BAB 3
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 Masalah

Berdasarkan latar belakang yang sudah diuraikan maka masalah yang didapat adalah sebagai berikut :

1. Wayang kurang diminati oleh generasi muda karena cerita yang dirasakan berat, penuh renungan dan filosofis berat dan lebih memilih media yang lebih interaktif dan menghibur.
2. Jalan cerita pada *game* sejenis tidak tersampaikan dengan jelas.
3. *Game*play pada *game* sejenis masih sederhana, pemain hanya bisa menghindari dan menangkis tidak bisa menyerang dan dan tidak ada musuh yang bisa menyerang.

3.1.2 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.

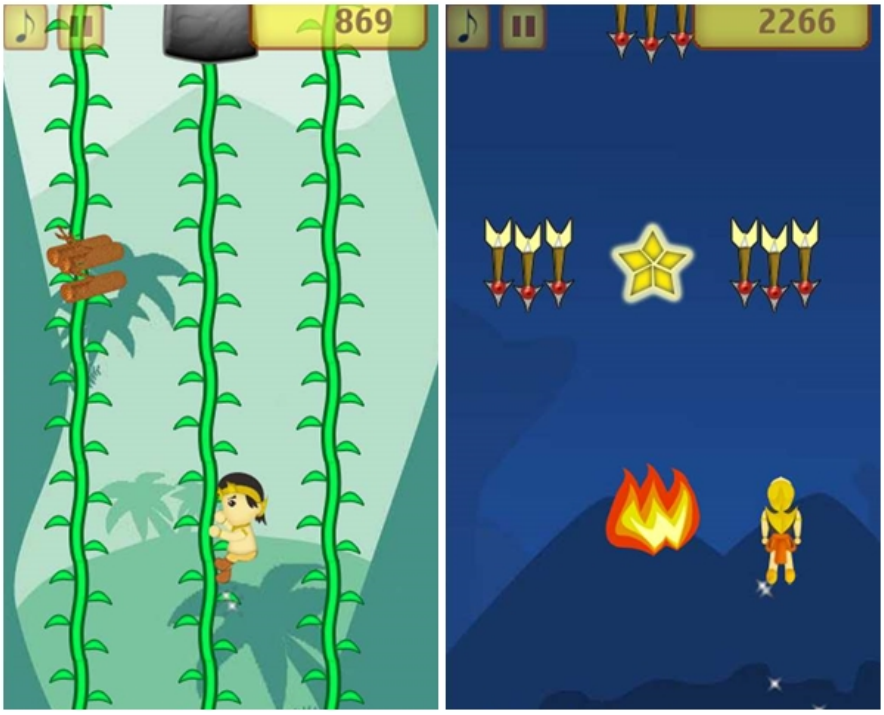
3.1.2.1 *Game The* Wayang

Game The Wayang merupakan *game adventure* berbasis *mobile* yang dikembangkan oleh developer *game* yang bernama *SwiftGame*. *Game* The Wayang ini bergenre *adventure game* dengan permainan yang sangat sederhana, yaitu pemain harus mengendalikan Gatot Kaca menghindari rintangan. Tujuan dari *game* ini adalah menyelesaikan setiap level dari *game*.

1. *Storyline*

Game ini menceritakan tentang Gatot Kaca yang dibuang ke kawah Candradimuka melakukan perjalanan sampai mendapatkan kostum pusaka lalu

melanjutkan lagi perjalannnya sampai Ia menjadi Raja Pringgodani dan akhirnya melawan Pandawa Lima.



Gambar 3-1 Tampilan *Game* The Wayang

2. Antarmuka

Game The Wayang menggunakan bahasa Inggris untuk interaksinya. Pada awal memasuki *game* terdapat 6 menu yaitu, *story mode*, *gallery*, *how to play*, *high score*, *about* dan *exit*, pada tampilan awal juga terdapat tombol *mute. Story mode* merupakan menu untuk memainkan *game The* Wayang. *Gallery* merupakan menu dimana *user* bisa melihat barang pusaka apa saja yang telah Gatot Kaca kumpulkan pada *game*. *How to play* merupakan menu bantuan bagaimana cara memainkan *game. High score* merupakan menu untuk menampilkan skor paling tinggi yang pernah dicapai pemain. *About* adalah menu yang menampilkan informasi tentang developer dari *game The* Wayang. *Exit* adalah menu untuk keluar dari *game* pada halaman muka. Pada tampilan saat *game* dimulai terdapat tombol *pause* untuk menunda permainan atau bisa juga untuk keluar dari *game* ke menu utama dan tombol *mute* untuk mengheningkan musik dari *game*, pada setiap level tombol *pause* dan *mute* tersebut selalu ada. Tampilan *score* saat *game* dimainkan berada di sebelah kanan atas layar.

3. *Gameplay*

Game The Wayang terdiri dari 5 level, tiap level kesulitan yang dihadapi semakin meningkat dan membutuhkan konsentrasi yang kuat untuk memainkannya. Tugas utama pemain pada *game* ini adalah mengendalikan Gatot Kaca untuk menghindari rintangan-rintangan yang ada seperti panah, api, kayu, batu dan batu berapi dan mengambil bintang untuk mendapatkan *score*. Pada level terakhir Gatot Kaca melawan Pandawa lima dengan cara menghindari kelima pandawa. Karakter Gatot Kaca dikendalikan dengan menggunakan *touchscreen* dengan cara men-*tap* layar *touchscreen* sehingga Gatot Kaca akan berpindah posisi menghindari rintangan-rintangan. Pada tiap level kesulitan yang dihadapi pemain semakin meningkat dari jumlah rintangan rintangan yang harus dilawati. Rintangan- rintangan yang dilewati pada level 1 adalah batu berapi, level 2 adalah batu, level 3 adalah baja, level 4 adalah kayu-kayu dan level terakir adalah panah dan api dan pandawa lima. Jika Gatot Kaca mengenai rintangan-rintangan maka Gatot Kaca akan mati dan kembali ke awal permainan yang sedang dijalani dan jika Gatot Kaca mati di pertengahan level lalu keluar dan pemain kembali lagi untuk memainkannya, pemain bisa mulai dari level terakhir yang pernah dimainkannya. Gatot kaca akan berpindah ke level selanjutnya setelah gatot kaca mendapatkan barang pusaka yang ada dalam permainan.

4. *Scoring*

Skor pada *game* The Wayang didapat setiap Gatot Kaca bergerak dan saat mengambil bintang. Skor akan terus bertambah setiap Gatot Kaca bergerak naik, dan pada saat mengambil bintang skor akan bertambah 500 poin. Skor pada *game* The Wayang diakumulasikan pada tiap levelnya.

3.2.1.2 *Game* Tetuko

Tetuko adalah sebuah *game* Android yang pemeran utamanya adalah Tetuko, nama dari Gatot Kaca muda. Tantangan dari *game* ini adalah menggerakkan Tetuko agar bisa selamat dari berbagai serangan dari arah-arah yang berlainan. *Game* Tetuko bergenre *adventure* karena pemain harus menghadapi rintangan-rintangan yang ada.

1. Storyline

Tetuko merupakan Gatot Kaca muda dari cerita wayang, ksatria yang gagah dan sangat sakti. Sehingga dia harus waspada dalam menjalani hidupnya karena adanya tantangan-tantangan atau musuh yang menyerang.



Gambar 3-2 Tampilan *Game* Tetuko

2. Antarmuka

Game Tetuko menggunakan bahasa Inggris untuk interaksinya. Pada awal memasuki *game* terdapat 3 menu yaitu, *play, option* dan *quit. Play* merupakan menu untuk memainkan *game* Tetuko. *Option* merupakan menu dimana *user* bisa mengatur *sound* dan *music*, user juga dapat melihat *credit* dari *game* dan dapat kembali ke menu utama. *Quit* adalah menu untuk keluar dari *game* pada halaman muka. Pada tampilan saat *game* dimulai terdapat tombol *pause* untuk menunda permainan atau bisa juga untuk mengatur sound mngatur musik, kembali ke menu utama dan keluar dari *game*, pada setiap level tombol *pause* tersebut selalu ada. Tampilan score saat *game* dimainkan berada di bagian tengah atas layar. Dan tampilan life dari tetuko berada di bagian kiri atas layar.

3. *Game*play

Game teuko memiliki level yang tak terbatas. Pemain mengendalikan karakter dengan tombol yang berada di sebelah kanan dan kiri layar. Tugas pemain adalah menghindari bebagai macam seangan dari arah-arah yang berlainan. Serangan yang terdapat pada *game* adalah serangan batu dan panah.

Jika tetuko terkena batu dan panah maka nyawanya akan hilang satu. Tetuko juga akan mendaatkan nyaa selain mendapatkan serangan. Jika nyawa tetuko habis maka *game* akan berakhir.

4. Scoring

Skor pada *game* tetuko akan didapat jika tetuko menangkis serangan berupa panah dan batu. Skor yang didapat adalah 10 poin, jika tetuko menghindari seangan makan tidak akan mendapakan skor.

Game Larila Arjuna yang dibangun merupakan *game* 2D yang ber*genre action adventure*. *Game* yang akan dibangun ini mempunyai tema yang sama dengan *game* The Wayang dan Tetuko yaitu tentang cerita wayang. *Game* Larilah Arjuna akan mengangkat cerita perang Baratayuda, yang memiliki *gameplay* berpetualang mengalahkan pihak musuh.

Secara garis besar perbandingan *game* sejenis dengan *game* yang akan dibangun akan dijelaskan dengan **Tabel 3-1**.

Tabel 3-1 Perbandingan *Game* Sejenis

	***Game The* Wayang**	**Tetuko**	***Game* yang akan dibangun**
Genre Game	*Adventure*	*Adventure*	*Action Adventure*
Desain Grafis	2D	2D	2D
Karakter Utama	Gatotkaca	Tetuko/ gatotkaca muda	Arjuna
Pergerakan Animasi	Pergerakan karakter berupa memanjat dan melompat	Pergerakan karakter berupa menunduk menangkis dan melompat	Pergerakan karakter berupa berlari, melompat dan menembak
Enemy	Tidak memiliki karakter musuh yang bisa menyerang	Tidak memiliki karakter musuh	Memiliki karakter musuh yang dapat meyerang dan yang tidak dapat menyerang.
Gameplay	Mendapatkan benda pusaka untuk mencapai akhir dari cerita perjalanan Gatotkaca, setiap levelnya memiliki gamplay yang sama.	Menghindari atau menangkis serangan berupa panah dan batu agar tetuko bisa selamat.	Membunuh musuh demi mengambil kembali tahta yang telah direbut oleh musuh, seiap levelnya memiliki *game*play yang sama.
Level	Memiliki 5 level.	Tidak terhingga	Memiliki 3 level.
Item	Bintang	Panah, batu dan nyawa	Koin perak dan panah
Basis	*Mobile*	*Mobile*	*Mobile*

Dari tabel *game* sejenis diatas akan diambil poin-poin yang akan diperbaiki pada *game* yang akan dibangun dengan acuan sebagai berikut:

1. Jalan cerita yang dipahami oleh pemain dengan adanya intro di setiap level.
2. *Gameplay*, pemain dapat menyerang musuh.
3. Karakter musuh yang lebih bervariasi.
4. AI pada musuh yang tidak hanya mempunyai kemampuan untuk pencarian jalan tetapi juga kemampuan menyerang pemain.

Dalam pembangunan *game* ini terdapat kesamaan atau korelasi antara *game* sejenis dengan *game* yang akan dibangun diantaranya adalah:

1. Memiliki kesamaa dari tema cerita, yaitu wayang.
2. Memilik kesamaan dari karakter pemain.
3. Memiliki kesamaan grafis, yaitu 2D.
4. Memiliki kesamaan dari basis, yaitu *mobile*.

