.

Pada aplikasi permainan *Pirates of Money*, metode yang akan diterapkan pada peluru yang ditembakan oleh para *hero* untuk membunuh setiap musuh yang ada adalah metode *collision* antara objek dengan objek yang terjadi apabila objek tersebut saling bersentuhan pada titik yang sama.

Diagram *Activity* metode *collision* peluru dengan musuh ditunjukkan pada gambar 3.6.



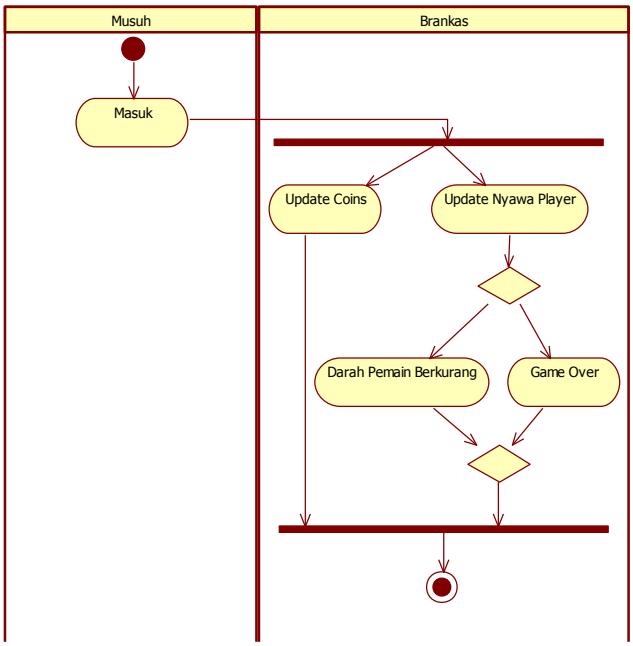
Gambar 3.6 Diagram *Activity* Peluru dengan Musuh *Pirates of Money*

Pseudo code collision detection peluru dengan musuh pada aplikasi permainan *Pirates of Money* dapat dilihat pada tabel 3.4.

Table 3.4 *Pseudo Code Collision Detection* Peluru dengan Musuh

```
1   If shotmob != -1 && kotaktower terdapat mob
2      Hero menembak
    Else
3      Hero tidak menembak
4   End if
```

Diagram *activity* metode *collision* musuh dengan brankas ditunjukkan pada gambar 3.7.



Gambar 3.7 Diagram *Activity* Musuh dengan Brankas *Pirates of Money*

Pseudo code collision detection musuh dengan brankas pada aplikasi permainan *Pirates of Money* dapat dilihat pada tabel 3.5.

Tabel 3.5 *Pseudo Code Collision Detection* Musuh dengan Brankas

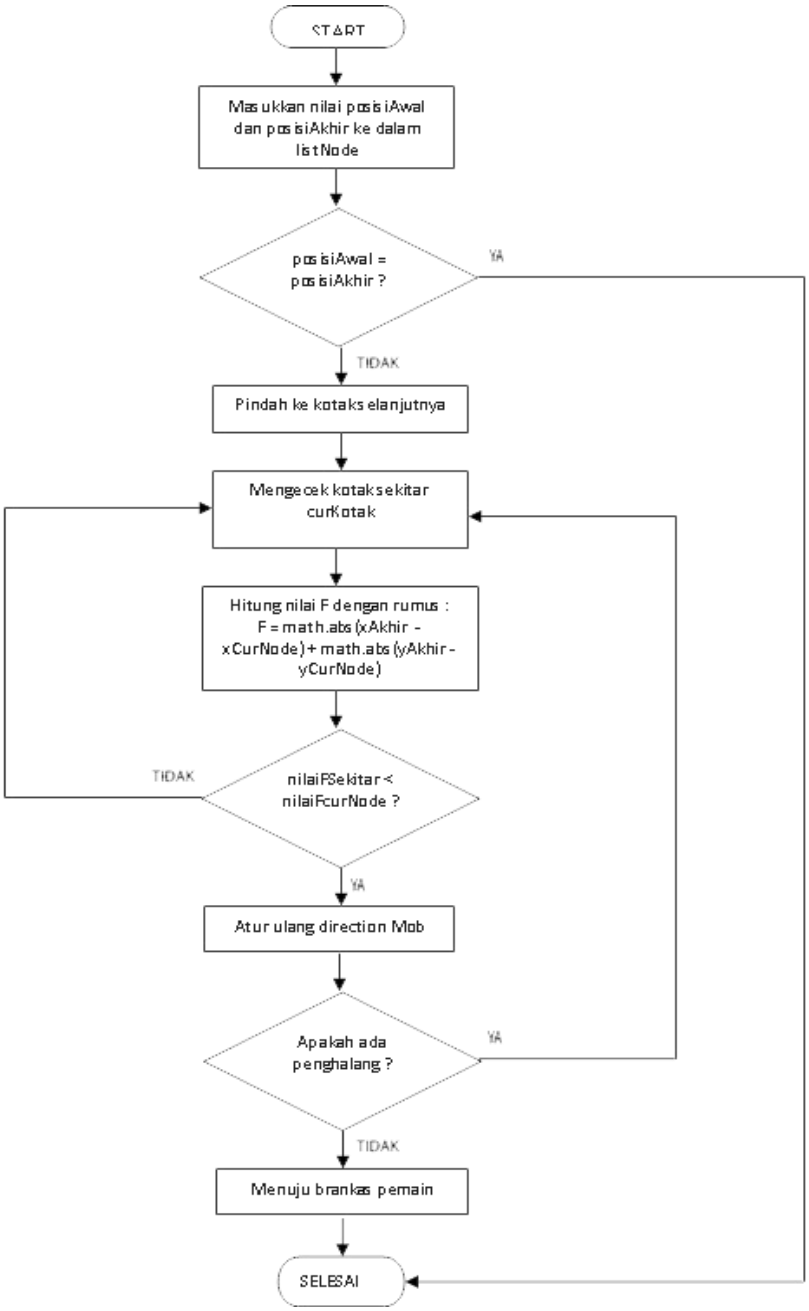
1	`If airID = airCave`
2	`Mob hilang`
	`Darah pemain berkurang`
	`else`
3	`Atur arah jalan mob`
4	`End if`

3.1.7.2 Penerapan Algoritma A* (A *Star*) *Pirates of Money*

Disamping menerapkan metode *collision detection* pada aplikasi permainan *Pirates of Money*, pada *level* tertentu juga diterapkan algoritma A* (A

star) pada musuh sebagai *pathfinding*, yaitu untuk mencari rute terpendek pada musuh dimulai dari markas musuh menuju tempat penyimpanan uang.

Flowchart dari algoritma A* (A star) pada aplikasi permainan *Pirates of Money* ditunjukkan pada gambar 3.8.



Gambar 3.8 *Flowchart* algoritma A* *Pirates of Money*

A. Analisis Algoritma A* (A *star*) *level* 3

Analisis algoritma A* (A *star*) pada *level* 3 yang terdapat di aplikasi permainan *pirates of money* ini digambarkan melalui sebuah tahapan proses

dimulai ketika musuh muncul dari markas (*point* A) menuju tempat tersimpannya brankas (*point* B), musuh hanya bergerak secara vertikal dan horizontal.

Berikut merupakan tahapan proses yang terjadi pada *level* 3 aplikasi permainan *pirates of money* yang menerapkan algoritma A* (A *star*) sebagai *pathfinding* dari markas menuju tempat penyimpanan brankas :

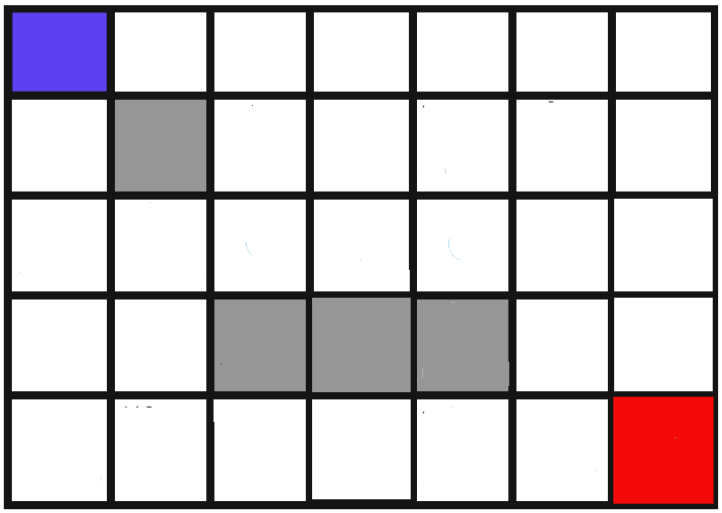
Keterangan :

: Markas musuh (*point* A)

: Tempat penyimpanan brankas (*point* B)

: *Hero* yang ditempatkan oleh pemain

Map untuk *level* 3 terdapat pada gambar 3.9.



Gambar 3.9 Map *level* 3 *Pirates of Money*

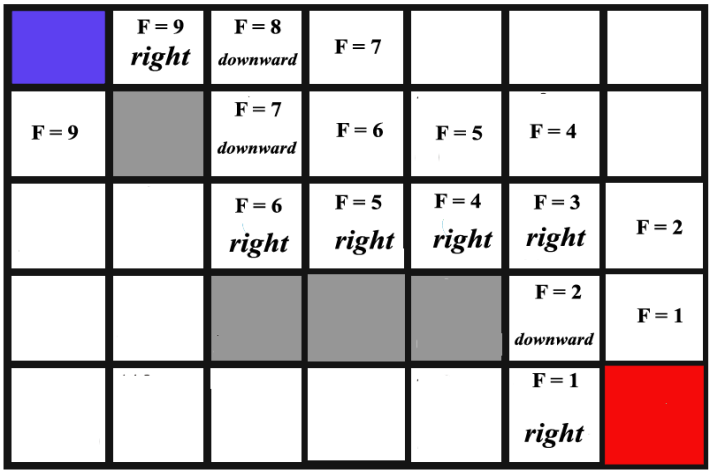
Disaat musuh mulai bergerak maju dan berpindah dari satu kotak ke kotak lainnya, maka lakukan pengecekan kotak sekitar musuh berdiri, yaitu dengan

rumus : F = ((x kotak tujuan – x kotak musuh berdiri) + (y kotak musuh tujuan – y kotak musuh berdiri)).

Setelah melakukan pengecekan kotak sekitar, kotak sekitar yang terisi oleh *hero* maupun oleh brankas maka tidak dilakukan perhitungan, dan untuk kotak sekitar yang kosong maka dilakukan perhitungan pada setiap musuh berpindah dari satu kotak ke kotak lainnya.

Pada setiap perhitungan kotak sekitar, dilakukan perbandingan antara nilai F (kotak tempat musuh berdiri) dengan nilai F (kotak awal), dari hasil perbandingan tersebut maka *direction* musuh diubah berdasarkan nilai F yang paling kecil dari hasil perbandingan tersebut.

Hasil dari perhitungan kotak sekitar disaat musuh bergerak satu langkah ke depan dan pemain menempatkan *hero* untuk menghalangi musuh, terdapat pada gambar 3.10.



Gambar 3.10 Penerapan algoritma A* pada *Pirates of Money*

Pseudo code untuk penerapan algoritma A* (A *star*) *level* 3 pada *Pirates of Money* dapat dilihat pada tabel 3.6.

Tabel 3.6 *Pseudo Code* Algoritma A* pada *Level* 3

```
1   Membuat list node baru
2   If current node posisi x-1 >= 0
3      If current node posisi x-1 = node akhir posisi x &&
4         Current node posisi y = node akhir posisi y ||
5         airID posisi x-1 != airPemudaLaser &&
          airID posisi x-1 != airPenghalang &&
          airID posisi x-1 != airPoliceLaser &&
          airID posisi x-1 != airTentaraLaser &&
          airID posisi x-1 != airMeriamLaser
6            F posisi jalan = (x akhir-x current posisi)+
                (y akhir-y current posisi)
             F posisi awal = (x akhir-x current posisi)+
                (y akhir-y current posisi)
7            If F posisi jalan > F posisi awal
8               Atur direction kiri
9            End if
10     End if
11  End if


12  If current node posisi x+1 >= 0
13     If current node posisi x+1 = node akhir posisi x &&
14        Current node posisi y = node akhir posisi y ||
15        airID posisi x+1 != airPemudaLaser &&
          airID posisi x+1 != airPenghalang &&
          airID posisi x+1 != airPoliceLaser &&
          airID posisi x+1 != airTentaraLaser &&
          airID posisi x+1 != airMeriamLaser
16           F posisi jalan = (x akhir-x current posisi)+
```

```
            (y akhir-y current posisi)
17          F posisi awal = (x akhir-x current posisi)+
               (y akhir-y current posisi)
18          If F posisi jalan < F posisi awal
19             Atur direction kanan
20          End if
21    End if
22 End if

23 If current node posisi y-1 >= 0
24    If current node posisi x = node akhir posisi x &&
25       Current node posisi y-1 = node akhir posisi y ||
         airID posisi y-1 != airPemudaLaser &&
         airID posisi y-1 != airPenghalang &&
         airID posisi y-1 != airPoliceLaser &&
         airID posisi y-1 != airTentaraLaser &&
         airID posisi y-1 != airMeriamLaser
26          F posisi jalan = (x akhir-x current posisi)+
               (y akhir-y current posisi)
            F posisi awal = (x akhir-x current posisi)+
               (y akhir-y current posisi)
27          If F posisi jalan > F posisi awal
28             Atur direction atas
29          End if
30    End if
31 End if

32 If current node posisi y+1 >= 0
33    If current node posisi x = node akhir posisi x &&
34       Current node posisi y+1 = node akhir posisi y ||
```

```
35        airID posisi y+1 != airPemudaLaser &&
          airID posisi y+1 != airPenghalang &&
          airID posisi y+1 != airPoliceLaser &&
          airID posisi y+1 != airTentaraLaser &&
          airID posisi y+1 != airMeriamLaser
36           F posisi jalan = (x akhir-x current posisi)+
                (y akhir-y current posisi)
37           F posisi awal = (x akhir-x current posisi)+
                (y akhir-y current posisi)
38           If F posisi jalan < F posisi awal
39              Atur direction kanan
40           End if
41     End if
42  End if
```

B. Analisis Algoritma A* (A *star*) *Level* 4

Pada aplikasi permainan *pirates of money level* 4 ini, dimulai dari munculnya musuh dari tiga *point* awal yang telah ditentukan sebelumnya menuju satu *point* tujuan, pada *level* ini juga ditempatkan penghalang permanen sebagai blok yang tidak dapat dilewati oleh musuh yang ada, musuh yang ada hanya dapat bergerak vertikal dan horizontal saja.

Berikut merupakan tahapan proses yang terjadi pada *level* 4 aplikasi permainan *pirates of money* yang menerapkan algoritma A* (A *star*) sebagai *pathfinding* dari markas menuju tempat penyimpanan brankas :

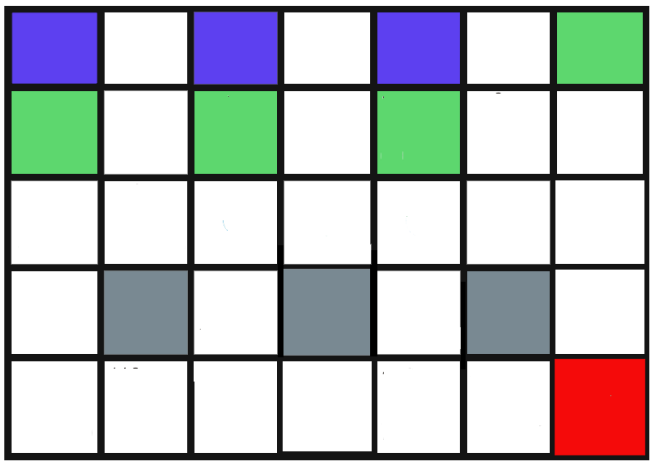
 : Markas musuh (*point* A)

: Tempat penyimpanan brankas (*point* B)

: *Hero* yang ditempatkan oleh pemain

: Penghalang permanen

Map untuk *level* 4 terdapat pada gambar 3.11.



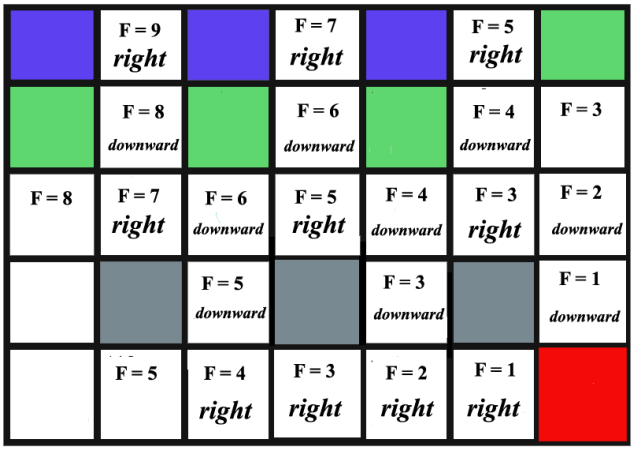
Gambar 3.11 Map *Level* 4 *Pirates of Money*

Disaat musuh mulai bergerak maju dan berpindah dari satu kotak ke kotak lainnya, maka lakukan pengecekan kotak sekitar musuh berdiri, yaitu dengan rumus : F = ((x kotak tujuan – x kotak musuh berdiri) + (y kotak musuh tujuan – y kotak musuh berdiri)).

Perhitungan dilakukan disetiap musuh berpindah kotak, pada *level* 4 ini terdapat dua penghalang, yaitu penghalang permanen berupa batu dan penghalang berupa *hero*. Proses perhitungan nilai F untuk setiap kotak sekitar dilakukan hanya dilakukan pada kotak yang kosong atau tidak terdapat batu maupun *hero*.

Setiap nilai F disetiap kotak sekitar, musuh akan bergerak secara otomatis ke kotak yang mengandung nilai F terkecil. Hasil akhir perhitungan untuk setiap

musuh yang berjalan untuk menuju *point* akhir yaitu brankas, dapat dilihat pada gambar 3.12.



Gambar 3.12 Hasil Akhir Perhitungan Map *Level* 4

Pseudo code untuk penerapan algoritma A* (A *star*) *level* 4 pada *Pirates of Money* dapat dilihat pada tabel 3.7.

Tabel 3.7 *Pseudo Code* Algoritma A* *Level* 4

```
1   Membuat list node baru
2   If current node posisi x-1 >= 0
3      If current node posisi x-1 = node akhir posisi x &&
          Current node posisi y = node akhir posisi y ||
          airID posisi x-1 != airPemudaLaser &&
          airID posisi x-1 != airPenghalang &&
          airID posisi x-1 != airPoliceLaser &&
          airID posisi x-1 != airTentaraLaser &&
          airID posisi x-1 != airMeriamLaser
4            F posisi jalan = (x akhir-x current posisi)+
                (y akhir-y current posisi)
             F posisi awal = (x akhir-x current posisi)+
                (y akhir-y current posisi)
5            If F posisi jalan > F posisi awal
6               Atur direction kiri
7            End if
8      End if
9   End if

10  If current node posisi x+1 >= 0
11     If current node posisi x+1 = node akhir posisi x &&
          Current node posisi y = node akhir posisi y ||
          airID posisi x+1 != airPemudaLaser &&
          airID posisi x+1 != airPenghalang &&
          airID posisi x+1 != airPoliceLaser &&
          airID posisi x+1 != airTentaraLaser &&
          airID posisi x+1 != airMeriamLaser
12           F posisi jalan = (x akhir-x current posisi)+
```

```
            (y akhir-y current posisi)
         F posisi awal = (x akhir-x current posisi)+
            (y akhir-y current posisi)
13       If F posisi jalan < F posisi awal
14          Atur direction kanan
15       End if
16    End if
17 End if

18 If current node posisi y-1 >= 0
19    If current node posisi x = node akhir posisi x &&
         Current node posisi y-1 = node akhir posisi y ||
         airID posisi y-1 != airPemudaLaser &&
         airID posisi y-1 != airPenghalang &&
         airID posisi y-1 != airPoliceLaser &&
         airID posisi y-1 != airTentaraLaser &&
         airID posisi y-1 != airMeriamLaser
20          F posisi jalan = (x akhir-x current posisi)+
               (y akhir-y current posisi)
            F posisi awal = (x akhir-x current posisi)+
               (y akhir-y current posisi)
21          If F posisi jalan > F posisi awal
22             Atur direction atas
23          End if
24    End if
25 End if

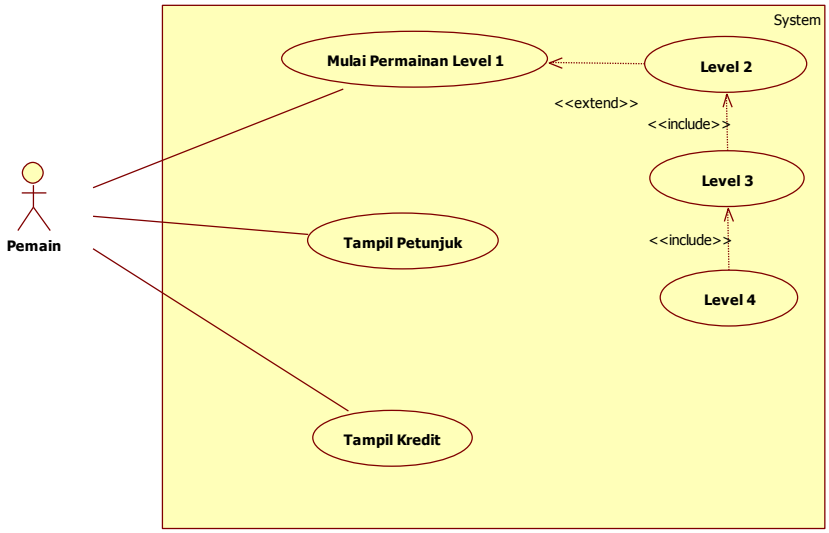
26 If current node posisi y+1 >= 0
27    If current node posisi x = node akhir posisi x &&
         Current node posisi y+1 = node akhir posisi y ||
```

```
      airID posisi y+1 != airPemudaLaser &&
      airID posisi y+1 != airPenghalang &&
      airID posisi y+1 != airPoliceLaser &&
      airID posisi y+1 != airTentaraLaser &&
      airID posisi y+1 != airMeriamLaser
28        F posisi jalan = (x akhir-x current posisi)+
             (y akhir-y current posisi)
          F posisi awal = (x akhir-x current posisi)+
             (y akhir-y current posisi)
29        If F posisi jalan < F posisi awal
30           Atur direction kanan
31        End if
32    End if
33 End if
```

3.1.8 Analisis dan Kebutuhan Fungsional

3.1.8.1 Diagram *Use Case*

Secara umum, interaksi antara pengguna yaitu pemain dengan aplikasi permainan *pirates of money* digambarkan pada diagram *Use Case*, seperti pada gambar 3.13.



Gambar 3.13 Diagram *Use Case Pirates of Money*

3.1.8.2 Skenario *Use Case*

Skenario pada diagram *use case* berisi tentang alur kejadian '*flow of event*' *use case* utama. Di dalamnya menjelaskan urutan interaksi antara aktor dengan *use case* tersebut dari awal sampai akhir. Adapun scenario *use case* yang terdapat pada aplikasi permainan *pirates of money* ini adalah sebagai berikut :

1. Skenario *Use Case* Mulai Permainan Level 1

Skenario *use case* ini menjelaskan interaksi antara aktor, yaitu pemain, dengan *use case* mulai permainan level 1 yang dijelaskan pada tabel 3.8.

Nama *Use Case* : Mulai Permainan Level 1

Skenario :

Tabel 3.8 Skenario *Use Case* Mulai Permainan *Level* 1

Aksi Aktor	Reaksi Sistem
Skenario Normal	
a. Klik Mulai Permainan	
	b. Tampilkan layar permainan *level* 1
	c. Inisialisasi informasi awal pemain
	d. inisialisasi posisi awal musuh dan posisi brankas
e. Menempatkan hero	
f. Menyelesaikan misi permainan	
	g. *Update* status permainan
	h. Menampilkan informasi untuk berpindah ke *level* selanjutnya
Skenario Alternatif	
a. Klik Mulai Permainan	
	b. Tampilkan layar permainan *level* 1
	c. Inisialisasi informasi awal pemain
	d. inisialisasi posisi awal musuh dan posisi brankas
e. Menempatkan hero	
f. Misi permainan tidak terselesaikan	

	g. Kembali ke menu utama
h. Klik Mulai Permainan	
	i. Tampilkan layar permainan *level* 1
	j. Inisialisasi informasi awal pemain
	k. inisialisasi posisi awal musuh dan posisi brankas
l. Menempatkan hero	
m. Menyelesaikan misi permainan	
	n. *Update* status permainan
	o. Menampilkan informasi untuk berpindah ke *level* selanjutnya

2. Skenario *Use Case Level* 2

Skenario *use case* ini menjelaskan interaksi antara aktor, yaitu pemain, dengan *use case level* 2 yang dijelaskan pada tabel 3.9.

Nama *Use Case* : Level 2

Skenario :

Tabel 3.9 Skenario *Use Case Level* 2

Aksi Aktor	**Reaksi Sistem**
Skenario Normal	
a. Klik Mulai Permainan	
	b. Tampilkan layar permainan *level* 2
	c. Inisialisasi informasi awal pemain

	d. inisialisasi posisi awal musuh dan posisi brankas
e. Menempatkan hero	
f. Menyelesaikan misi permainan	
	g. *Update* status permainan
	h. Menampilkan informasi untuk berpindah ke *level* selanjutnya
Skenario Alternatif	
a. Klik Mulai Permainan	
	b. Tampilkan layar permainan *level* 2
	c. Inisialisasi informasi awal pemain
	d. inisialisasi posisi awal musuh dan posisi brankas
e. Menempatkan hero	
f. Misi permainan tidak terselesaikan	
	g. Kembali ke menu utama
h. Klik Mulai Permainan	
	i. Tampilkan layar permainan *level* 2
	j. Inisialisasi informasi awal pemain
	k. inisialisasi posisi awal musuh dan posisi brankas

l. Menempatkan hero	
m. Menyelesaikan misi permainan	
	n. *Update* status permainan
	o. Menampilkan informasi untuk berpindah ke *level* selanjutnya

3. Skenario *Use Case Level* 3

Skenario *use case* ini menjelaskan interaksi antara aktor, yaitu pemain, dengan *use case level* 3 yang dijelaskan pada tabel 3.10.