3.1.3 Analisisi *Game* yang Akan Dibangun

Game yang akan dibangun berjudul Larilah Arjuna yang ber-genre *Action Adventure*, dimana memiliki tema yang sama dengan *game The* Wayang dan *game* Tetuko yaitu tentang wayang. *Game* Larilah Arjuna mengangkat kisah Pearang Baratayuda, dimana pandawa lima yang termasuk didalamnya adalah Arjuna akan melakukan perjalanan untuk mengalahkan musuh yaitu korawa yang telah merebut takhta kerajaan dari pihak pandawa yang terjadi dalam perang Baratayuda. *Game* ini dibangun dengan menggunakan karakter Arjuna dalam cerita pewayangan.

Fitur-fitur yang akan dibuat pada *game adventure* larilah Arjuna ini adalah sebagai berikut :

1. Sistem *single player*
2. Karakter utama diambil dari tokoh wayang yaitu Arjuna.
3. Tema cerita dari *game* adalah perang Baratayuda.
4. Karakter utama memiliki kemampuan untuk melewati rintangan dan menyerang musuh yang datang.
5. Memiliki 3 level dengan tingkat kesulitan yang berbeda-beda.

6. Alur permainan *game* adalah sequensial, yaitu pemain harus menyelesaikan terlebih dahulu pemainan yang ada baru dapat bermain permainan selanjutnya.
7. Kontrol karakter menggunakan *touchscreen*.
8. Grafik yang digunakan merupakan grafik 2D.

3.1.3.1 Storyline

Terjadinya perang antar saudara saudara yaitu Pandawa(melambangkan kekuatan baik) dan sepupu mereka Korawa(melambangkan kekuatan jahat) akibat perebutan takhta kerajaan yang seharusnya jatuh ke tangan Puntadewa, malah diberikan pamannya(Destrarestra) ke anaknya yaitu Duryudana, anak sulung dari para Korawa. Padahal Destraresta berjanji akan mengembalikan takhta ke tangan Puntadewa saat mereka sudah cukup dewasa. Di saat mereka sudah mulai berkuasa, datanglah Kurawa dan Sengkuni yang mencurangi mereka dalam permainan judi sehingga para Pandawa pun harus pergi ke hutan sebagai akibat dari kekalahan mereka. Pandawa yang terusir tidak tinggal diam perlahan-lahan mereka membuat sebuah kerajaan yang bisa menandingi Astina, mereka beri nama “Indaprasta”. Karena merasa tertandingi degan adanya kerajaan Indaprasta pihak Korawa mulai meyerang. Balatentara Korawa menyerang laksana gelombang lautan yang menggulung-gulung, sedang pasukan Pandawa menyerang dengan dahsyat seperti senjata yang menusuk langsung ke pusat kematian.

1. Level 1

Intro: Para pandawa pergi ke hutan sebagai akibat dari kekalahan mereka. Balatentara Korawa menyerang laksana gelombang lautan yang menggulung-gulung, sedang pasukan Pandawa salah satunya adalah Arjuna menyerang dengan dahsyat seperti senjata yang menusuk langsung ke pusat kematian.

Latar : Hutan

Misi : Melawan dan membunuh musuh

Rintangan : Prajurit, Prabu Malawapati

Apa yang harus dilakukan : Berlari menelusuri hutan mengejar dan membunuh pihak musuh. Arjuna akan mengalahkan musuh besar dengan Panah Pasopati Arjuna.

Outro: Setelah berasil melawan musuh kemudian Arjuna dan para pandawa menyusun rencana untuk melanjutkan misinya.

2. Level 2

Intro: Setelah melihat komandan pasukannya berguguran kemudian Bisma maju ke medan pertempuran, mendesak maju menggempur lawan. Atas petunjuk Kresna, Pandawa kemudian mengirim Dewi Srikandi untuk maju menghadapi Bisma. Dengan tampilnya prajurit wanita tersebut di medan pertempuran menghadapi Bisma. Bisma merasa bahwa tiba waktunya maut menjemputnya, sesuai dengan kutukan Dewi Amba yang tewas di tangan Bisma. Dalam pertempuran melawan Bisma, Arjuna ikut menyerang bersama Dewi Srikandi.

Latar : Hutan

Misi : Membunuh musuh bersama Dewi Srikandi

Rintangan : Prajurit, Bisma

Apa yang harus dilakukan: Arjuna berlari menelusuri hutan diikuti oleh Srikandi untuk mengalahkan Bisma. Setelah bertemu Bisma Srikandi bertarung dengan bisma dibantu dengan Arjuna mengunakan panah Psasopati Arjuna sampai akhirnya Bisma mati.

Outro: Arjuna dan Srikandi berhasil membunuh Bisma kemudian mendapat kabar bahwa putranya, Abimanyu dibunuh dengan tidak mematuhi aturan perang yang berlaku oleh Jayadarta an Karna.

3. Level 3

Intro: Setelah mengetahui bahwa putranya, Abimanyu terbunuh. Arjuna langsung pergi mencari Jayadarta dan Karna untuk membalas dendam atas kematian putranya yang dibunuh dengan tidak mematuhi aturan yang berlaku.

Latar : Desa

Misi : Membunuh Jayadarta dan Karna

Rintangan : Prajurit, Jayadarta dan Karna

Apa yang harus dilakukan: Berlari menemukan Jayadarta dan Karna setelah bertemu lalu membunuh Jayadarta dan Karna.

Outro: Setelah membunuh Jayadarta dan Karna, Arjuna mendapat kabar bahwa Pandawa yang lain berhasil mengalahkan dan membunuh semua pihak Korawa,

memenangkan pertarungan. Arjuna dan para Pandawa yang lain akhirrnya berhasil mengambil alih Kerajaan Astina dari pihak Korawa.

3.1.3.2 *Gameplay*

Gameplay dilakukan untuk menggambarkan aturan-aturan dalam *Game*. *Gameplay* yang terdapat pada *game adventure* "Larilah arjuna" sebagai berikut.

1. Pemain dapat langsung memulai *game* saat memilih *menu* main.
2. Sebelum memulai *game* akan ada intro yang tampil di setiap levelnya dan pemain harus memilih tombol lanjut untuk memulai permainan.
3. Pemain menggerakkan karakter Arjuna dengan men-tap layar *smartphone* agar aruna dapat berpindah dan menembak dengan men-tap tombol tembak.
4. Pemain bertugas menghindari rintangan yang ada, jika terkena rintangan maka darah Arjuna akan berkurang.
5. Pemain dapat membunuh atau menghindari prajurit tetapi jika bertabrakan dengan prajurit maka darah Arjuna akan berkurang.
6. Jika darah Arjuna telah habis maka Arjuna mati dan *game* akan berakhir.
7. Darah dapat bertambah jika Arjuna mengambil *item* buah.
8. Pemain dapat mengambil koin perak yang berguna untuk ditukarkan dengan perlengkapan yang bisa didapat pada menu perlengkapan.
9. Setiap mendapatkan koin maka koin akan bertambah 1 dan jika mengalahkan musuh besar maka koin akan bertambah 100.
10. Pemain dapat menyerang musuh besar sampai akhirnya musuh besar mati dan dapat melanjutkan ke level selanjutnya.
11. Pemain dapat membeli perlengkapan dengan menukarkan koin perak. Perlengkapan berupa baju sakti untuk kekebalan saat diserang oleh musuh.
12. Pemain dapat melihat cara bermain *game* pada menu petunjuk.
13. Pemain akan selalu memulai *game* dari level pertama saat memilih tombol main.
14. Jika pemain mati di level tertentu, pemain dapat melanjutkan permainan dari level saat pemain mati.

3.1.3.3 Tingkat Kesulitan

Dalam *game adventure* Larilah Arjuna, terdapat 3 level yang masing-masing level dibedakan berdasarkan alur cerita, tingkat kesulitan rintangan yang harus dihadapi, jumlah musuh yang bertambah. Berikut ini penjelasan tingkat level kesulitan dapat dilihat pada Tabel 3-2.

Tabel 3-2 Tingkat Level Kesulitan

Level	Tingkat level kesulitan	Deskripsi
1	Mudah	Prajurit yang akan menghadang sebanyak 5 prajurit dan terdapat 1 musuh besar yaitu Prabu Malawapati yang harus dikalahkan Arjuna dengan 4 kali panah tepat sasaran untuk melanjutkan ke level selanjutnya.
2	Sulit	Prajurit yang akan menghadang sebanyak 8 prajurit dan terdapat 1 musuh besar yaitu Bisma yang harus dikalahkan Arjuna dengan 8 kali panah tepat sasaran untuk melanjutkan ke level selanjutnya.
3	Sedang	Prajurit yang akan menghadang sebanyak 6 prajurit dan terdapat 2 musuh besar yaitu Jayadarta dan Karna yang harus dikalahkan. Arjuna. Jayadarta dikalahkan dengan 3 kali panah dan Karna dikalahkan dengan 4 kali panah agar dapa menyelesaikan *game*.

3.1.3.4 *Scoring*

Pada bagian ini akan dijelaskan bagaimana pemberian skor yang ada pada *game*. Berikut cara memperoleh score dari *game adventure* “Larilah Arjuna” :

1. Setiap pemain membunuh 1 prajurit maka akan mendapatkan 100 poin.
2. Setiap pemain mengalahkan musuh besar maka akan 1000 poin.

3.1.4 Analisis Struktur Data

Struktur data yang digunakan didalam penelitian ini adalah struktur data array. Parameter yang menggunakan array antara lain dideklarasikan dalam bentuk berikut.

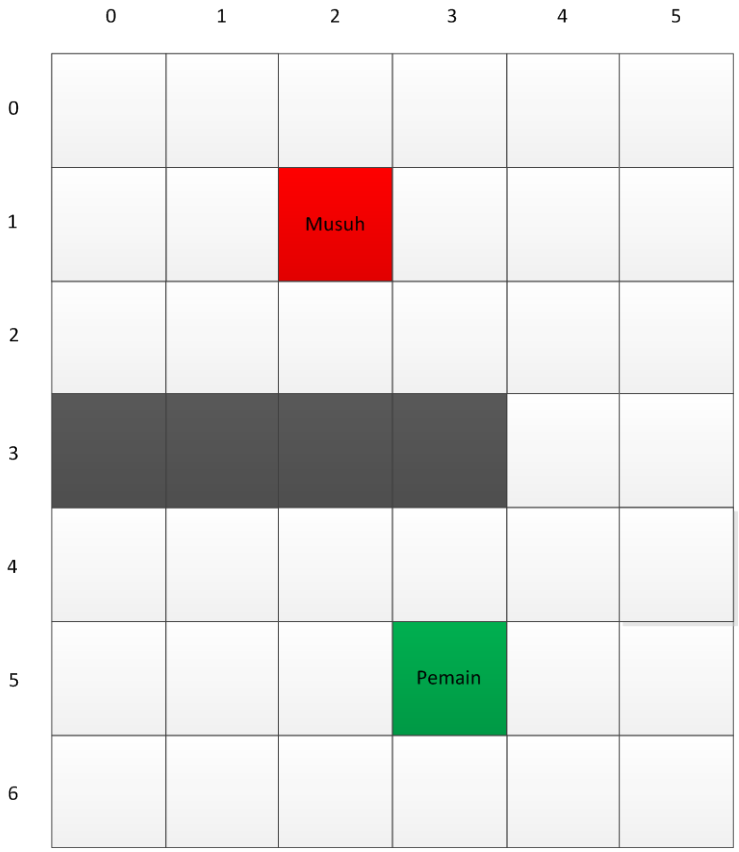
```
Deklarasi:
     xIni : array [1..14] of integer
     yIni : array [1..21] of integer
     xFin : array [1..14] of integer
     yFin : array [1..21] of integer
     level: array [1..14,1..21] of integer
     openList : array [1..14,1..21] of integer
     closedList : array [1..14,1..21] of integer
```

Parameter yang menggunakan array yang berisikan atribut xIni, yIni, xFin, yFin, level, openlist, closedlist. Atribut tersebut menggunakan array dua dimensi

dan merupakan *resource* yang akan berpengaruh dalam perhitungan nilai keputusan menggunakan algoritma A*.