Nama *Use Case* : Level 3

Skenario :

Tabel 3.10 Skenario *Use Case Level* 3

Aksi Aktor	**Reaksi Sistem**
Skenario Normal	
a. Klik Mulai Permainan	
	b. Tampilkan layar permainan *level* 3
	c. Inisialisasi informasi awal pemain
	d. inisialisasi posisi awal musuh dan posisi brankas
e. Menempatkan hero	
f. Menyelesaikan misi permainan	
	g. *Update* status permainan
	h. Menampilkan informasi untuk

	berpindah ke *level* selanjutnya
Skenario Alternatif	
a. Klik Mulai Permainan	
	b. Tampilkan layar permainan *level* 3
	c. Inisialisasi informasi awal pemain
	d. inisialisasi posisi awal musuh dan posisi brankas
e. Menempatkan hero	
f. Misi permainan tidak terselesaikan	
	g. Kembali ke menu utama
h. Klik Mulai Permainan	
	i. Tampilkan layar permainan *level* 3
	j. Inisialisasi informasi awal pemain
	k. inisialisasi posisi awal musuh dan posisi brankas
l. Menempatkan hero	
m. Menyelesaikan misi permainan	
	n. *Update* status permainan
	o. Menampilkan informasi untuk berpindah ke *level* selanjutnya

4. Skenario *Use Case Level* 4

Skenario *use case* ini menjelaskan interaksi antara aktor, yaitu pemain, dengan *use case level* 4 yang dijelaskan pada tabel 3.11.

Nama *Use Case* : Level 4

Skenario :

Tabel 3.11 Skenario *Use Case Level* 4

Aksi Aktor	**Reaksi Sistem**
Skenario Normal	
a. Klik Mulai Permainan	
	b. Tampilkan layar permainan *level* 4
	c. Inisialisasi informasi awal pemain
	d. inisialisasi posisi awal musuh dan posisi brankas
e. Menempatkan hero	
f. Menyelesaikan misi permainan	
	g. *Update* status permainan
	h. Menampilkan informasi untuk berpindah ke *level* selanjutnya
Skenario Alternatif	
a. Klik Mulai Permainan	
	b. Tampilkan layar permainan *level* 4
	c. Inisialisasi informasi awal pemain
	d. inisialisasi posisi awal musuh dan

	posisi brankas
e. Menempatkan hero	
f. Misi permainan tidak terselesaikan	
	g. Kembali ke menu utama
h. Klik Mulai Permainan	
	i. Tampilkan layar permainan *level* 4
	j. Inisialisasi informasi awal pemain
	k. inisialisasi posisi awal musuh dan posisi brankas
l. Menempatkan hero	
m. Menyelesaikan misi permainan	
	n. *Update* status permainan
	o. Menampilkan informasi untuk berpindah ke *level* selanjutnya

5. Skenario *Use Case* Tampil Petunjuk

Skenario *use case* ini menjelaskan interaksi antara aktor, yaitu pemain, dengan seluruh *use case* tampil petunjuk yang dijelaskan pada tabel 3.12.

Nama *Use Case* : Tampil Petunjuk

Skenario :

Tabel 3.12 Skenario *Use Case* Tampil Petunjuk

Aksi Aktor	Reaksi Sistem
Skenario Normal	
a. Klik Petunjuk	
	b. Tampil layar petunjuk permainan
Skenario Alternatif	
a. Klik Petunjuk	
	b. Layar petunjuk permainan tidak tampil
	c. Kembali ke menu utama
d. Klik Petunjuk	
	e. Tampil layar petunjuk permainan

6. Skenario *Use Case* Kredit

Skenario *use case* ini menjelaskan interaksi antara aktor, yaitu pemain, dengan seluruh *use case* Kredit yang dijelaskan pada tabel 3.13.

Nama *Use Case* : Tampil Kredit

Skenario :

Tabel 3.13 Skenario *Use Case* Tampil Kredit

Aksi Aktor	Reaksi Sistem
Skenario Normal	
a. Klik Kredit	
	b. Tampil layar kredit permainan

Skenario Alternatif	
a. Klik Kredit	
	b. Layar kredit permainan tidak tampil
	c. Kembali ke menu utama
d. Klik Kredit	
	e. Tampil layar kredit permainan

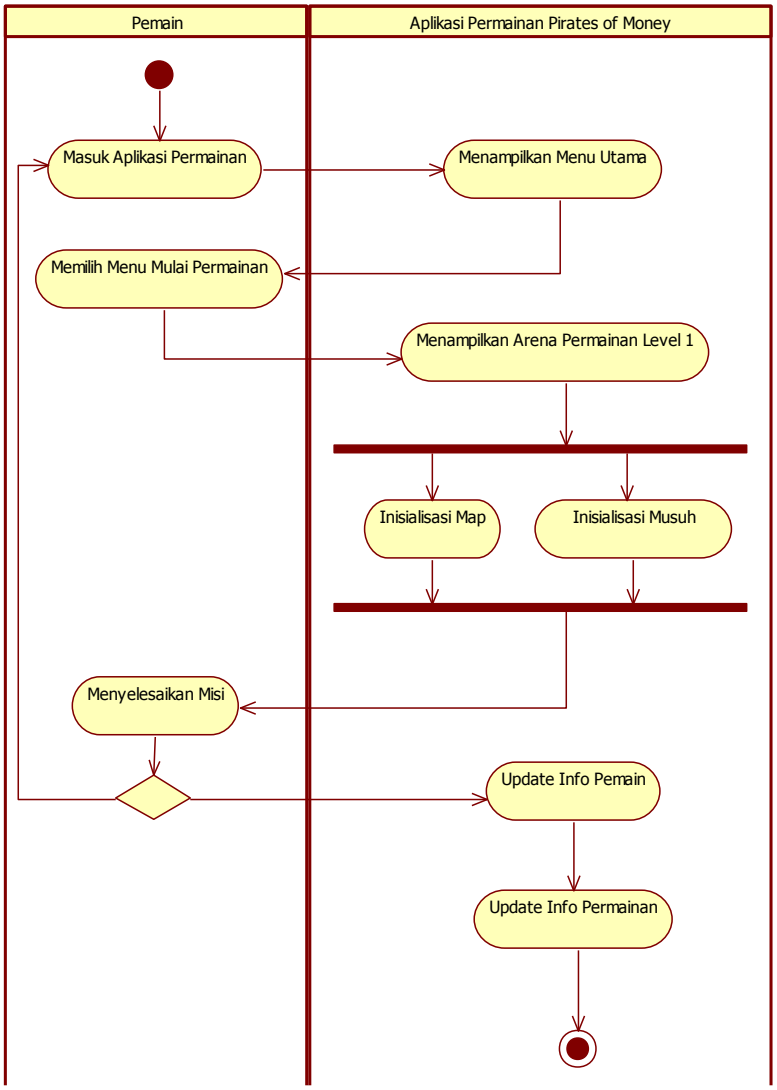
3.1.8.3 Diagram *Activity*

Diagram ini berisi penjelasan secara lengkap mengenai urutan aktivitas yang terjadi antara pemain dengan aplikasi permainan *Pirates of Money*.

A. Diagram *Activity* Mulai Permainan *Level* 1

Proses ini terjadi saat pemain memilih menu mulai permainan pada menu utama aplikasi permainan *Pirates of Money*. Sistem akan menampilkan arena permainan dimana pemain dapat langsung memainkan *level* 1 dan menyelesaikan misi permainan pada *level* 1, apabila misi tidak terselesaikan maka pemain kembali ke menu utama untuk memulai permainan *level* 1 lagi.

Diagram *activity* mulai permainan *level* 1 terdapat pada gambar 3.14.



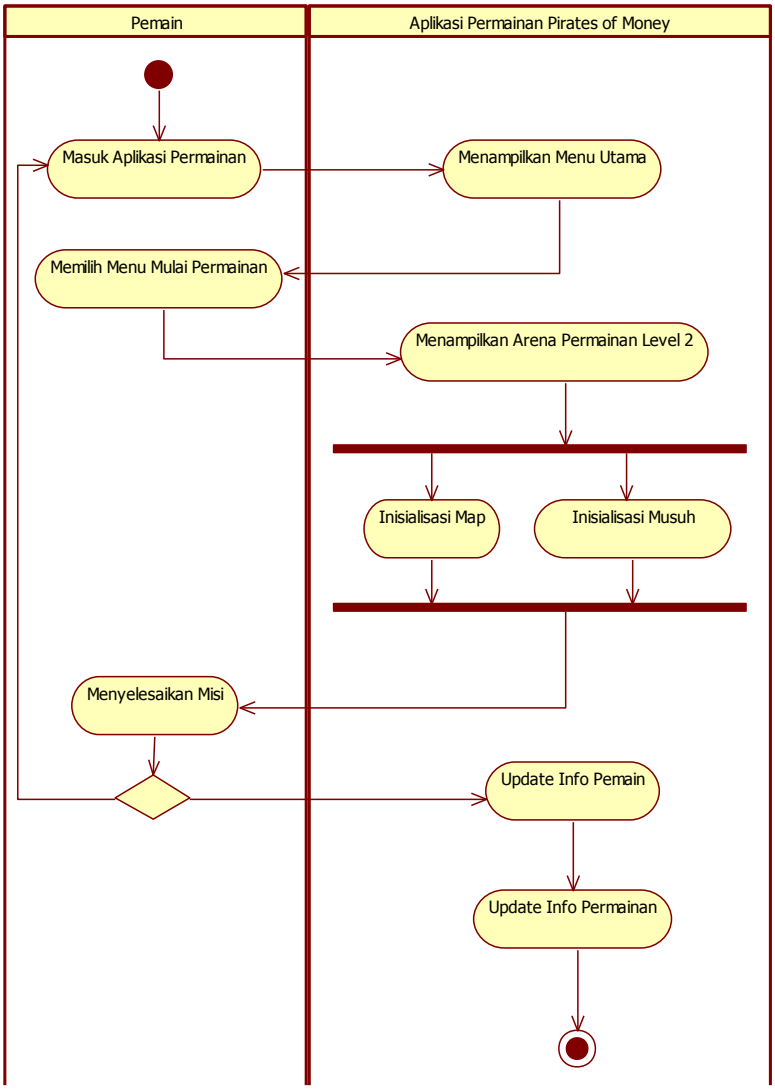
Gambar 3.14 Diagram *Activity* Mulai Permainan *Level* 1 *Pirates of Money*

B. Diagram *Activity Level* 2

Proses ini terjadi saat pemain telah menyelesaikan aplikasi permainan *Pirates of Money* pada *level* 1. Sistem akan menampilkan arena permainan *level* 2 dan pemain harus menyelesaikan misi permainan pada *level* 2, apabila misi permainan

tidak terselesaikan maka pemain akan kembali ke menu utama untuk memulai permainan dari *level* 1 lagi.

Diagram *activity level* 2 terdapat pada gambar 3.15.

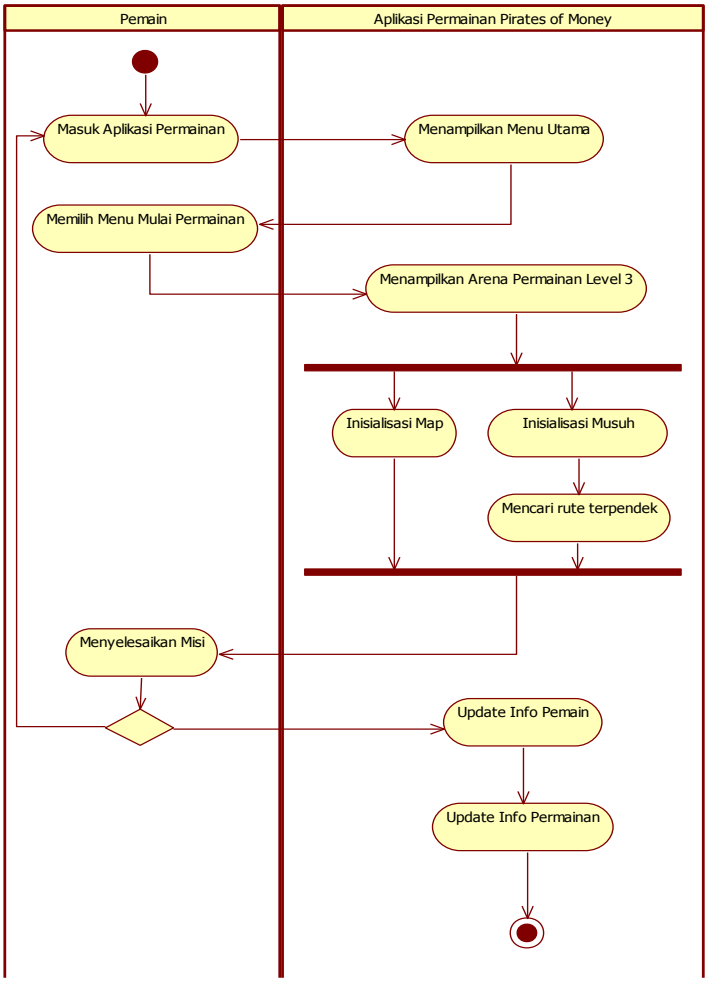


Gambar 3.15 Diagram *Activity Level* 2 *Pirates of Money*

C. Diagram Activity *Level* 3

Proses ini terjadi saat pemain telah menyelesaikan aplikasi permainan *Pirates of Money* pada *level* 2. Sistem akan menampilkan arena permainan *level* 3 dan pemain harus menyelesaikan misi permainan pada *level* 3, apabila misi permainan tidak terselesaikan maka pemain akan kembali ke menu utama untuk memulai permainan dari *level* 1 lagi.

Diagram *activity level* 3 terdapat pada gambar 3.16.

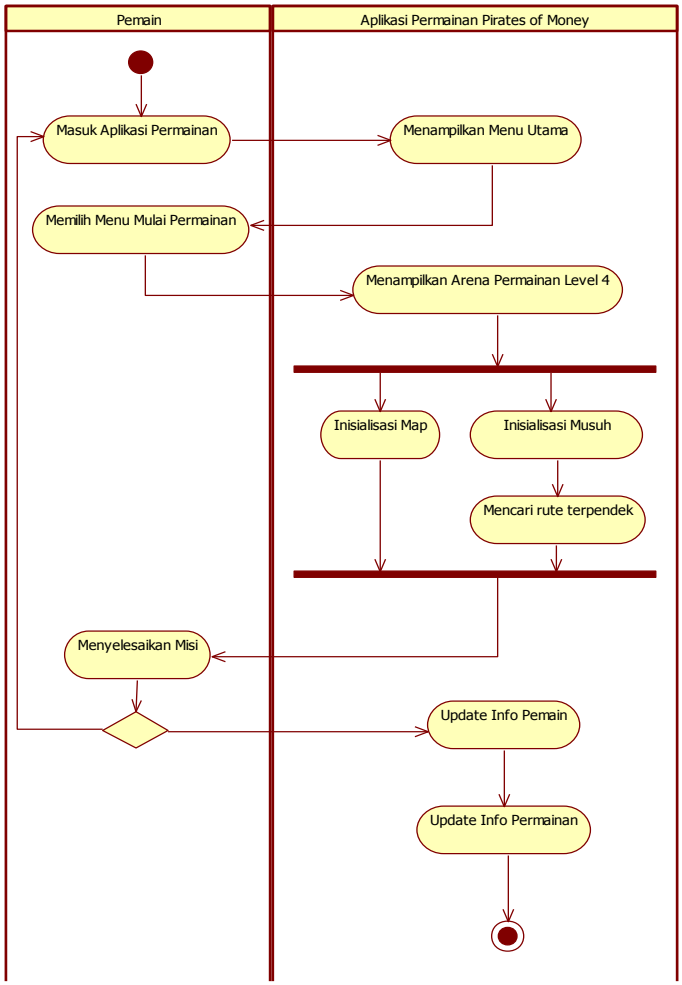


Gambar 3.16 Diagram *Activity Level* 3 *Pirates of Money*

D. Diagram *Activity Level* 4

Proses ini terjadi saat pemain telah menyelesaikan aplikasi permainan *Pirates of Money* pada *level* 3. Sistem akan menampilkan arena permainan *level* 4 dan pemain harus menyelesaikan misi permainan pada *level* 4, apabila misi permainan tidak terselesaikan maka pemain akan kembali ke menu utama untuk memulai permainan dari *level* 1 lagi.

Diagram *activity level* 4 terdapat pada gambar 3.17.

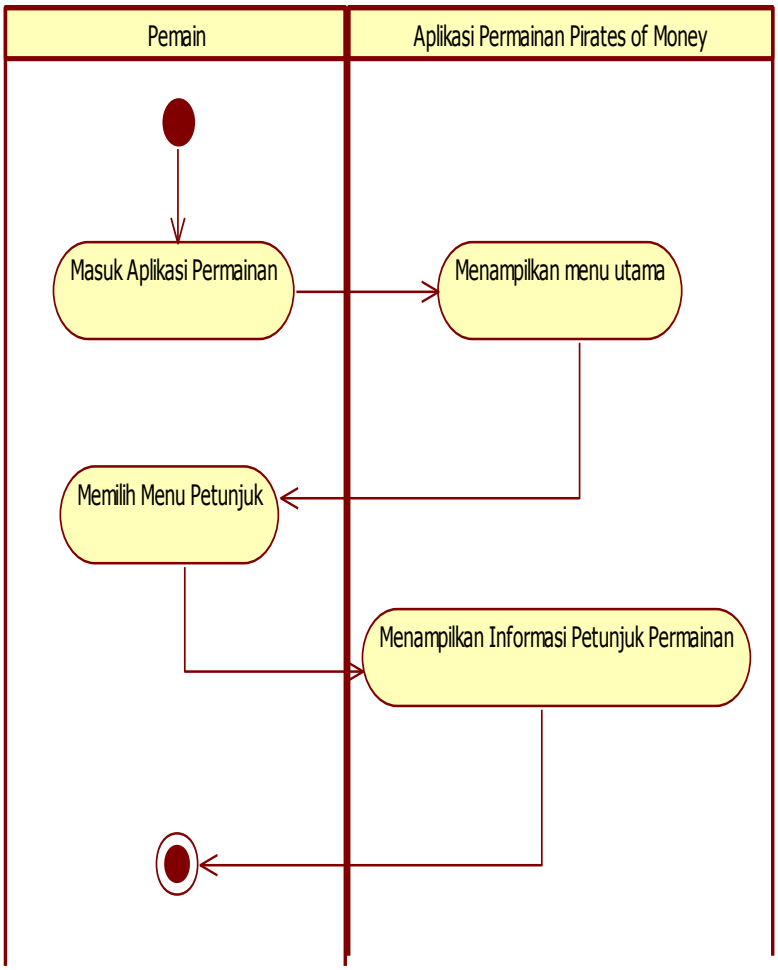


Gambar 3.17 Diagram *Activity Level* 4 *Pirates of Money*

E. Diagram *Activity* Tampil Petunjuk

Proses ini terjadi saat pemain memilih menu petunjuk pada menu utama aplikasi permainan *Pirates of Money*. Sistem akan menampilkan petunjuk permainan dari aplikasi permainan *Pirates of Money*.

Diagram *activity* tampil petunjuk terdapat pada gambar 3.18.

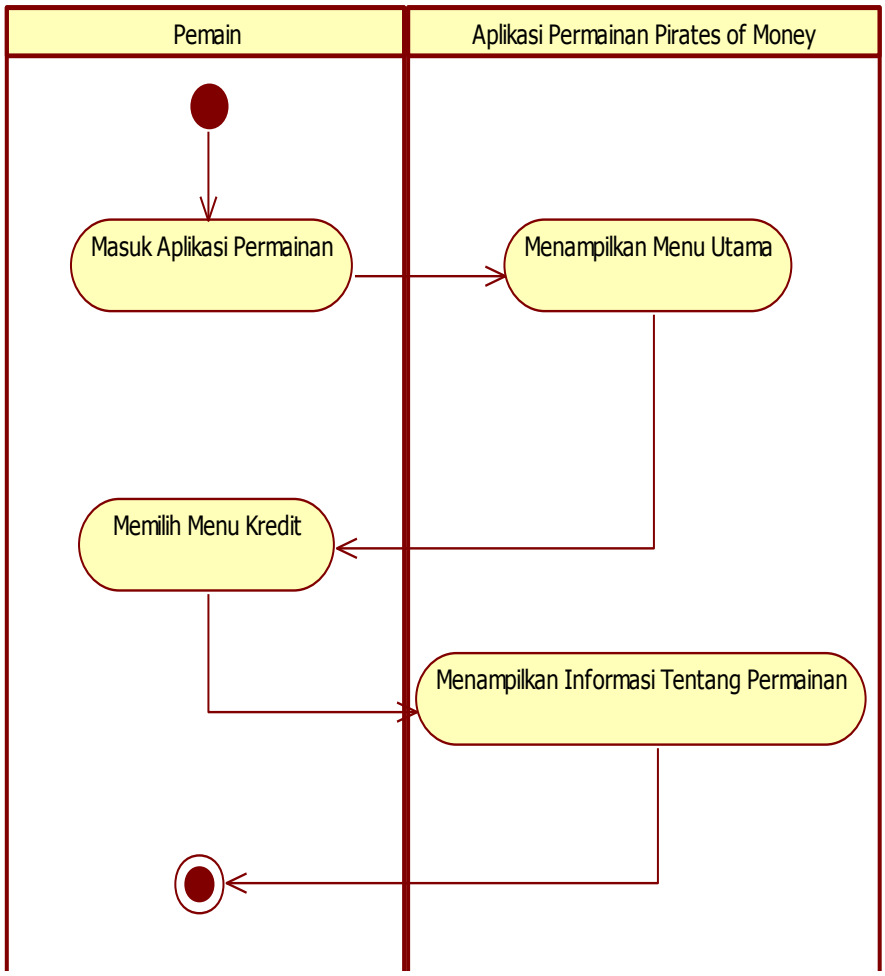


Gambar 3.18 Diagram *Activity* Tampil Petunjuk *Pirates of Money*

F. Diagram *Activity* Tampil Kredit

Proses ini terjadi saat pemain memilih menu kredit pada menu utama aplikasi permainan *Pirates of Money*. Sistem akan menampilkan kredit permainan dari aplikasi permainan *Pirates of Money*.

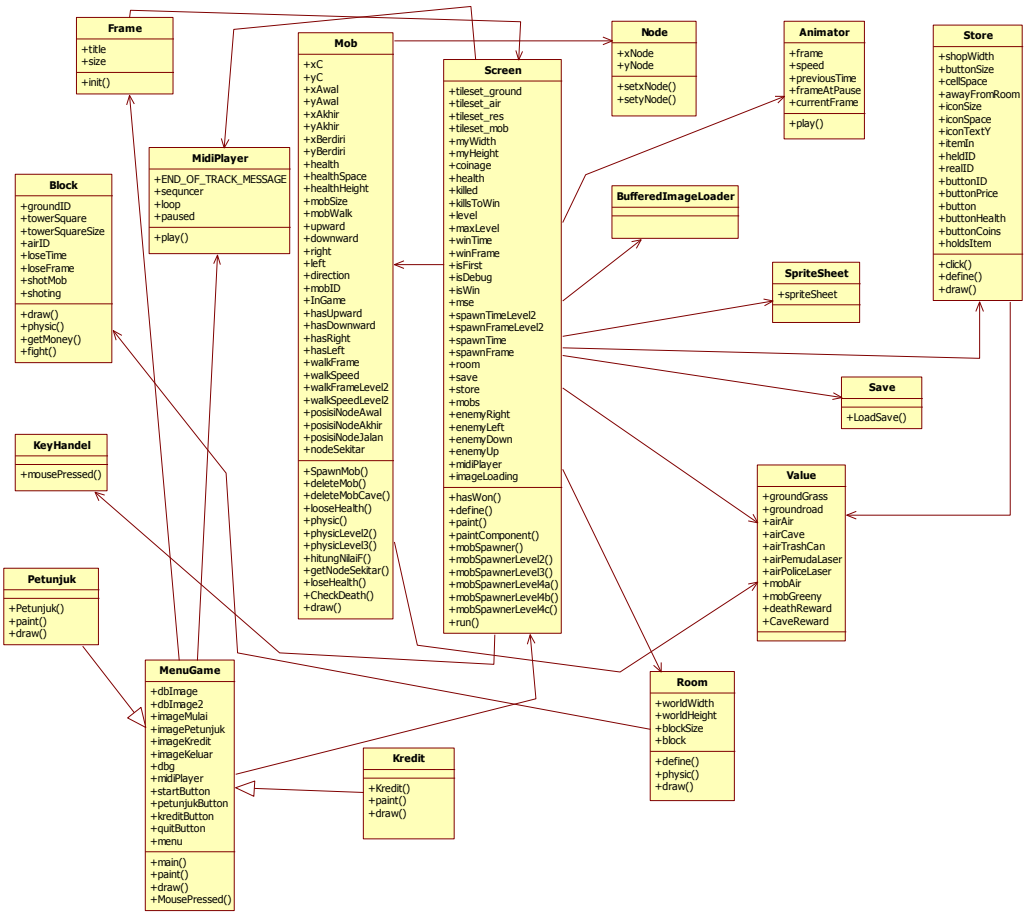
Diagram *activity* tampil kredit terdapat pada gambar 3.19.



Gambar 3.19 Diagram *Activity* Tampil Kredit *Pirates of Money*

3.1.8.4 Diagram *Class*

Diagram *class* pada aplikasi permainan *pirates of money* terdapat pada gambar 3.20.



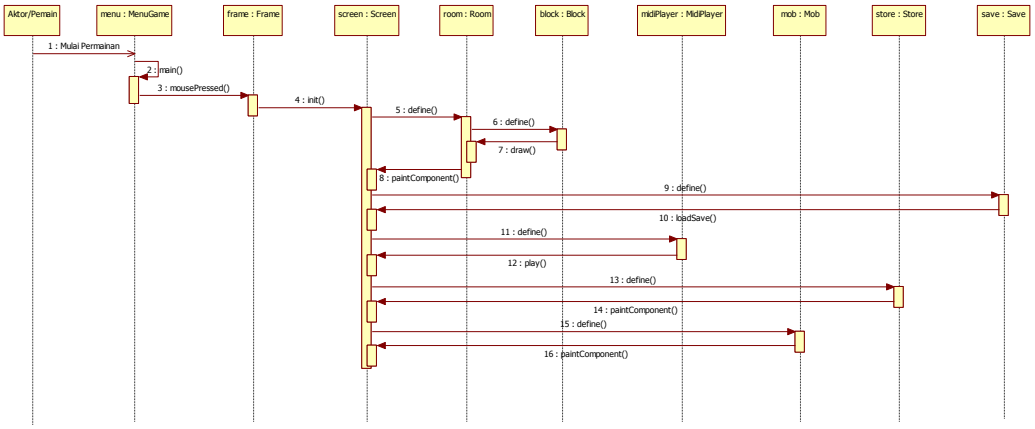
Gambar 3.20 Diagram *Class Pirates of Money*

3.1.8.5 Diagram *Sequence*

Berikut merupakan diagram *sequence* yang terdapat pada aplikasi permainan *pirates of money* berdasarkan *use case* yang ada.

Diagram *sequence* mulai permainan *level* 1 terdapat pada gambar 3.21.