3.1.5 Analisis Algoritma

Algoritma A* pada gam larilah Arjuna akan diterapkan pada NPC sebagai penentu jalur terpendek menuju pemain. Arena permainan pada *game adventure* terdiri dari orde X×Y. Dalam analisis ini akan dijelaskan dengan ordo 6×7 karena perhitungan dengan ordo berapapun hasilnya akan sama. Simpul awal atau posisi musuh dicontohkan berada pada koordinat (2,1), sedangkan simpul akhir atau posisi karakter pemain berada pada koordinat (3,5) di arena permainan.



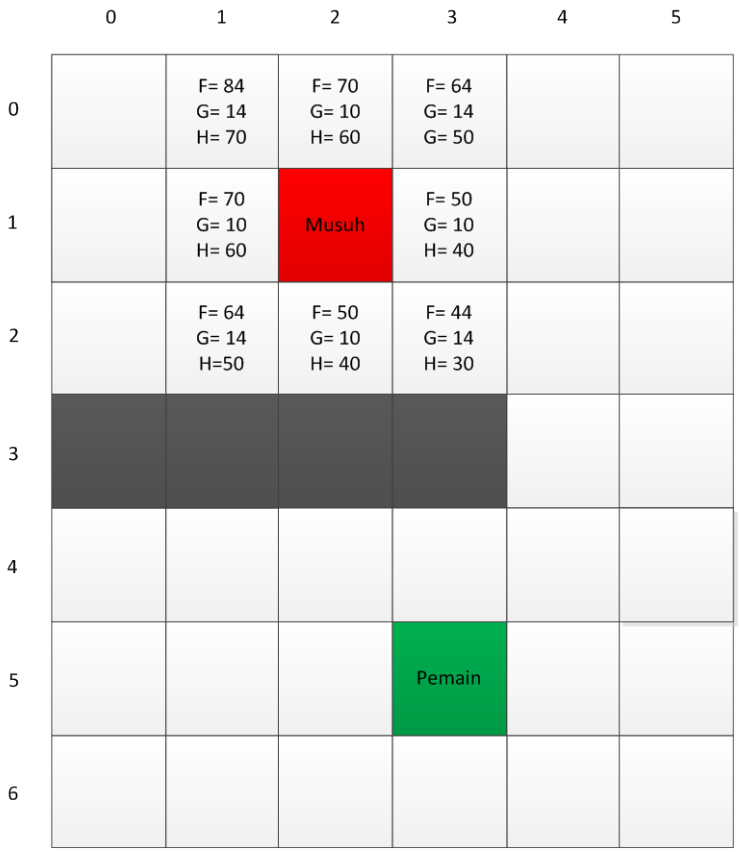
Gambar 3-3 Posisi Musuh Dan Pemain Dalam Grid

Langkah pencarian berikutnya ketika telah mendapatkan simpul awal adalah menentukan rute terpendek dengan cara memeriksa setiap simpul yang berdekatan dengan simpul awal sampai simpul akhir. Dibawah ini merupakan penjelasan langkah-langkahnya.

1. Langkah pertama adalah memasukan simpul awal ke dalam *open list.*

2. Ambil simpul awal sebagai *current node*.
3. Periksa apakah *current node* sama dengan simpul akhir atau *current node* yang sudah didapatkan merupakan simpul akhir. Jika iya maka jalan sudah ditemukan, jika tidak maka hapus simpul awal dari open list dan tambahkan ke dalam *closed list.*
4. Periksa semua *neighbor node* atau simpul yang berdekatan dengan simpul awal apakah *walkable* atau tidak, abaikan *node* dengan penghalang lalu tambahkan semua simpul tersebut ke dalam *open list*. Dan untuk setiap simpul yang baru ditambahkan, simpan awal sebagai *parent* yang nantinya akan digunakan untuk menelusuri jalan.

Proses pencarian akan menggunakan fungsi heurictic *manhattan distance,* nilai g yang digunakan adalah 10 sedangkan untuk g diagonal akan menggunakan nilai 14. Proses pencarian yang lebih jelasnya dapat dilihat pada gambar 3-4.



Gambar 3-4 Proses 1 Pencarian

Pada langkah pertama, simpul (2,1) dimasukan ke dalam *open list* dan selanjutnya memeriksa *neighbor node* atau simpul yang berdekatan dengan simpul (2,1) yaitu simpul (1,0), (2,0), (3,0), (3,1), (3,2), (2,2), (1,2), (1,1). Setelah diperiksa simpul-simpul tersebut dimasukan ke dalam *open list*. Hapus simpul (2,1) dari *open list* dan masukan kedalam *closed list*, lalu dari simpul-simpul yang berada didalam *open list* pilih simpul dengan nilai F yang paling rendah, yaitu simpul (3,2) dengan nilai F=44. Untuk memperjelas perhitungan maka akan dijelaskan langkah-langkah sebagai berikut :

1. Langkah pertama dengan menghitung simpul (1,0)

   nilai n.x= 1 nilai n.y = 0

   Simpul Tujuan (3.5) sehingga *targetX* = 3 dan *targetY* = 5

   G(1.0) = 10x√2 = 14.14 = 14

   H(1.0) = 10*(abs(*current*X-*target*X) + abs(*current*Y-*target*Y))

   = 10*(abs(1-3) + abs(0-5))

   = 10*(abs(-2) + abs(-5))

   = 10*(2+5)

   = 10*7

   = 70

   F(1,0) = 14+70

   = 84

2. Langkah ke dua dengan menghitung simpul (2,0)

   nilai n.x= 2 nilai n.y = 0

   Simpul Tujuan (3.5) sehingga *targetX* = 3 dan *targetY* = 5

   G(2.0) = 10 (nilai simpul ke simpul yang akan dihitung)

   H(2.0) = 10*(abs(*current*X-*target*X) + abs(*current*Y-*target*Y))

   = 10*(abs(2-3) + abs(0-5))

   = 10*(abs(-1) + abs(-5))

   = 10*(1+5)

   = 10*6

   = 60

   F(2.0) = 10+60

= 70

Hasil hitungan dari simpul (3,0), (3,1), (3,2), (2,2), (1,2), (1,1) dapat dilhat pada tabel 3-3.

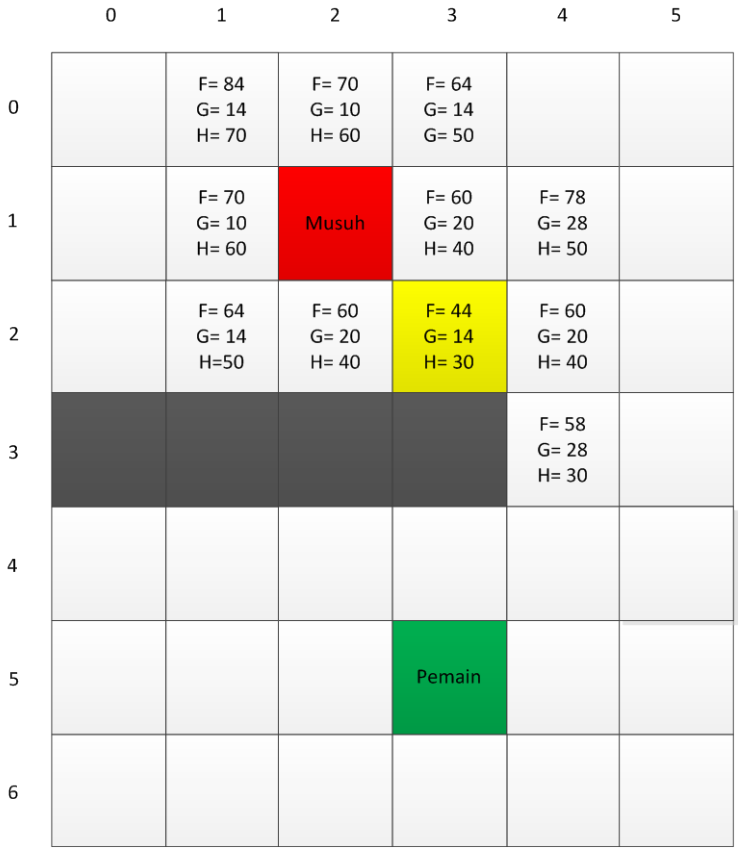
Tabel 3-3 Perhitungan Proses 1

Iterasi	Simpul	G	H	F
0	3,0	14	50	64
	3,1	10	40	50
	3,2	14	30	**44**
	2,2	10	40	50
	1,2	14	50	64
	1,1	10	60	70

Proses dijelaskan diatas, dilakukan berulang kali atau *looping*, dimulai dari *open list* dan menemukan simpul dengan nilai F paling rendah. Langkahnya adalah sebagai berikut :

1. Hapus *current node* tersebut ke dalam *open list* dan masukan ke dalam *closed list.*
2. Periksa semua simpul yang berdekatan dengan *current node*, abaikan simpul yang *unwalkable* atau node dengan penghalang atau *collider*. Jika tidak ada pada *open list*, tambahkan ke dalam *open list.* Dan tambahkan *parent* untuk simpul tersebut.
3. Jika simpul sudah ditemukan pada *open list*, periksa apakah ini jalan yang lebih baik dari simpul awal, dengan kata lain periksa apakah nilai G lebih rendah apabila menggunakan simpul ini, jika lebih rendah maka hitung ulang niali F dan G.

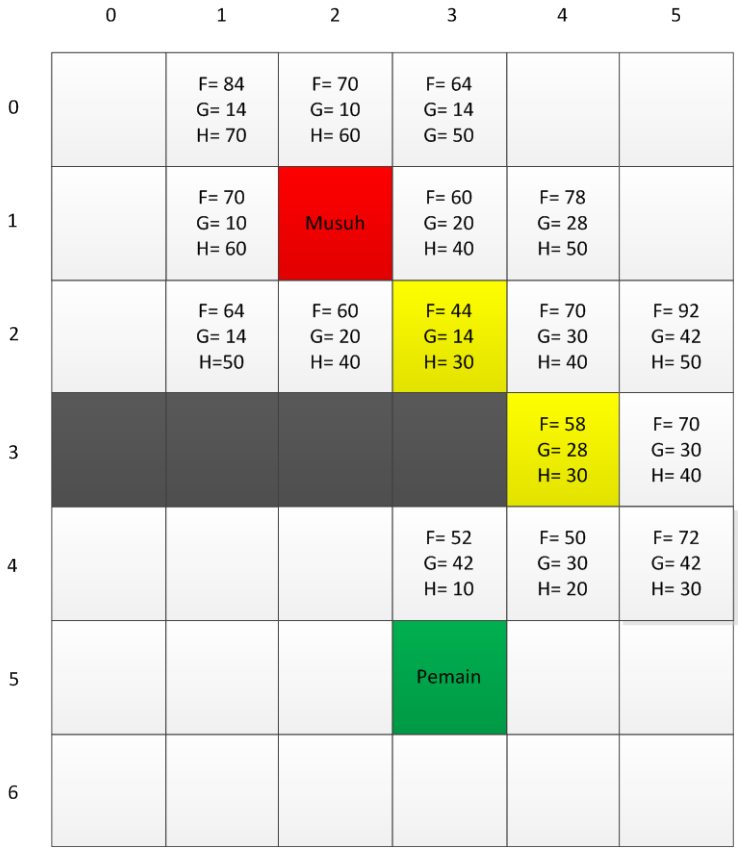
Berikut ini perhitungan pada langkah kedua yang berada pada simpul dengan koordinat (3,2).