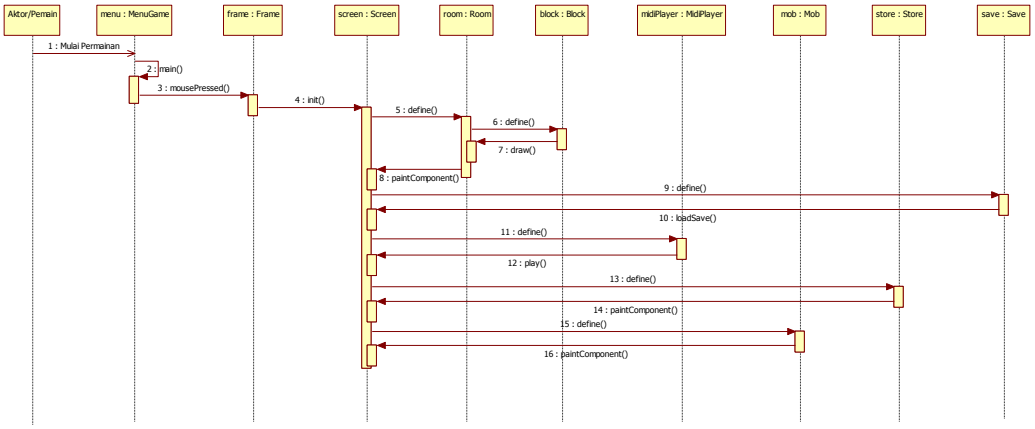
A. Diagram *Sequence* dari *Use Case* : Mulai Permainan *Level* 1



Gambar 3.21 Diagram *Sequence* untuk *Use Case* Mulai Permainan *Level* 1

B. Diagram *Sequence* dari *Use Case* : *Level* 2

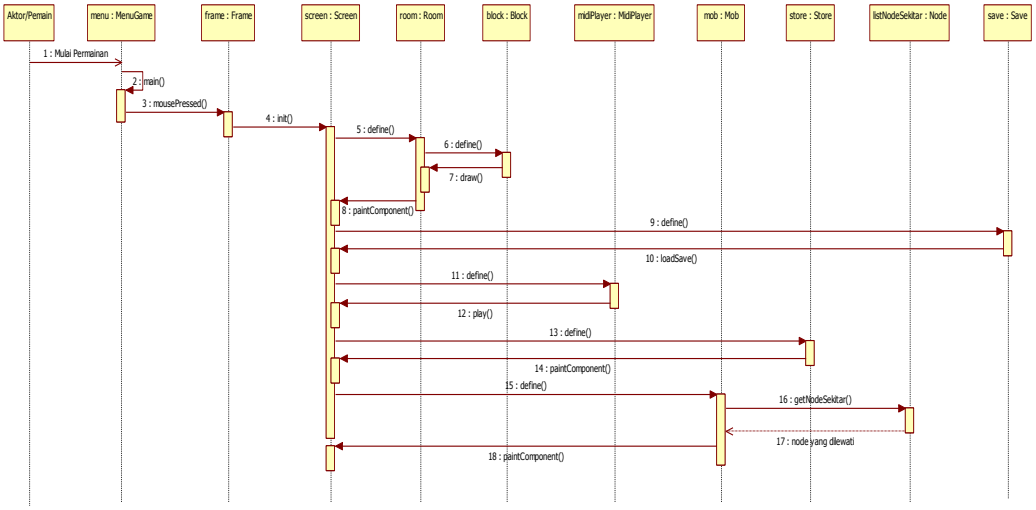
Diagram *sequence level* 2 terdapat pada gambar 3.22.



Gambar 3.22 Diagram *Sequence* untuk *Use Case Level* 2

C. Diagram *Sequence* dari *Use Case* : *Level* 3

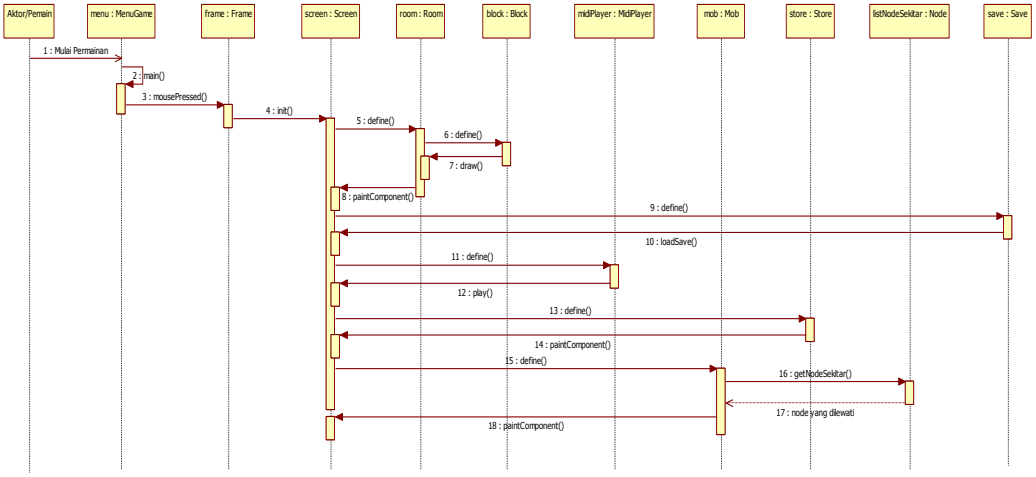
Diagram *sequence level* 3 terdapat pada gambar 3.23.



Gambar 3.23 Diagram *Sequence* untuk *Use Case Level* 3

D. Diagram *Sequence* dari *Use Case* : *Level* 4

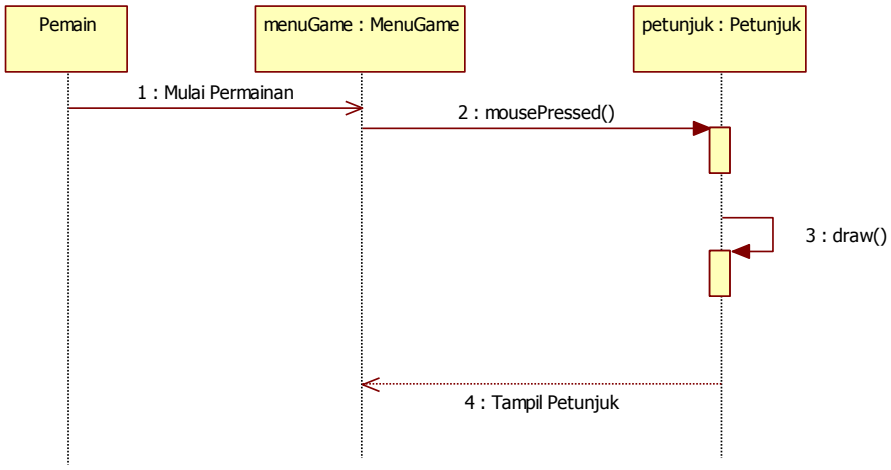
Diagram *sequence level* 4 terdapat pada gambar 3.24.



Gambar 3.24 Diagram *Sequence* untuk *Use Case Level* 4

E. Diagram *Sequence* dari *Use Case* : Tampil Petunjuk

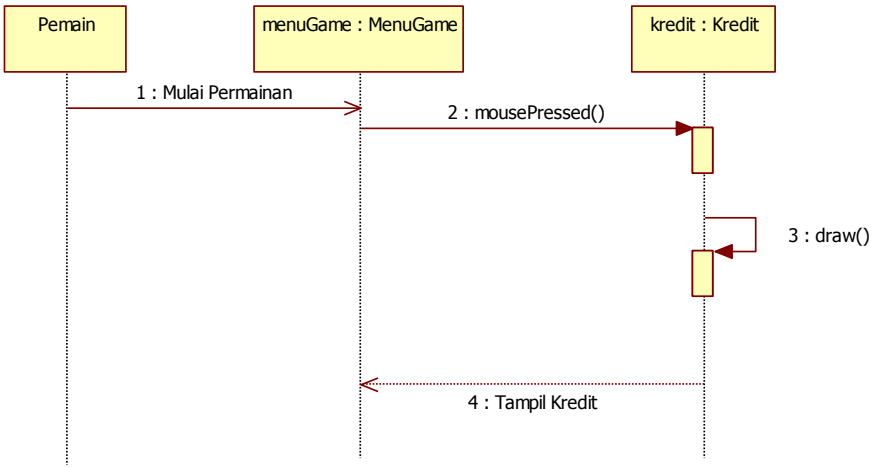
Diagram *sequence* tampil petunjuk terdapat pada gambar 3.25.



Gambar 3.25 Diagram *Sequence* untuk *Use Case* Tampil Petunjuk

F. Diagram *Sequence* dari *Use Case* : Tampil Kredit

Diagram *sequence* tampil kredit terdapat pada gambar 3.26.



Gambar 3.26 Diagram *Sequence* untuk *Use Case* Tampil Kredit

3.1.8.6 Analisis Kebutuhan Non-Fungsional

A. Analisis Perangkat Keras

Analisis perangkat keras digunakan untuk mengetahui spesifikasi perangkat keras dalam menjalankan aplikasi permainan *pirates of money*, seperti pada tabel 3.14.

Tabel 3.14 Spesifikasi Perangkat Keras

Nama Perangkat	Spesifikasi
Prosesor	Dual Core
RAM	1024MB
Harddisk	40G
Display	Resolusi 800 x 600
Keyboard	Standar
Mouse	Standar

B. Analisis Perangkat Lunak

Analisis perangkat lunak digunakan untuk mengetahui spesifikasi perangkat keras dalam menjalankan aplikasi permainan *pirates of money*, seperti pada tabel 3.15.

Tabel 3.15 Spesifikasi Perangkat Lunak Pengguna

Nama Perangkat Lunak	Spesifikasi
Sistem Operasi	Microsoft Windows XP, Linux
Tools	JDK 6

C. Analisis Pengguna

Analisis pengguna digunakan untuk mengetahui spesifikasi pengguna untuk dapat memainkan aplikasi permainan *pirates of money*, seperti pada tabel 3.16.

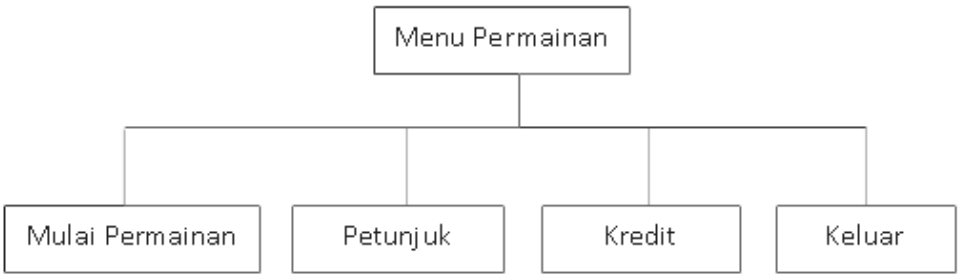
Tabel 3.16 Spesifikasi Pengguna

Peran	Pemain
Usia	7 Tahun ke atas
Tingkat Keterampilan	Dapat menggunakan komputer
Pengalaman	Pernah memainkan aplikasi permainan lainnya seperti *garden defense, warcraft,* dan lain-lain.

3.2 Perancangan Sistem

3.2.1 Perancangan Arsitektur Menu

Untuk mempermudah dalam pembuatan sistem, 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 *Pirates of Money* dapat dilihat pada gambar 3.27.



Gambar 3.27 Arsitektur Menu *Pirates of Money*

A. Mulai Permainan : merupakan submenu untuk memulai permainan baru

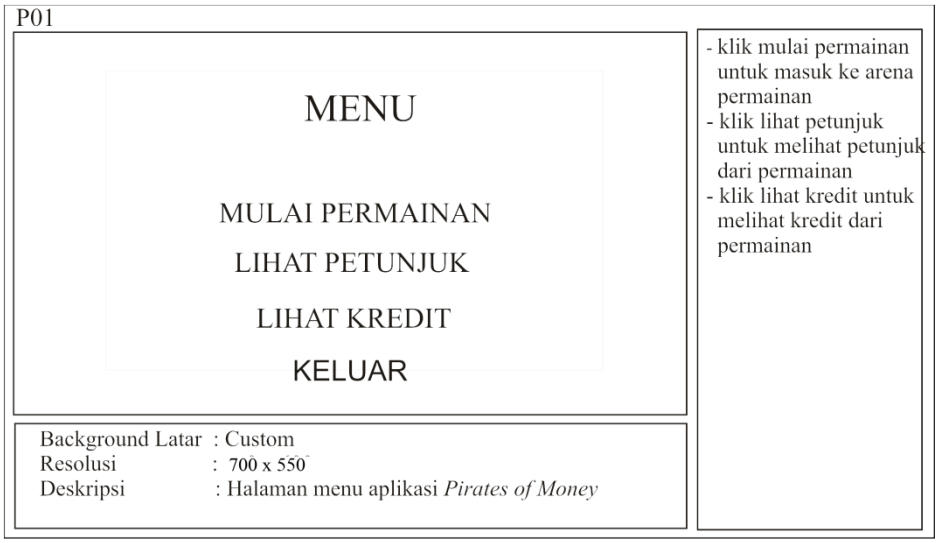
B. Petunjuk : merupakan submenu untuk menampilkan informasi tentang cara memainkan aplikasi permainan *pirates of money*

C. Kredit : merupakan submenu untuk menampilkan informasi mengenai pembuat aplikasi permainan *pirates of money*

D. Keluar : merupakan submenu untuk keluar dari aplikasi permainan *pirates of money*

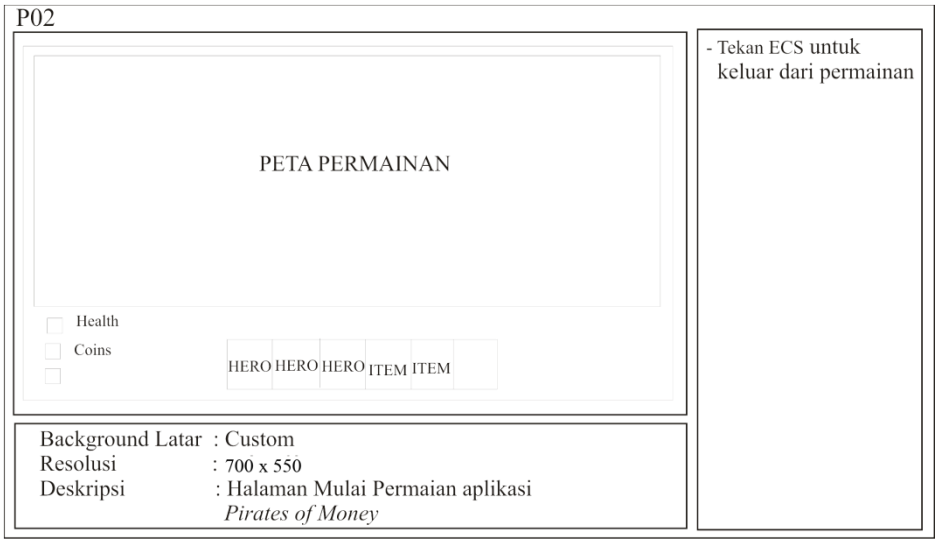
3.2.2 Perancangan Antarmuka *Pirates of Money*

Perancangan antarmuka menu pada aplikasi permainan *pirates of money* dapat dilihat pada gambar 3.28.



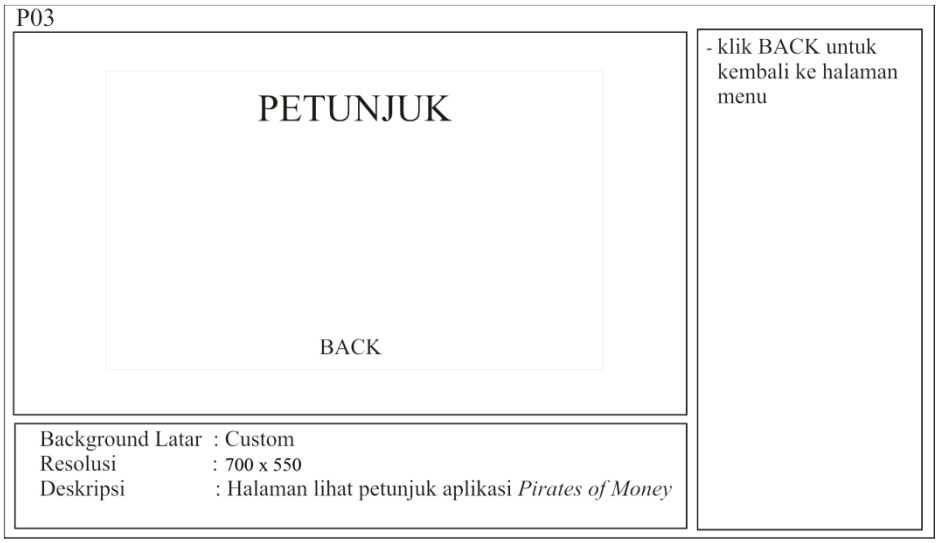
Gambar 3.28 Perancangan Antarmuka Menu *Pirates of Money*

Perancangan antarmuka mulai permainan pada aplikasi permainan *pirates of money* dapat dilihat pada gambar 3.29.



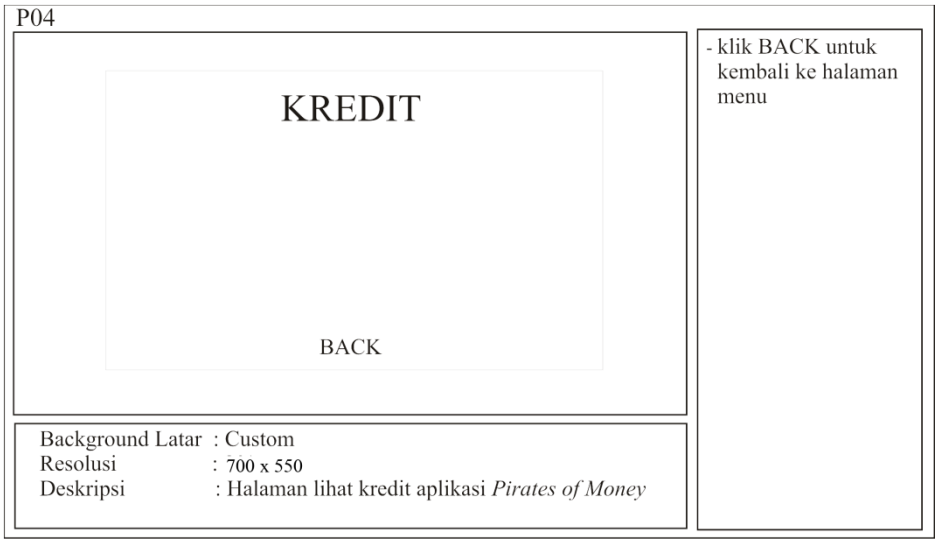
Gambar 3.29 Perancangan Antarmuka Mulai Permainan *Pirates of Money*

Perancangan antarmuka lihat petunjuk pada aplikasi permainan *pirates of money* dapat dilihat pada gambar 3.30.



Gambar 3.30 Perancangan Antarmuka Lihat Petunjuk *Pirates of Money*

Perancangan antarmuka lihat kredit pada aplikasi permainan *pirates of money* dapat dilihat pada gambar 3.31.

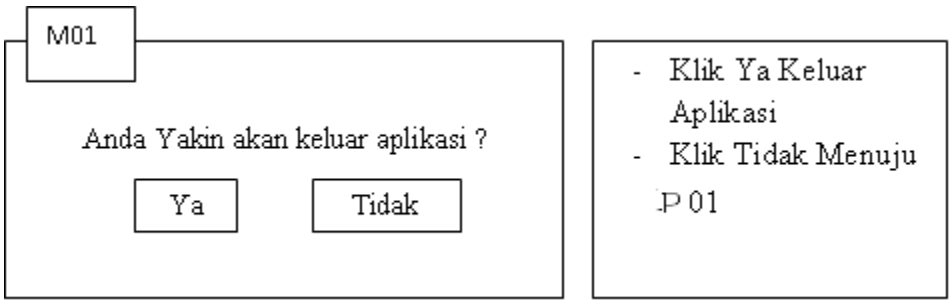


Gambar 3.31 Perancangan Antarmuka Lihat Kredit *Pirates of Money*

3.2.3 Perancangan Pesan

Pesan yang dirancang dalam konteks aplikasi yang dikembangkan merupakan pesan peringatan yang keluar jika pengguna melakukan aktivitas tertentu yang akan memicu event tertentu.

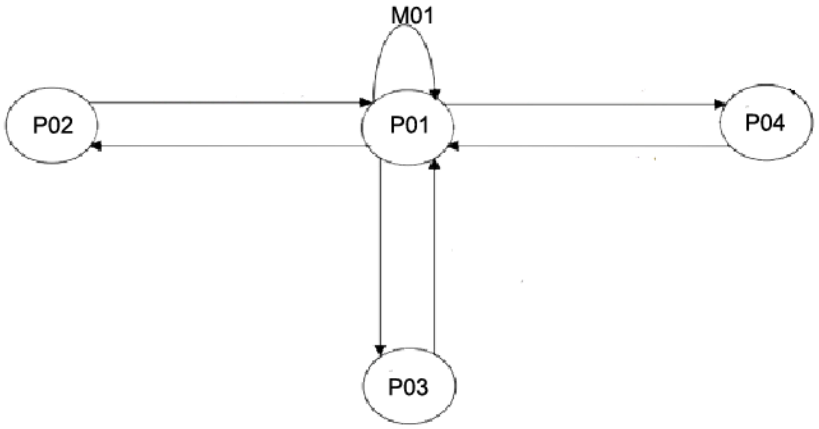
Perancangan pesan peringatan keluar aplikasi pada aplikasi permainan *pirates of money* dapat dilihat pada gambar 3.32.



Gambar 3.32 Perancangan Pesan Peringatan Keluar Aplikasi

3.2.4 Jaringan Semantik

Jaringan semantik pada aplikasi permainan *pirates of money* dapat dilihat pada gambar 3.33.



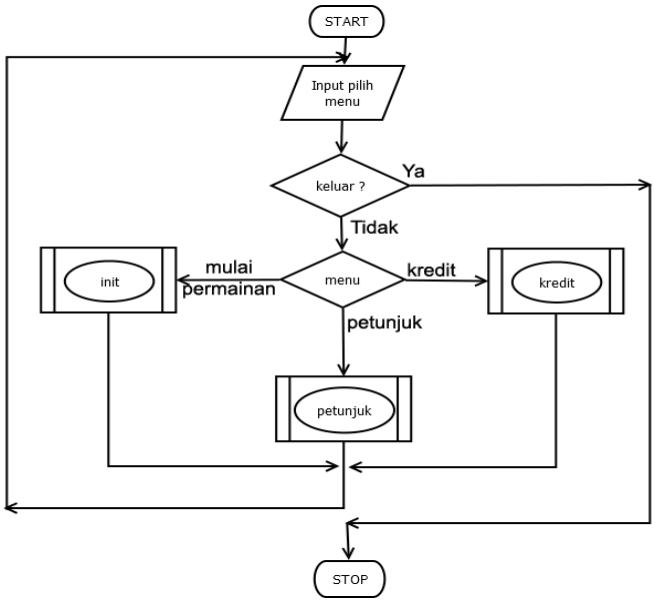
Gambar 3.33 Jaringan Semantik *Pirates of Money*

3.2.5 Perancangan Method

Perancangan komponen method merupakan perancangan yang dibuat setelah perancangan arsitektur menu, antarmuka, pesan, dan jaringan semantik. Perancangan ini berfungsi untuk mendeskripsikan method-method yang berada di dalam aplikasi permainan *pirates of money*. Adapun method-method yang terdapat dalam aplikasi permainan *pirates of money* yang akan dibangun adalah sebagai berikut :

A. Method Main()

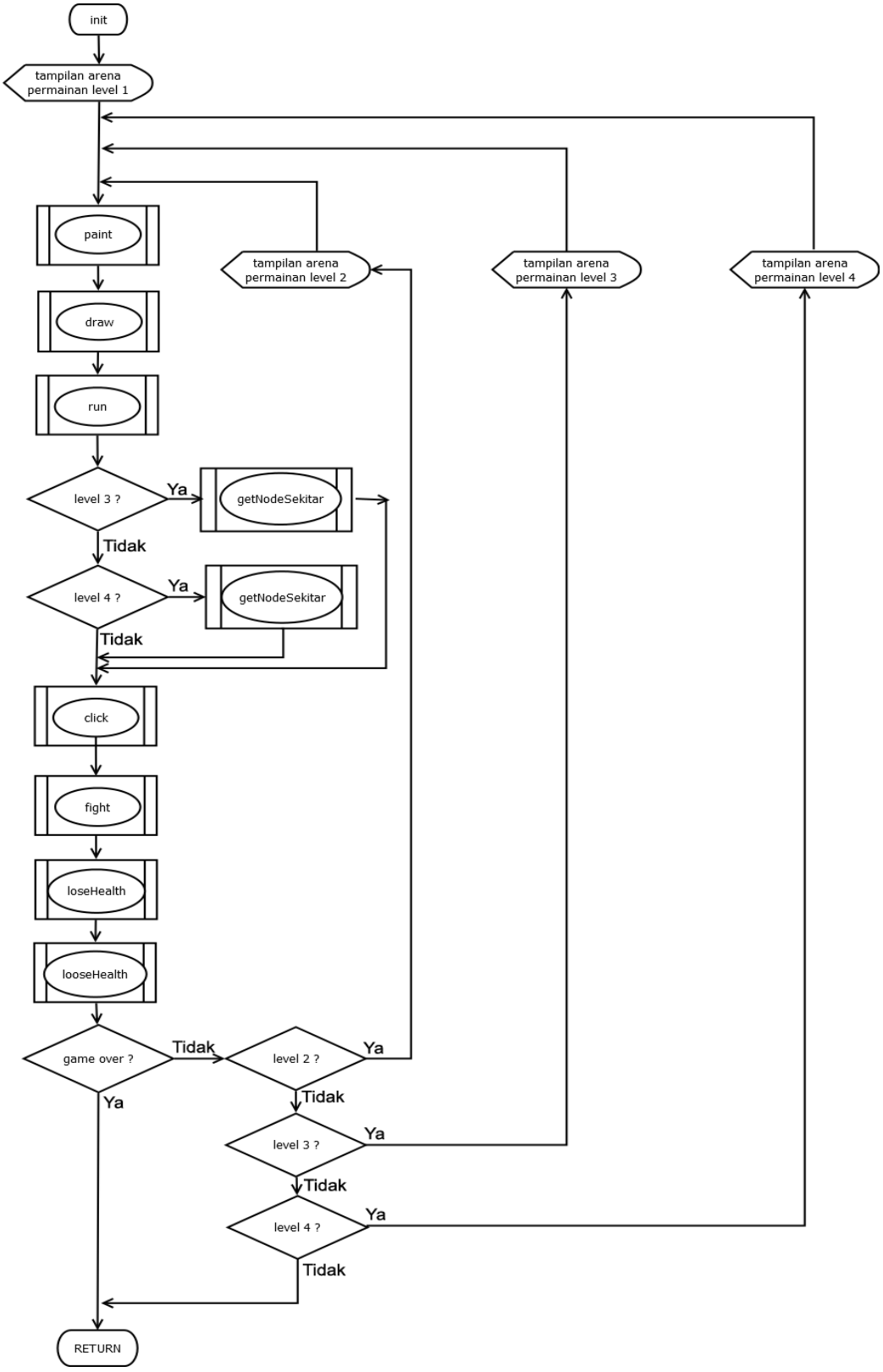
Method main() akan dipanggil ketika memulai permainan, dapat dilihat pada gambar 3.34.