Gambar 3-5 Proses 2 Pencarian

Proses pencarian pada gambar 3-5 terlihat ada 5 simpul yang masuk ke dalam *open list* yaitu simpul dengan koordinat (3,1), (4,1), (4,2), (4,3), dan (2,2) sedangkan untuk simpul (2,1) dan (3,2) merupakan simpul *closed list*, sementara simpul (3,3) dan (2,3) merupakan penghalang. Dari langkah kedua simpul yang masuk ke dalam *open list* terdapat niali F yang paling rendah yaitu simpul (4.3) dengan nilai F = 58.

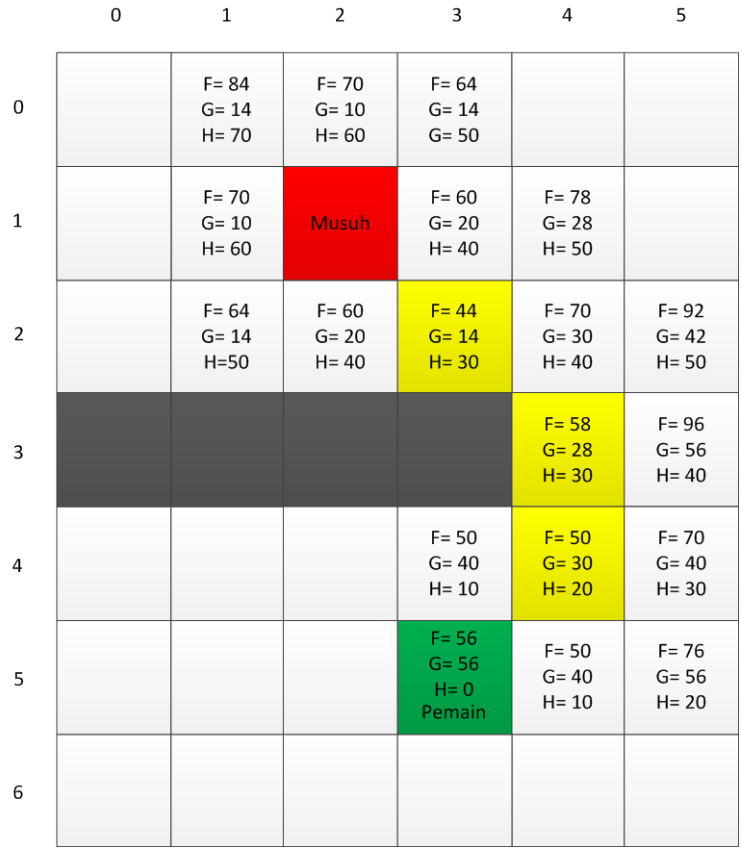
Proses pencarian pada gambar 3-6 menunjukan perhitungan pada langkah ketiga yang berada pada node (4,3).



Gambar 3-6 Proses 3 Pencarian

Pada gambar 3-6 terlihat ada 6 simpul yang masuk ke dalam *open list* yaitu simpul dengan koordinat (4,2), (5,2), (5,3), (5,4), (4,4) dan (3,4) sedangkan untuk simpul (3,2) dan (4,3) merupakan simpul *closed list*, sementara simpul (3,3) merupakan penghalang. Dari langkah ketiga simpul yang masuk ke dalam *open list* terdapat niali F yang paling rendah yaitu simpul (4.4) dengan nilai F = 50.

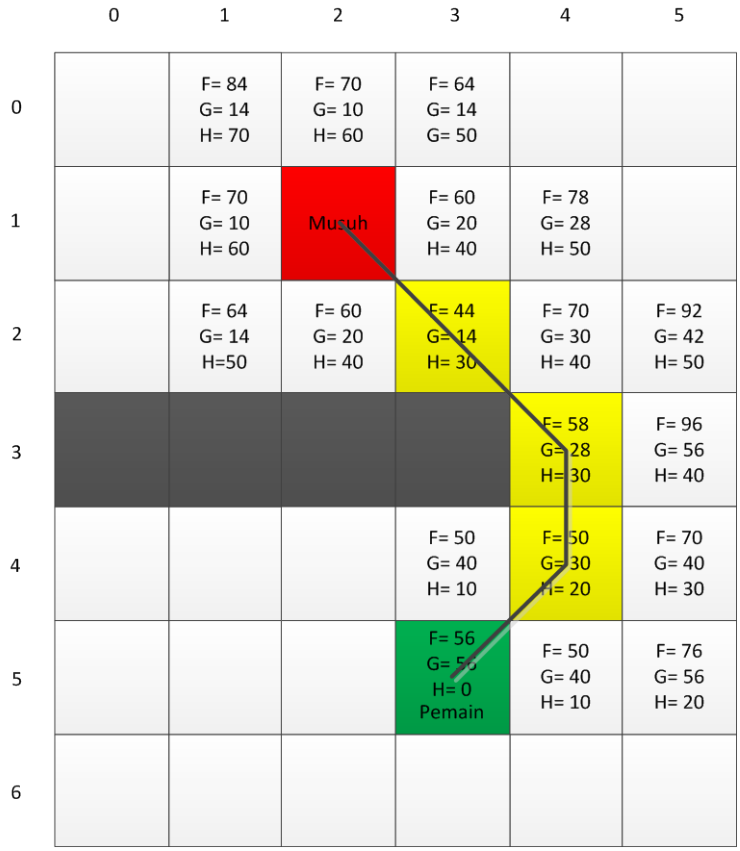
Pada gambar 3-7 menunjukan perhitungan pada langkah keempat yang berada pada node (4,4).



Gambar 3-7 Proses 4 Pencarian

Proses pencarian pada gambar 3-7 terlihat ada 6 simpul yang masuk ke dalam *open list* yaitu simpul dengan koordinat (5,3), (5,4), (5,5), (4,5), (3,5) dan (3,4) sedangkan untuk simpul (4,3) dan (4,4) merupakan simpul *closed list*, sementara simpul (3,3) merupakan penghalang. Dari langkah keempat simpul (3,5) dengan nilai F = 56 dan merupakan simpul tujuan dicapai sehingga pencarian pun selesai.

Penentuan jalan terbaik dengan melakukan pemeriksaan dimulai dari simpul awal, dalam ilustrasi adalah simpul yang berwarna musuh. Dengan mengikuti arah garis yang berarti akan membawa kembali ke simpul tujuan, maka akan didapatkan jalan terbaik, maka jika diimplementasikan pada *game*, nantinya NPC akan menemukan rute terpendek untuk samapai ke tujuan, untuk lebih jelasnya seperti pada Gambar 3-8



Gambar 3-8 Jalur pencarian yang ditempuh NPC

Berikut adalah *pseudo-code* algoritma A* yang ada di dalam *game* Larilah Arjuna pada tabel 3-4.

Tabel 3-4 Pseudocode Algoritma A*

	Pseudocode Algoritma A*
1	Deklarasi
2	Finx : array
3	Finy : array
4	Level : array
5	Openlist : dictionary
6	Closedlist : dictionary
7	Path :array
9	Inisialisasi
10	finX <- xFin
11	finY <- yFin
12	level <- lvlData
13	openList <- new Dictionary
14	closedList <- new Dictionary
15	path <- new Array
16	//algoritma a* pathfinding
17	Function SearchLevel()
18	curNode : PathNode
19	lowF : <integer>
20	finished : boolean

Pseudocode Algoritma A*

```
21        lowF <- 100000
22        finished <- false
23  //pertama tentukan node dengan nilai F terkecil
24      for each node in openList
25               curF <- node.g + node.h
26               if lowF > curF then
27                     lowF = curF
28                     curNode = node
29             endif
30        endfor
31  //jalan tidak ditemukan!
32    if curNode = null then
33        return
34    endif
35  //ubah node terpilih dari open ke closed
36    hapus openList[curNode.x + " " + curNode.y]
37    closedList[curNode.x + " " + curNode.y] <- curNode
38  //cek target
39  if curNode.x = finX and curNode.y = finY then
40        endNode = curNode
41        finished = true
42        endif
43  //cek disetiap node yang akan dilewati
44     for  i <- -1 to i < 2 do i+1
45      for j <- -1 to j < 2 do j+1
46          col <- curNode.x + i
47          row <- curNode.y + j
48      //pastikan grid bukan current node
49        if (col > 0 and col < level[0].length) and (row > 0 and
50              row < level.length) and (i ≠  0 or j ≠ 0) then
51          //cek g adalah open list
52             if level[row][col] = 0 and closedList[col + " " +
53               row] = null and openList[col + " " + row] = null
54               then
55                //menetukan g
56                   g <- 10
57                    if i ≠ 0 and j ≠ 0 then
58                         g <- 14
59                   end if
60                 //menetukan h
61                  h  <-  absolut(col  -  finX))  +  absolut(row  -
62                    finY)) * 10
63                //membuat node
64                    found  <-  new  PathNode(col,  row,  g,  h,
65                    curNode)
66                    openList[col + " " + row] <- found
67              end if
68        end if
69      endfor
70     endfor
71  //temukan jalan lain jika target tidak tercapai
72         if finished = false then
73           SearchLevel() do
74           else
75           RetracePath() do
```

Pseudocode Algoritma A*	
76	end if
77	// membangun berbagai poin dengan menapak mencari node
78	Function RetracePath
79	step <- new Point(node.x, node.y)
80	path.push(step)
81	if node.g > 0 then
82	RetracePath(node.parentNode) do
83	end if

3.1.6 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 keras, analisis kebutuhan perangkat lunak, dan analisis pengguna.

3.1.5.1 Analisis Kebutuhan Perangkat Keras

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.

a. Dari sisi pengembang

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

1. Prosesor 2 GHz
2. Memori 4 GB
3. Space kosong di harddisk 5 GB
4. VGA *card* 512 MB
5. *Display* LCD Monitor dengan resolusi 1366 x 768

b. Dari sisi pemain

Spesifikasi minimum perangkat keras yang dibutuhkan agar dapat menjalankan *game* ini adalah sebagai berikut:

1. *Smartphone* Android berlayar *touchscreen*
2. *Processor* 1 GHz, 512 MB RAM

Berdasarkan analisis yang telah dilakukan maka, kebutuhan minimum perangkat keras yang diperlukan dari sisi pengembang adalah prosesor 2GHz dengan memori 4GB, space kosong di hardisk 512 MB, dan dari sisi pemain adalah *smartphone* Android berlayar *touchscreen* dengan posesor 1 GHz, 256 MB RAM.

3.1.5.2 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. . Di bagian ini akan dijelaskan analisis kebutuhan perangkat keras dari sisi pengembang dan dari sisi pemain.

a. Dari sisi pengembang

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

1. Sistem operasi *Windows*
2. Adobe Flash 5.5

b. Dari sisi pemain

Game adventure “Larilah Arjuna” ini dibutuhkan perangkat lunak untuk menjalankannya, yaitu Sistem Operasi Android 2.2 (*Froyo*) keatas.

Beradasarkan analisis yang telah dilakukan untuk membangun *game* Larilah Arjuna, kebutuhan perangkat lunak yg dibutuhkan dari sisi pengembang adalah Sistem operasi windows dan Adobe Flash 5.5 sedangkan dari sisi pemain perangkat lunak g dibutuhkan untuk menjalankan *game* adalah Android dengan sistem operasi *Froyo* (2.2) karena sistem operasi *Froyo* sudah mendukung file flash.

3.1.5.3 Analisis Pengguna

Analisis pengguna gambaran tentang target maupun profil pengguna untuk *game* yang akan dibangun. Perangkat keras dan perangkat lunak yang ada tidak akan berguna apabila tidak ada pengguna yang mengoperasikannya. Berikut ini analisis pengguna untuk *game adventure* "Larilah Arjuna", dapat dilihat pada tabel 3-5.