Gambar 3.34 *Flowchart* Method Main()

B. Method Init()

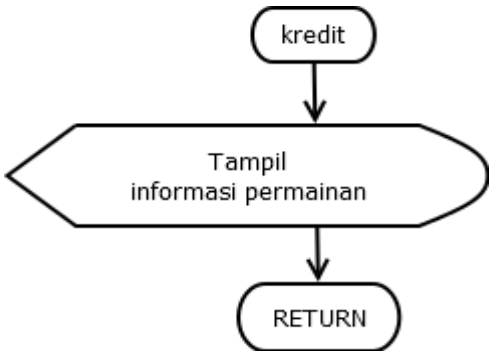
Method init() akan dipanggil ketika pemain memilih menu mulai permainan, dapat dilihat pada gambar 3.35.



Gambar 3.35 *Flowchart* Method Init()

C. Method Kredit()

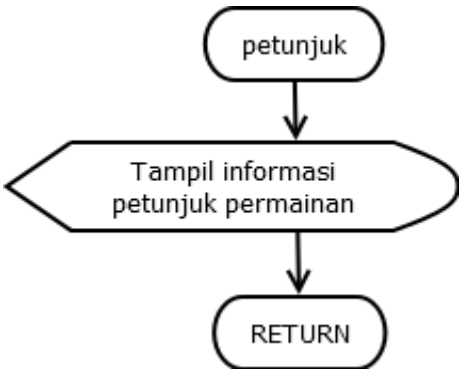
Method Kredit() akan dipanggil ketika memilih menu kredit, dapat dilihat pada gambar 3.36.



Gambar 3.36 *Flowchart* Method Kredit()

D. Method Petunjuk()

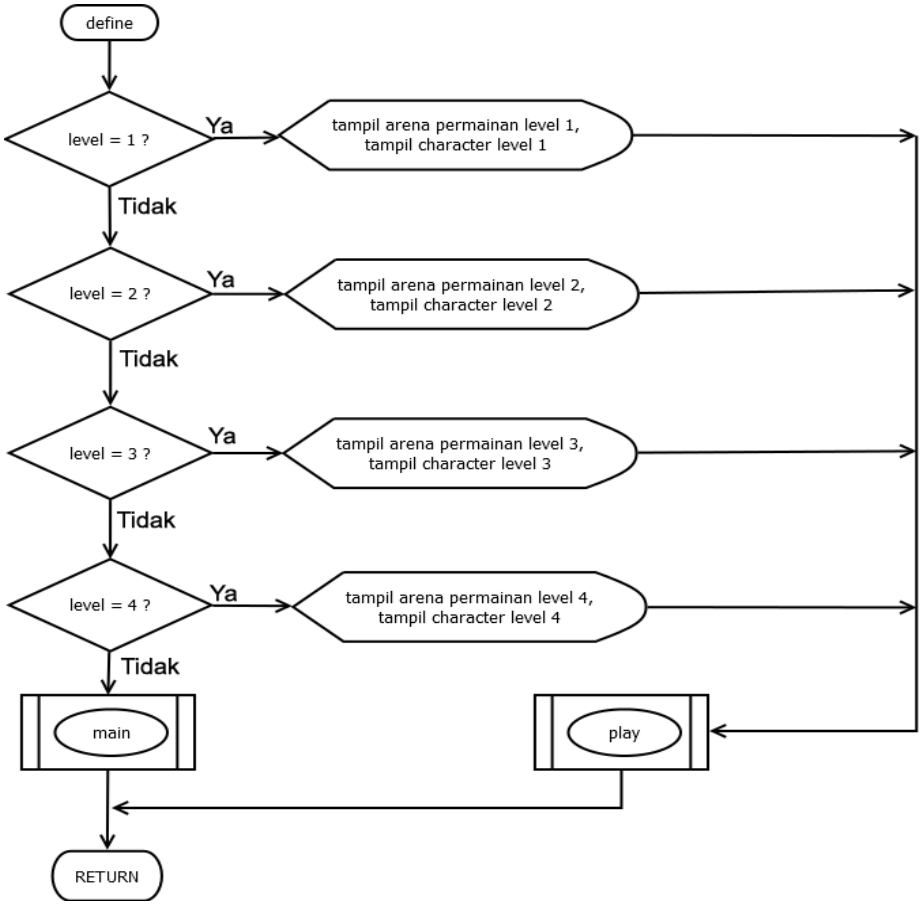
Method petunjuk() akan dipanggil ketika memilih menu petunjuk, dapat dilihat pada gambar 3.37.



Gambar 3.37 *Flowchart* Method Petunjuk()

E. Method Define()

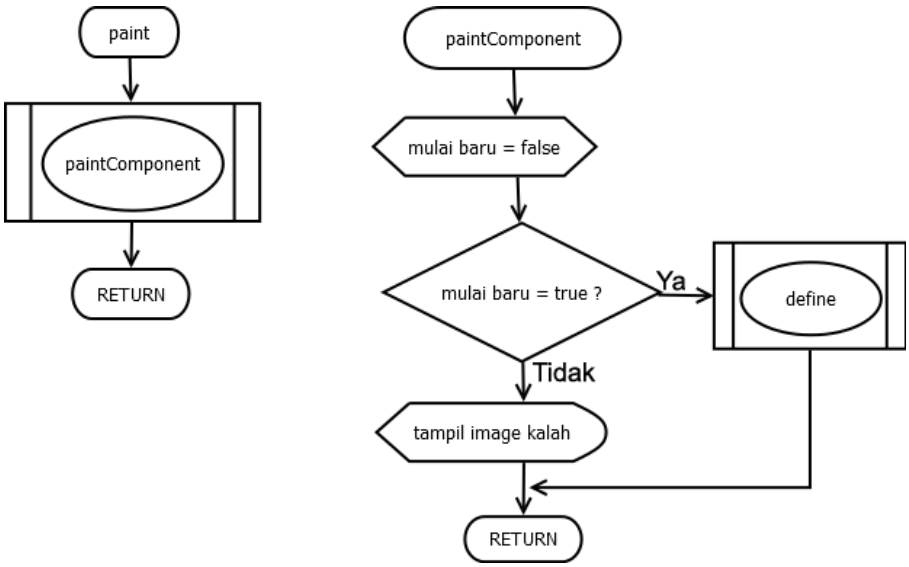
Method define() akan dipanggil di dalam method init(), dapat dilihat pada gambar 3.38 .



Gambar 3.38 *Flowchart* Method Define()

F. Method Paint()

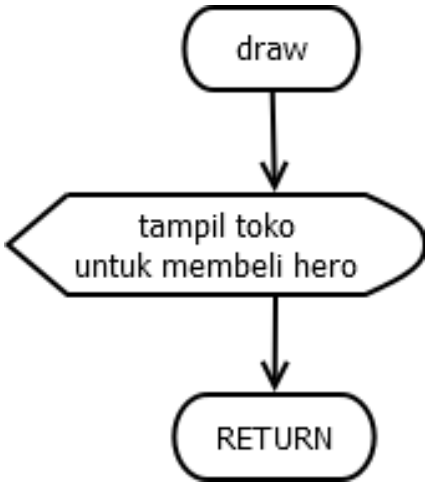
Method paint() akan dipanggil di dalam method init(), dapat dilihat pada gambar 3.39.



Gambar 3.39 *Flowchart* Method Paint()

G. Method draw()

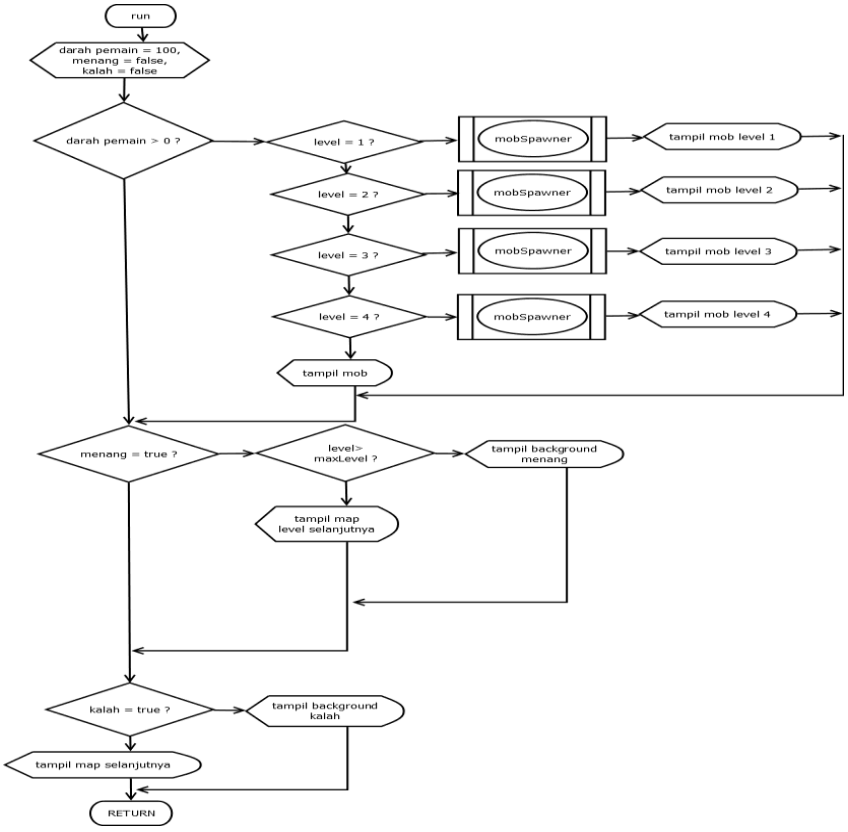
Method draw() akan di dalam method init(), dapat dilihat pada gambar 3.40.



Gambar 3.40 *Flowchart* Method Draw()

H. Method Run()

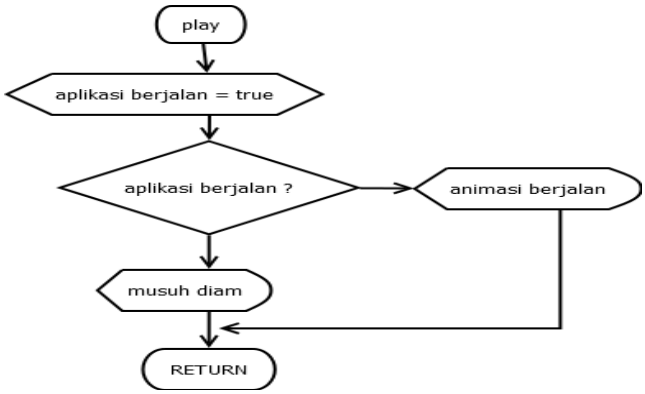
Method run() akan dipanggil di dalam method init(), dapat dilihat pada gambar 3.41.



Gambar 3.41 *Flowchart* Method Run()

I. Method Play()

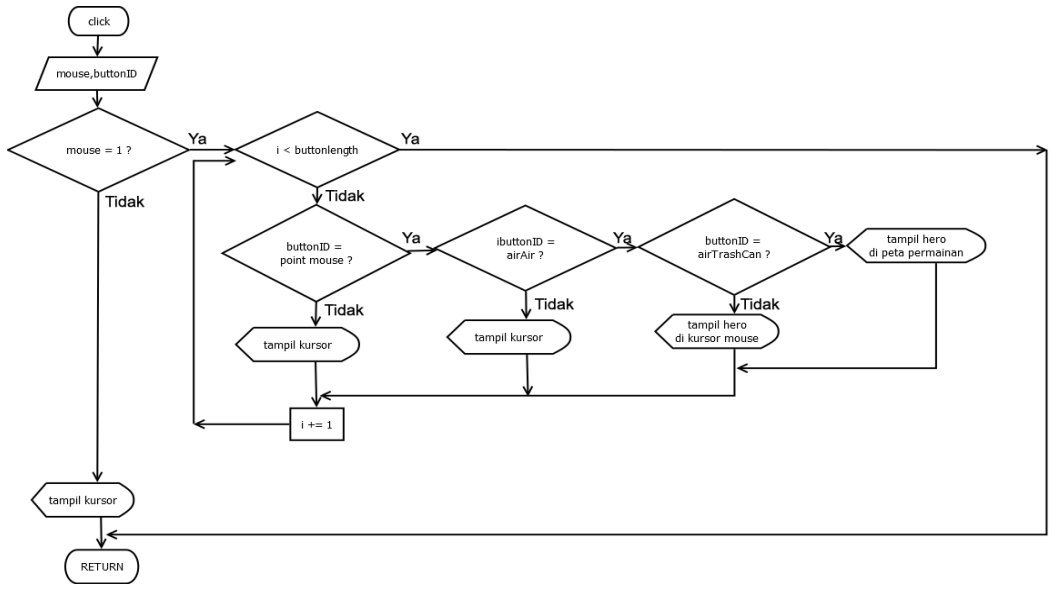
Method play() akan dipanggil pada method init(), dapat dilihat pada gambar 3.42.



Gambar 3.42 *Flowchart* Methode Play()

J. Method Click()

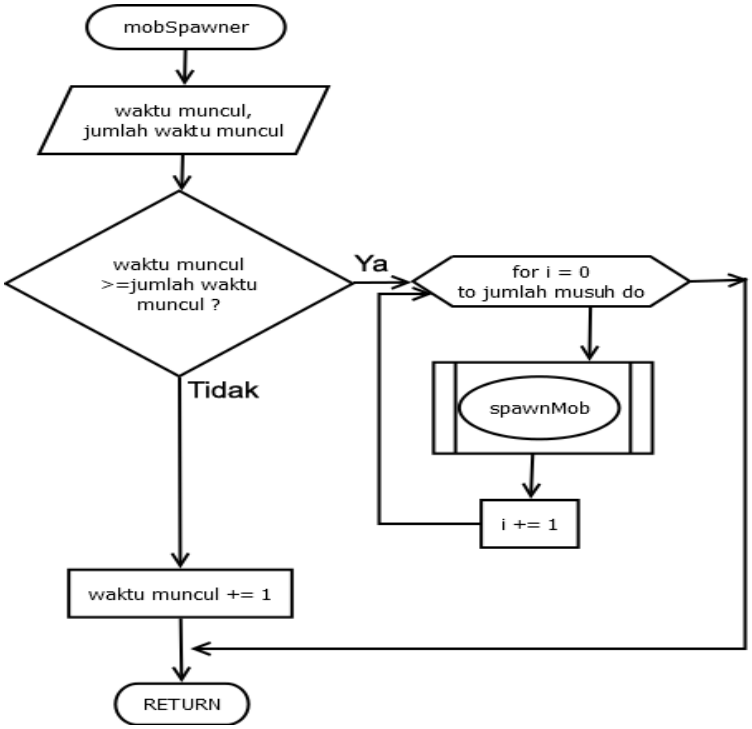
Method click() akan dipanggil di dalam method init(), dapat dilihat pada gambar 3.43.