Tabel 3-5 Analisis Pengguna

Karakeristik fisik pengguna	
Umur	12 tahun keatas
Gender	Laki-lagi dan perempuan
Handedness	Tangan kanan dan kiri
Disabilities	Tidak buta warna total
Pengetahuan dan pengalaman pengguna	
Computer Literacy	Memiliki pengalaman dalam mengoperasikan sebuah *smartphone* android
System Experience	Pernah memainkan aplikasi *game mobile*
Aplication Experience	Pernah memainkan *game* sejenis
Education	Sekolah dasar
Task Experience	Mengetahui cara menggunakan Android
Native language or culture	Bahasa Indonesia
Karakteristik Psikologi pengguna	
Motivation	Mempunyai motivasi yang cukup dalam menyelesaikan permainan
Patience	Mempunyai kesabaran dalam menyelesaikan permainan
Stress Level	Tidak mudah *stress*
Attitude	*Neutral*

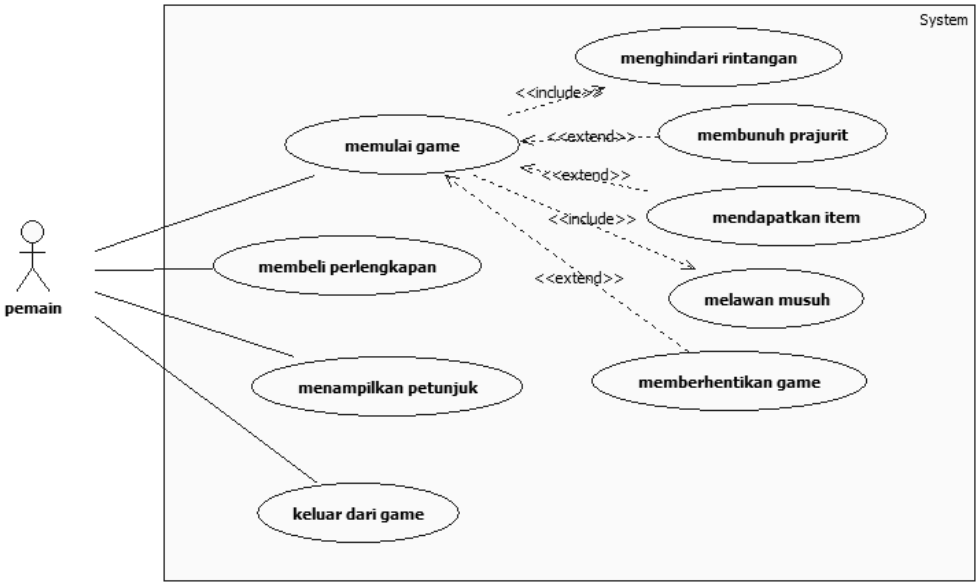
Dari analisis pengguna yang telah dilakukan, maka dapat disimpulkan bahwa pengguna yang akan memainkan *game* merupakan pengguna dari umur 12 tahun keatas, tidak memiliki cacat pada penglihatan, dapat mengoperasikan *smartphone* bersistem operasi Andorid, dapat atau pernah memainkan aplikasi *game* pada Android.

3.1.7 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, sequence diagram* dan *class diagram*.

3.1.6.1 *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 3-9.



Gambar 3-9 *Use case Diagram Game* "Larilah Arjuna"

1. Actor Definition

Actor Definition berfungsi untuk menjelaskan *Actor* yang terdapat pada *Use case diagram*. *Actor Definition* diterangkan pada tabel 3-6 berikut.

Tabel 3-6 *Actor Definition*

No	*Actor*	Deskripsi
1	Pemain	Orang yang memainkan *game*

2. Use case Definition

Use case Definition berfungsi untuk menjelaskan fungsi *Use case* yang terdapat pada *Use case diagram. Use case Definition* diterangkan pada tabel 3-7.

Tabel 3-7 *Use case Definition*

No.	*Use case*	Deskripsi
1	Memulai *game*	Proses untuk memulai *game*
2	Menghindari rintangan	Proses untuk menghindari rintangan

3	Membunuh prajurit	Proses untuk membunuh prajurit
4	Mendapatkan item	Proses untuk mengambil *item*
5	Melawan musuh	Proses untuk melawan musuh
6	Membeli perlengkapan	Proses untuk membeli perlengkapan
7	Memberhentikan *game*	Prses untuk memberhentikan *game*
8	Menampilkan petunjuk	Proses untuk menampilkan cara bermain *game*
9	Keluar dari *game*	Proses untuk keluar dari *game*

3. Use case Scenario

Use case Scenario mendeskripsikan urutan langkah-langkah dalam proses bisnis baik yang dilakukan aktor terhadap sistem maupun yang dilikakukan oleh sistem terhadap aktor. Berdasarkan *use case diagram* pada gambar 3-9, maka *use case scenario* untuk aplikasi yang dibangun akan dijelaskan sebagai berikut.

Tabel 3-8 *Use case Scenario* Memulai *Game*

Identifikasi	
Nomor	1
Nama	Memulai *game*
Tujuan	Untuk memulai *game*
Deskripsi	Proses untuk memulai *game*
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada di menu utama
Aksi Aktor	**Reaksi Sistem**
1. Memilih menu main	
	2. Menampilkan intro *game*
3. Menekan tombol skip	
	4. Menampilkan arena *game*
Kondisi Akhir	Masuk ke tampilan arena *game*

Tabel 3-9 *Use case Scenario* Menghindari Rintangan

Identifikasi	
Nomor	2
Nama	Menghindari rintangan
Tujuan	Untuk menghindari ritangan
Deskripsi	Proses menghindari rintangan
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada pada arena permainan
Aksi Aktor	**Reaksi Sistem**
1. Mentap layar smartphone	
	2. Menggerakkan karakter pemain
	3. Cek collison dengan rintangan
	4. Tidak mengenai rintangan maka dapat melanjutkan perjalanan
Skenario Tambahan	
	5. Mengenai rintangan maka mengurangi darah.
Kondisi Akhir	Rintangan terlewati

Tabel 3-10 *Use case Scenario* Membunuh Prajurit

Identifikasi	
Nomor	3
Nama	Membunuh prajurit
Tujuan	Untuk memunuh prajurit
Deskripsi	Proses membunuh prajurit
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada pada arena permainan
Aksi Aktor	**Reaksi Sistem**
1. Mentap layar smartphone	
	2. Menggerakkan karakter pemain
	3. Cek collison dengan prajurit
	4. Jika mengenai prajurit dan posis pemain lebih lebih tinggi maka prajurit hilang dari arena
	5. Menambahkan 100 poin
Skenario Tambahan	
	6. Jika tidak mengenai prajurit maka melanjutkan perjalanan.
	7. Jika mengenai prajurit dan posos pemain berada di bawah parajurit maka darah pemain berkurang
Kondisi Akhir	Prajurit terbunuh

Tabel 3-11 *Use case* Scenario Mendapatkan Item

Identifikasi	
Nomor	4
Nama	Mendapatkan item
Tujuan	Untuk mendapatkan item
Deskripsi	Proses mendapatkan item
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada pada arena permainan
Aksi Aktor	**Reaksi Sistem**
1. Mentap layar smatrphone	
	2. Menggerakkan karakter pemain
	3. Cek collison dengan item
	4. Jika mengenai item maka item bertambah
Skenario Tambahan	
	5. Jika tidak mengenai item maka item idak bertambah.
Kondisi Akhir	Item berhasil didapatkan

Tabel 3-12 *Use case* Scenario Melawan Musuh

Identifikasi	
Nomor	5
Nama	Melawan musuh
Tujuan	Untuk melawan musuh
Deskripsi	Proses melawan musuh
Aktor	Pemain

Skenario Utama	
Kondisi awal	Pemain berada pada arena permainan
Aksi Aktor	**Reaksi Sistem**
1. Menekan go	
	2. Masuk arena bos
3. Menggerakkan arjuna	
	4. Menggerakan arjuna dan menggerakkan musuh
5. Menembakkan panah kearah musuh	
	6. Mengecek collison dengan musuh
	7. Jika mengenai musuh darah musuh berkurang
	8. Musuh mencari pemain
	9. Musuh menyerang pemain
	10. Cek collison dengan pemain
	11. Jika terkena panah musuh maka dara pemin berkurang
Skenario Tambahan	
	12. Tidak mengenai musuh maka darah musuh tetap.
	13. Jika panah tidak terkena pemain maka darah pemain tetap
	14. Jika darah musuh habis maka musuh mati
	15. Jika darah pemain habis maka pemain mati
Kondisi Akhir	Darah musuh berkurang

Tabel 3-13 *Use case* Scenario Membeli Perlengkapan

Identifikasi	
Nomor	6
Nama	Membeli perlegkapan
Tujuan	Untuk membeli perlengkapan
Deskripsi	Proses mebeli perlengkapan
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada pada menu utama
Aksi Aktor	**Reaksi Sistem**
1. Memilih menu perlengkapan	
	2. Menapilkan perlengkapan *game*
3. Memilih perlengkapan yang akan dibeli	
	4. Sistem melakukan pengecekan jumlah koin
	5. Jika koin cukup, menampilakan pesan bembelian berhasil
6. Menekan tombol ok	
Skenario Tambahan	
	7. Jika koin tidak cukup, menampilkan pesan koin tidak cukup
8. Menekan tombol ok	
Kondisi Akhir	Perlengkapan dibeli

Tabel 3-14 *Use case* Scenario Memberhentikan *Game*

Identifikasi	
Nomor	7
Nama	Memberhentikan *game*
Tujuan	Untuk menghentikan *game* sementara
Deskripsi	Proses memberhentikan *game*
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada pada arena permainan
Aksi Aktor	**Reaksi Sistem**
1. Memilih menu pause	
	2. Memberhentikan permainan
3. Pemain meilih play	
	4. Melanjutkan permainan
Skenario Tambahan	
5. Pemain meilih menu	
	6. Kembali ke menu utama
Kondisi Akhir	Permainan berhenti

Tabel 3-15 *Use case* Scenario Menampilkan Petunjuk

Identifikasi	
Nomor	8
Nama	Menampilkan petunjuk
Tujuan	Untuk menampilkan petunjuk *game*
Deskripsi	Proses untuk menampilkan petunjuk
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada di menu utama
Aksi Aktor	**Reaksi Sistem**
1. Memilih menu petunjuk	
	2. Menampilkan cara bermain *game*
3. Memilih kembali	
	4. Kembali ke menu utama
Kondisi Akhir	Petunjuk tampil

Tabel 3-16 *Use case* Scenario Keluar dari *Game*

Identifikasi	
Nomor	9
Nama	Keluar dari *game*
Tujuan	Untuk keluar dari *game*
Deskripsi	Proses keluar dari gme
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada di menu utama
Aksi Aktor	**Reaksi Sistem**
1. Memilih menu keluar	
	2. Menampilkan pesan keluar
3. Memilih ya	
	4. Keluar dari *game*
Skenario Tambahan	

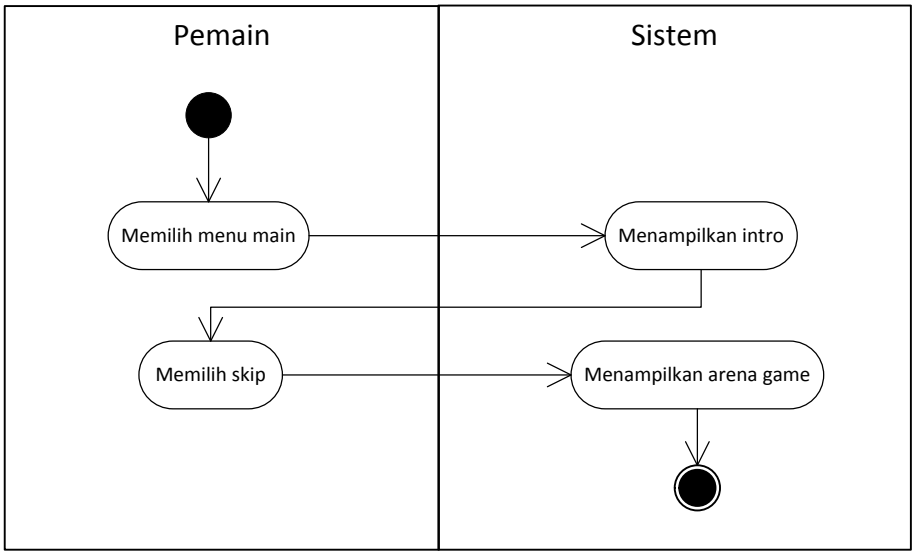
5. Memilih tidak	
	6. Kembali ke menu utama
Kondisi Akhir	Keluar dari aplikasi

3.1.6.2 *Acvity Diagram*

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. Berikut ini *activity diagram* pada *game adventure* "Larilah Arjuna".

1. *Activity Diagram* Memulai *game*

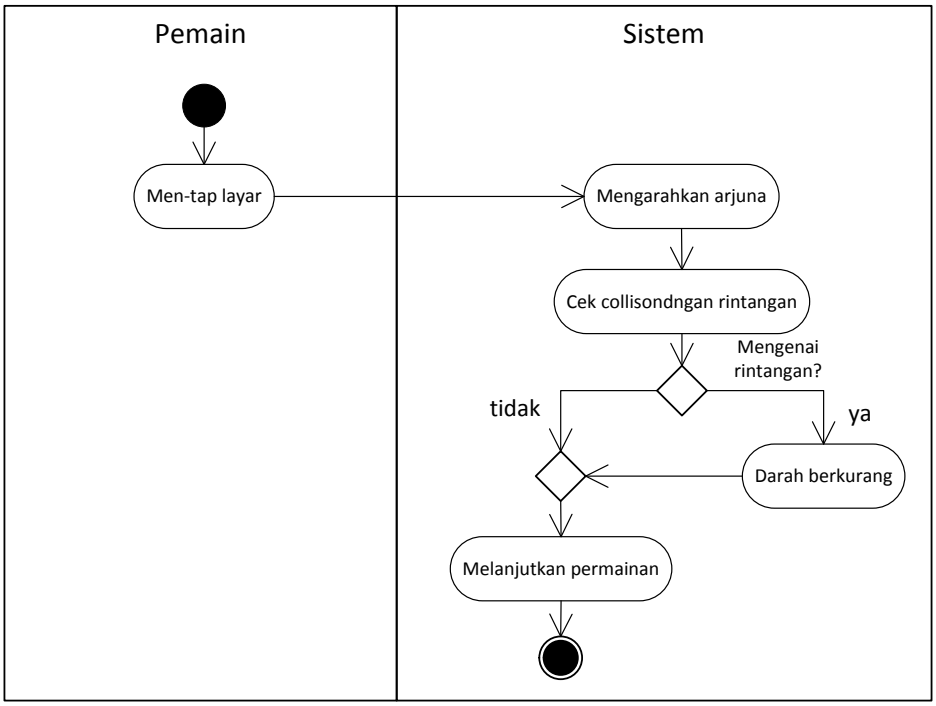
   Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas memulai *game* yang dapat dilihat pada gambar 3-10



Gambar 3-10 *Activity Diagram* Memulai *Game*

2. *Activity Diagram* Menghindari Rintangan

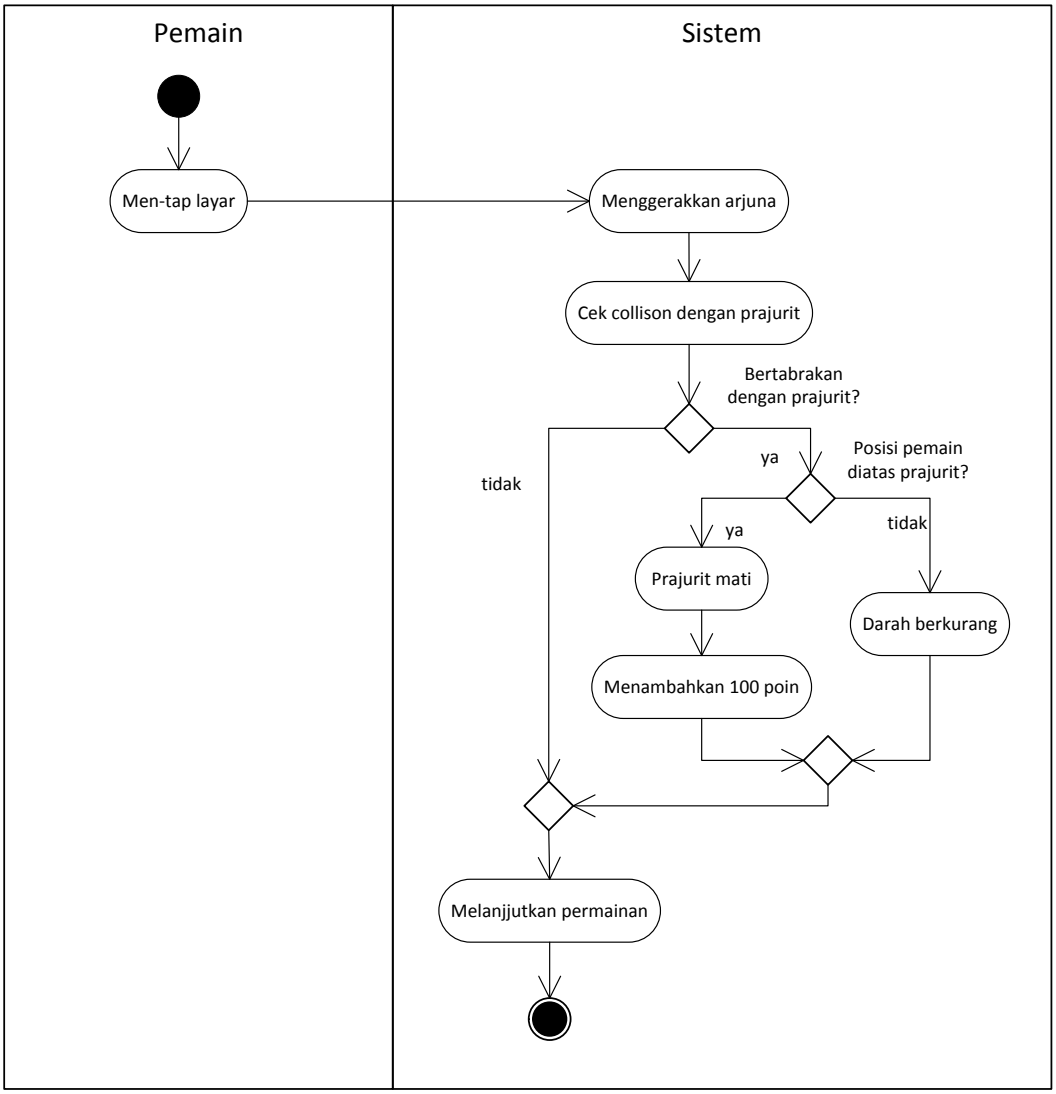
   Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas menghindari rintangan yang dapat dilihat pada gambar 3-11.



Gambar 3-11 *Activity Diagram* Menghindari Rintangan

3. *Activity Diagram* Membunuh Prajurit

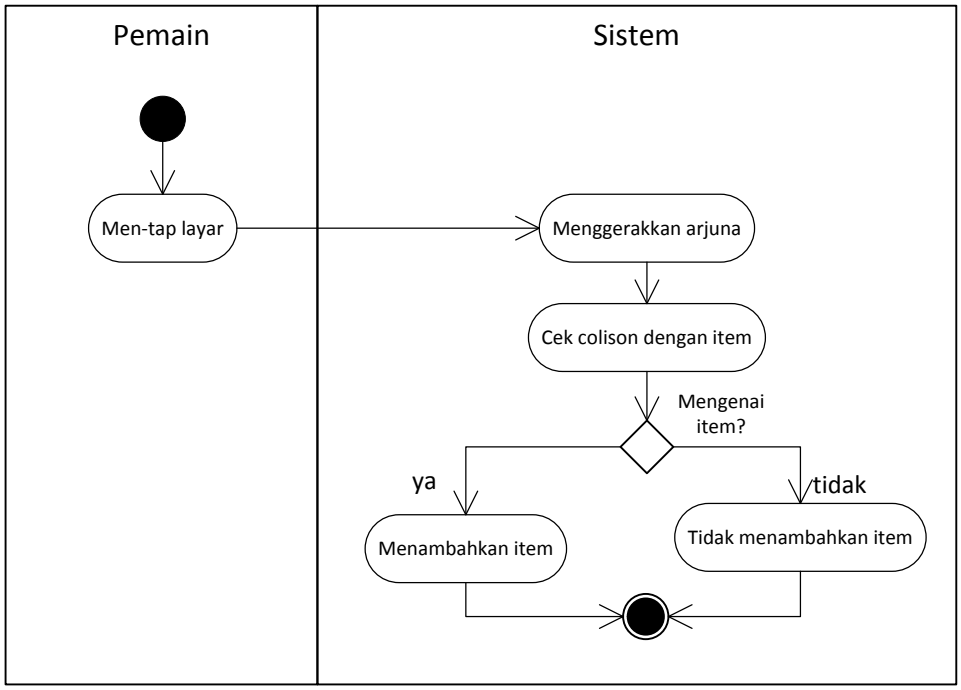
   Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas membunuh prajurit yang dapat dilihat pada gambar 3-12.



Gambar 3-12 *Activity Diagram* Membunuh Prajurit

4. *Activity Diagram* Mendapatkan *Item*

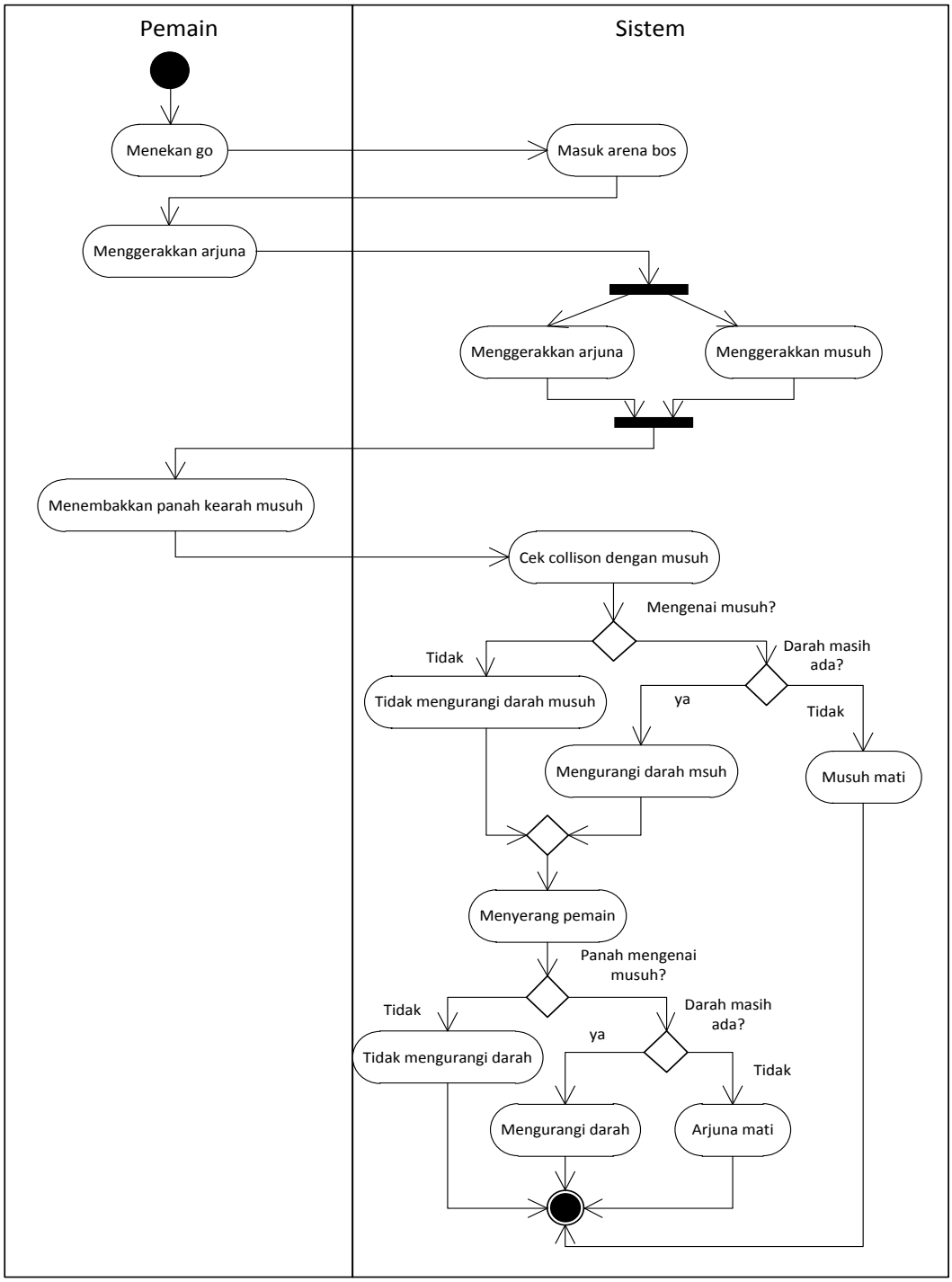
   Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas mendapatkan *item* yang dapat dilihat pada gambar 3-13.