Gambar 3.43 *Flowchart* Method Click()

K. Method mobSpawner()

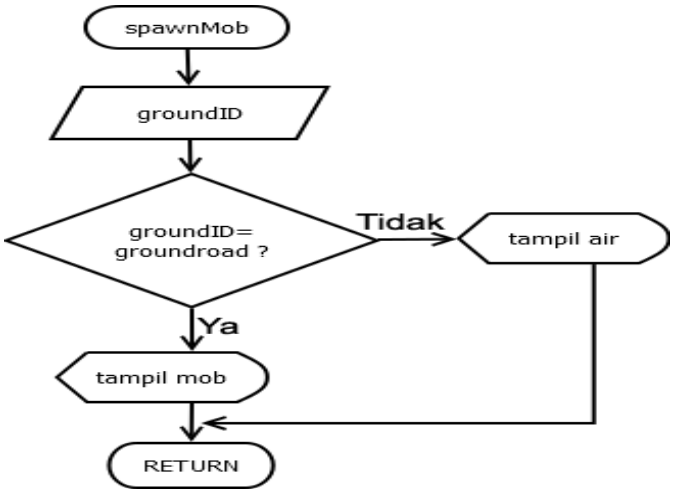
Method mobSpawner() akan dipanggil di dalam method init(), dapat dilihat pada gambar 3.44.



Gambar 3.44 *Flowchart* Method mobSpawner()

L. Method spawnMob()

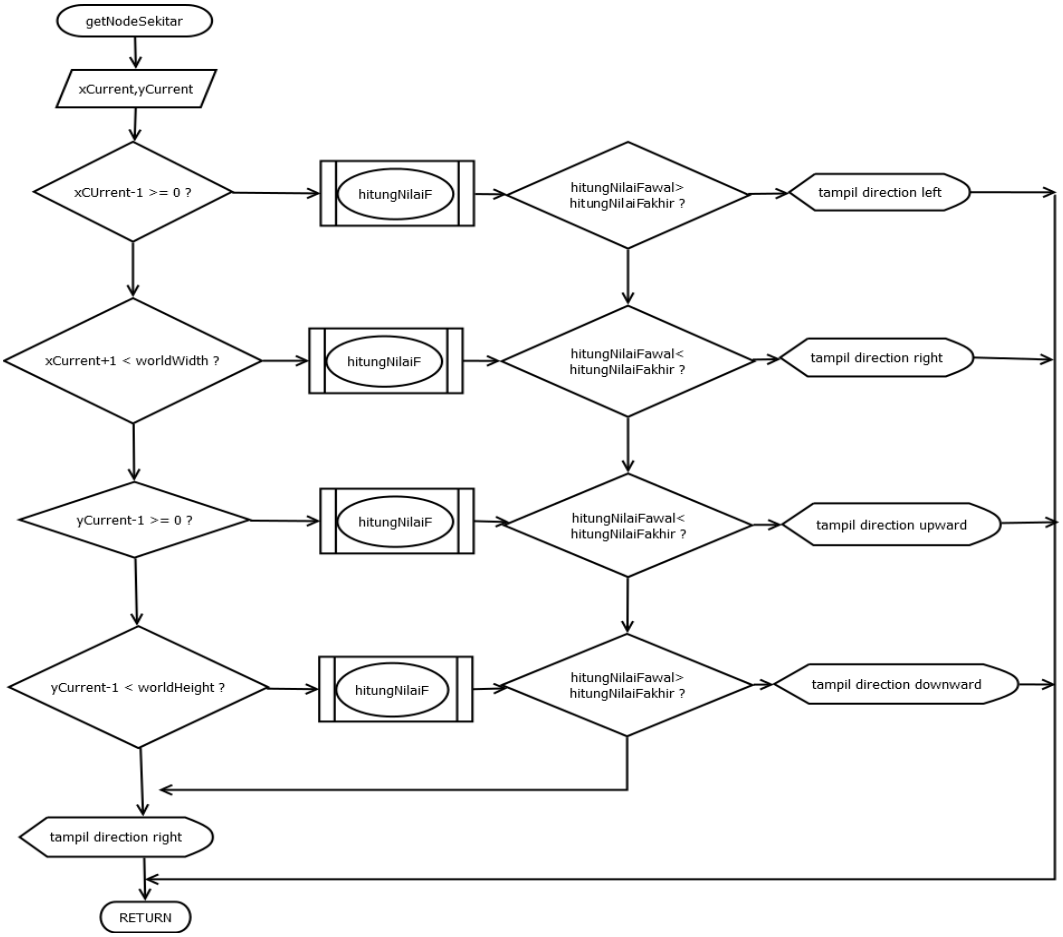
Method spawnMob() akan dipanggil di dalam method mobSpawner(), dapat dilihat pada gambar 3.45.



Gambar 3.45 *Flowchart* Method spawnMob()

M. Method getNodeSekitar()

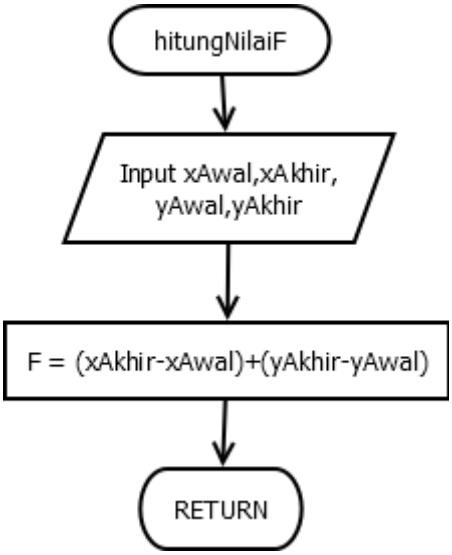
Method getNodeSekitar() akan dipanggil pada level 3 dan 4 di dalam method init(), dapat dilihat pada gambar 3.46.



Gambar 3.46 *Flowchart* Methode getNodeSekitar()

N. Method hitungNilaiF()

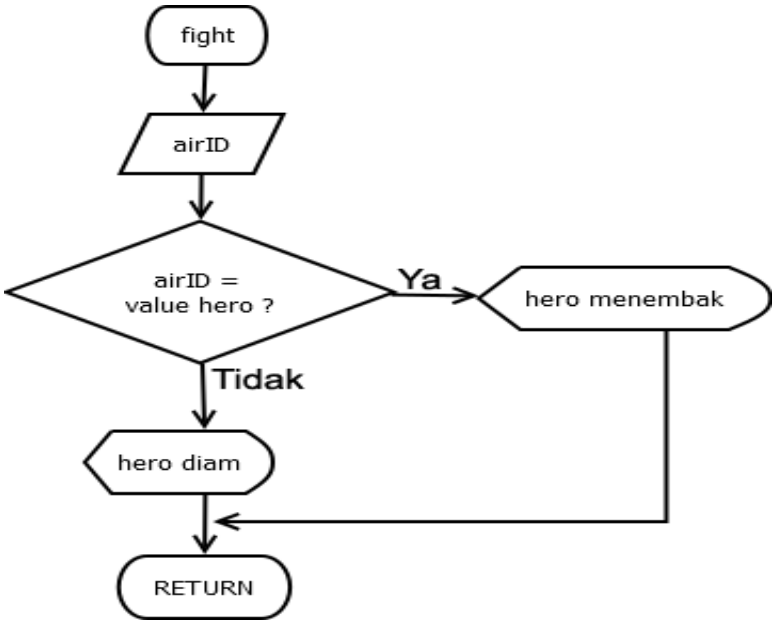
Method hitungNilaiF() akan dipanggil di dalam method init(), dapat dilihat pada gambar 3.47.



Gambar 3.47 *Flowchart* Methode hitungNilaiF()

O. Method Fight()

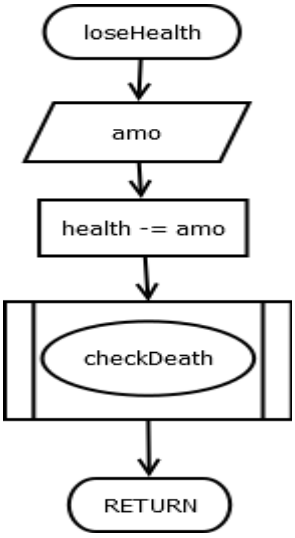
Method fight() akan dipanggil ketika musuh masuk ke dalam jangkauan tembak *hero*, dapat dilihat pada gambar 3.48.



Gambar 3.48 *Flowchart* Method Fight()

P. Method loseHealth()

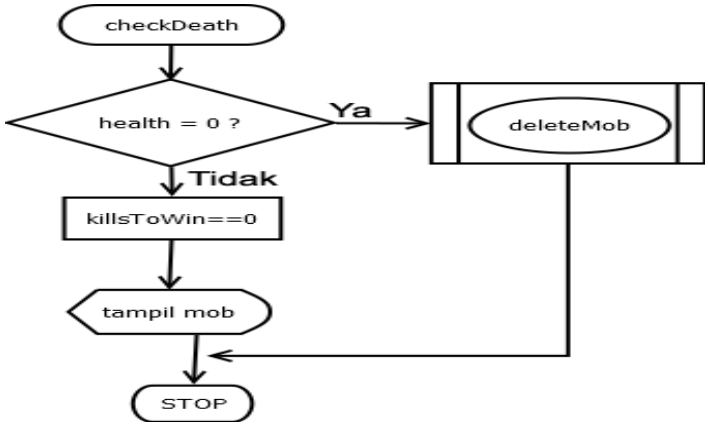
Method loseHealth() dipanggil di dalam method init(), dapat dilihat pada gambar 3.49.



Gambar 3.49 *Flowchart* Method loseHealth()

Q. Method checkDeath()

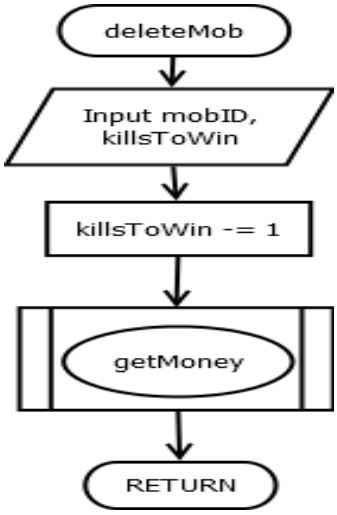
Method checkDeath() akan dipanggil di dalam method loseHealth(), dapat dilihat pada gambar 3.50.



Gambar 3.50 *Flowchart* Method checkDeath()

R. Method deleteMob()

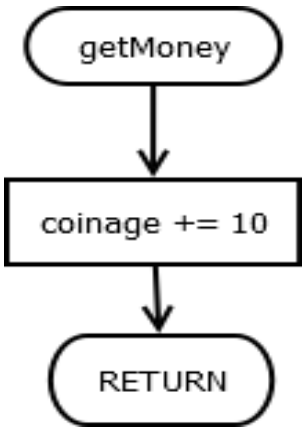
Method deleteMob() akan dipanggil di dalam method checkDeath(), dapat dilihat pada gambar 3.51.



Gambar 3.51 *Flowchart* Method deleteMob()

S. Method getMoney()

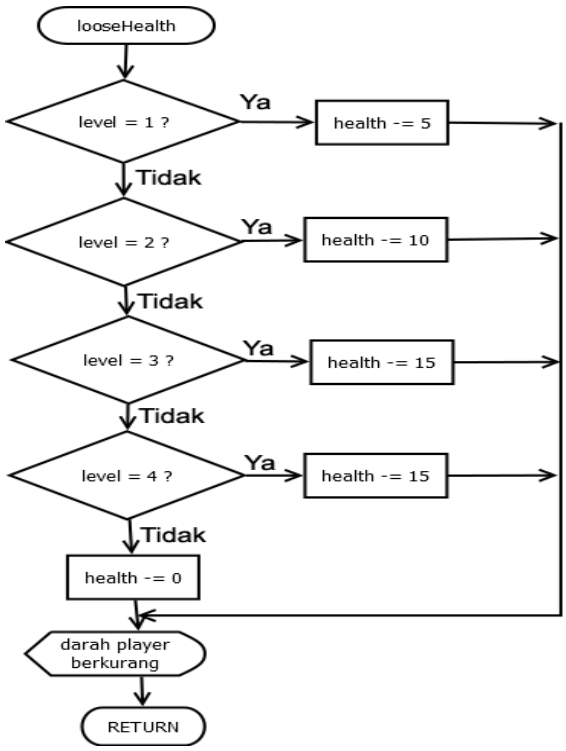
Method getMoney() akan dipanggil di dalam method deleteMob(), dapat dilihat pada gambar 3.52.



Gambar 3.52 *Flowchart* Method getMoney()

T. Method looseHealth()

Method looseHealth() akan dipanggil di dalam method init(), dapat dilihat pada gambar 3.54.



Gambar 3.53 *Flowchart* Method looseHealth()