Gambar 3-13 *Activity* Diagram Mendapatkan Item

5. *Activity Diagram* Melawan Musuh

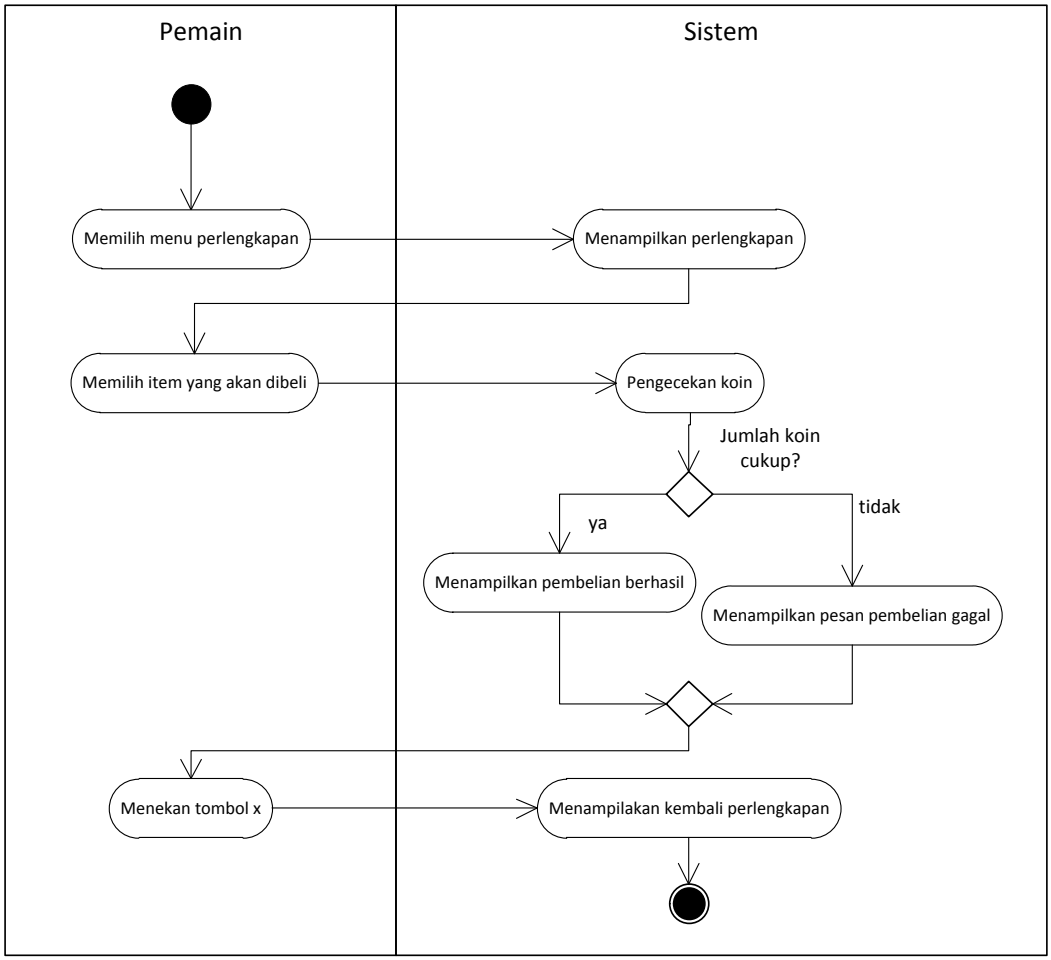
Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas melawan musuh yang dapat dilihat pada gambar 3-14.



Gambar 3-14 *Activity Diagram* Melawan Musuh

6. *Activity Diagram* Membeli Perlengkapan

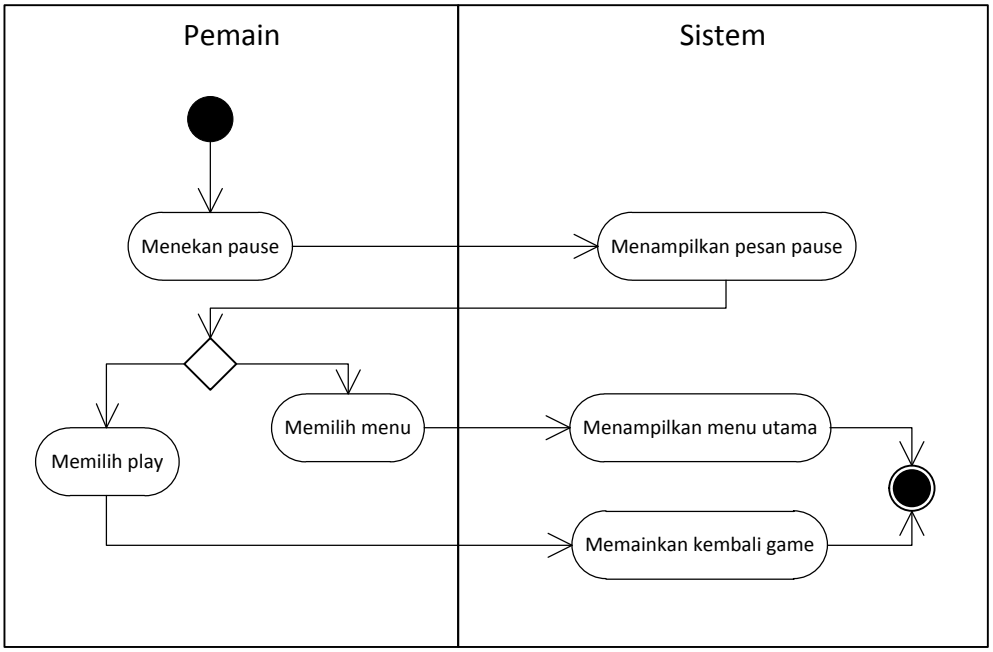
   Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas membeli perlengkapan yang dapat dilihat pada gambar 3-15.



Gambar 3-15 *Activity Diagram* Membeli Perlengkapan

7. *Activity Diagram* Memberhentikan *game*

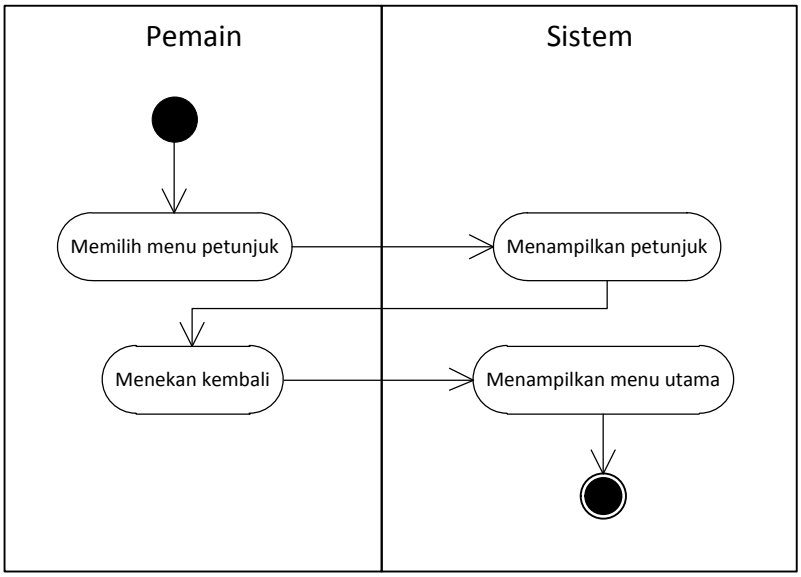
   Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas memberhentian gam sementara yang dapat dilihat pada gambar 3-16.



Gambar 3-16 *Activity Diagram* Memberhentikan *Game*

8. *Activity Diagram* Menampilkan Petunjuk

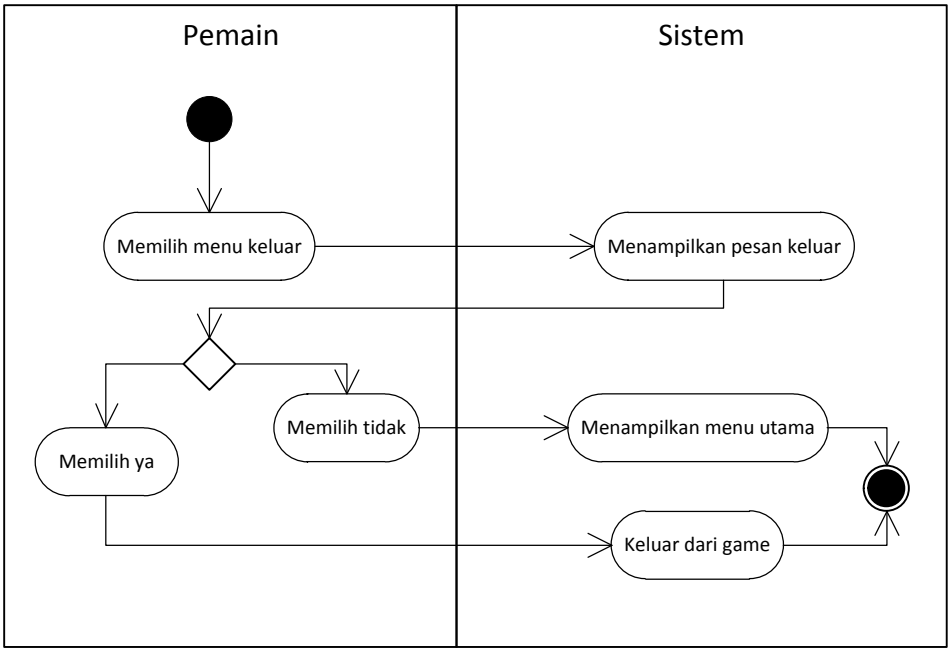
   Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas menampilkan petunjuk yang dapat dilihat pada gambar 3-17.



Gambar 3-17 *Activity Diagram* Menampilkan Petunjuk

9. *Activity Diagram* Keluar dari *Game*

   Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas keluar dari *game* yang dapat dilihat pada gambar 3-18.

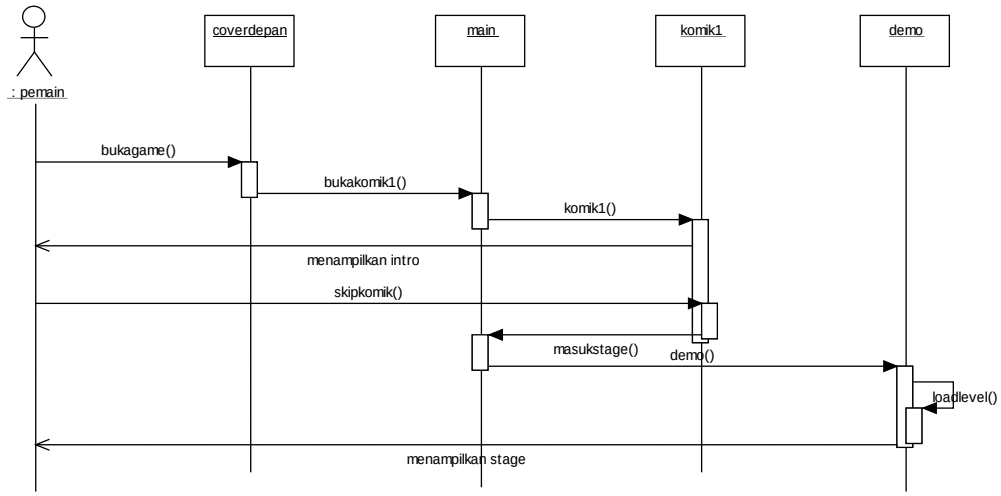


Gambar 3-18 *Activity Diagram* Keluar dari *Game*

3.1.6.3 *Sequence Diagram*

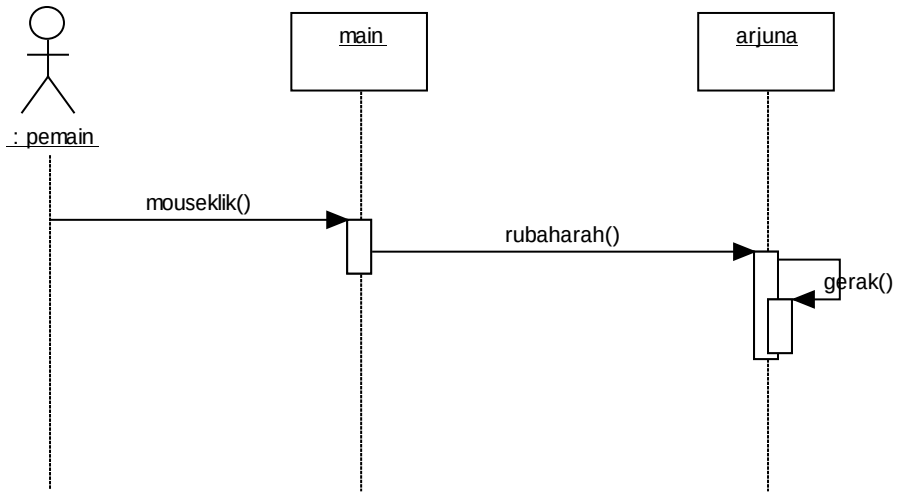
Sequence diagram menggambarkan tindakan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Berikut ini *sequence diagram* pada *game adventure* “Larilah Arjuna”.

1. *Sequence Diagram* Memulai *Game*



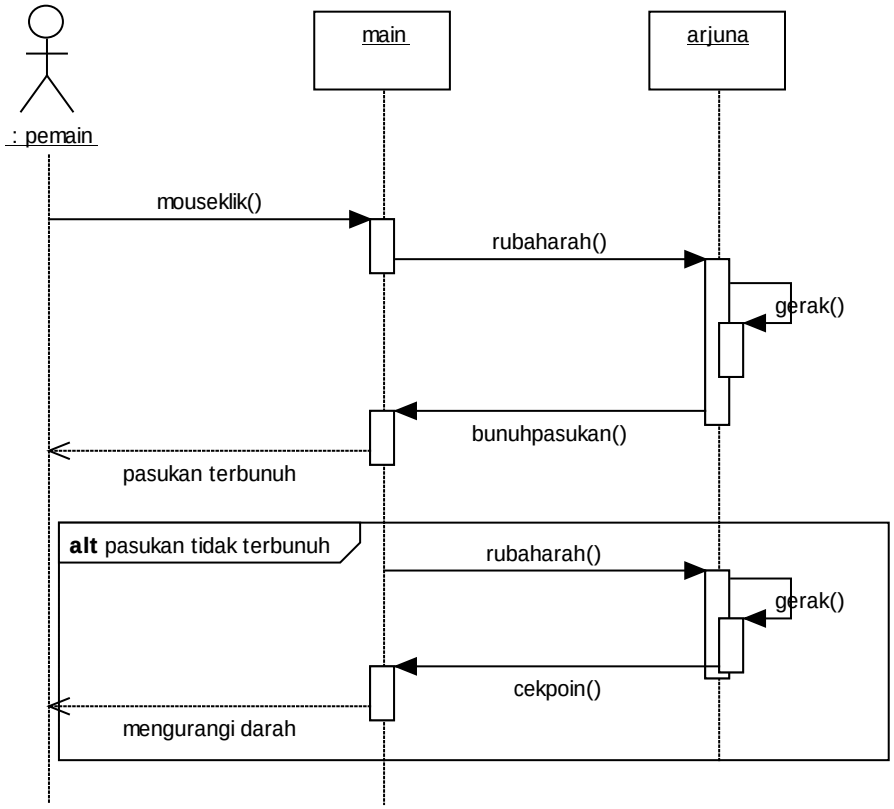
Gambar 3-19 *Sequence Diagram* Memulai *Game*

2. *Sequence Diagram* Menghindari Rintangan



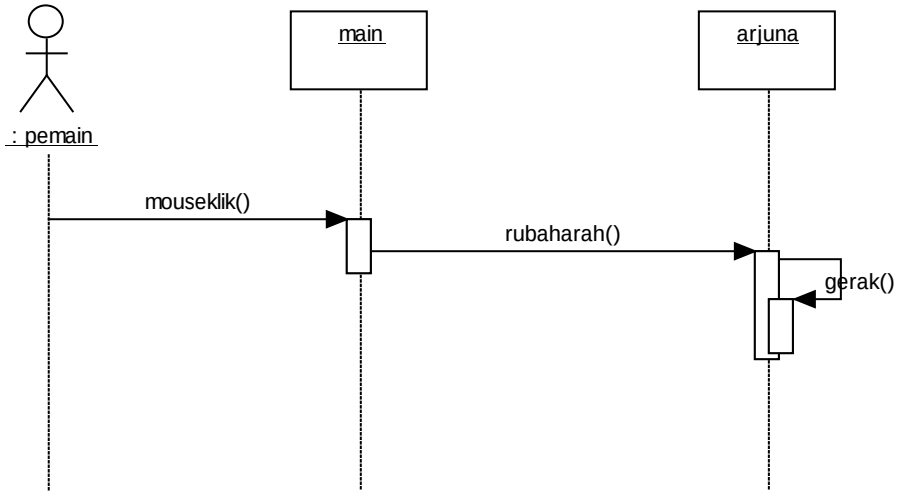
Gambar 3-20 *Sequence Diagram* Menghindari Rintangan

3. *Sequence Diagram* Membunuh Prajurit



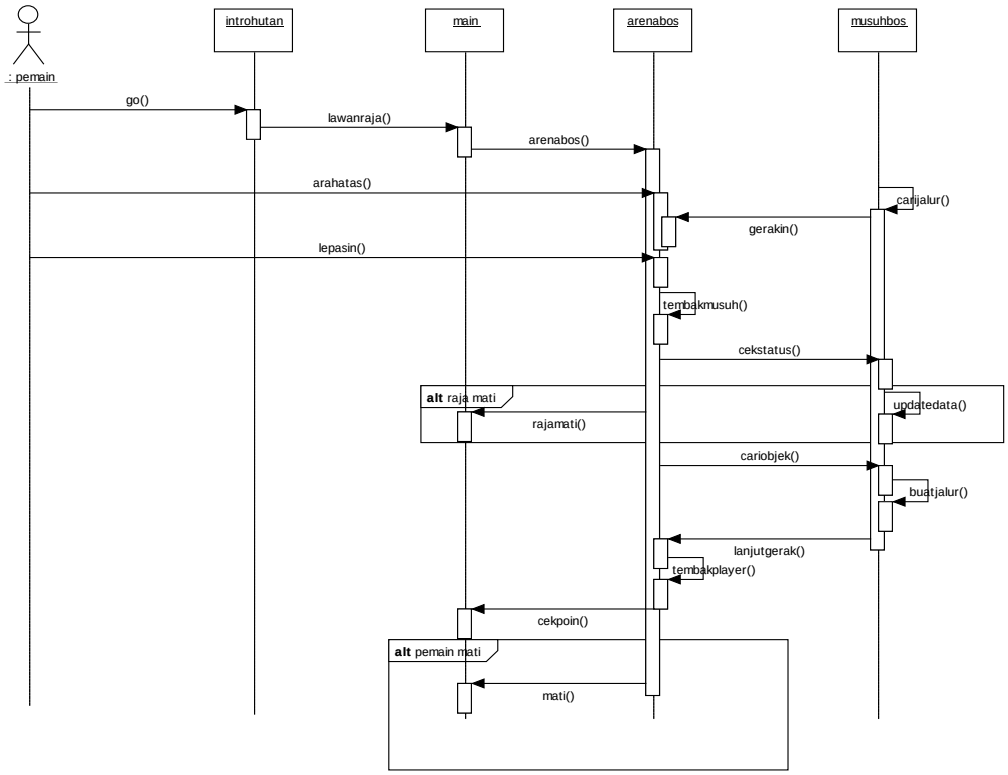
Gambar 3-21 *Sequence Diagram* Membunuh Prajurit

4. *Sequence Diagram* Mendapatkan Item



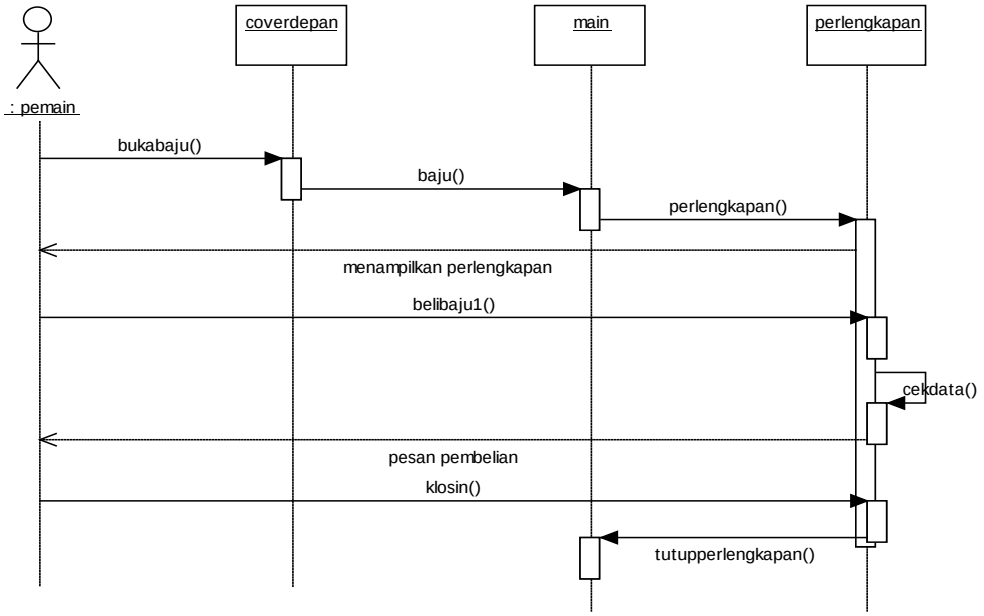
Gambar 3-22 *Sequence Diagram* Mendapatkan Item

5. *Sequence Diagram* Melawan Musuh



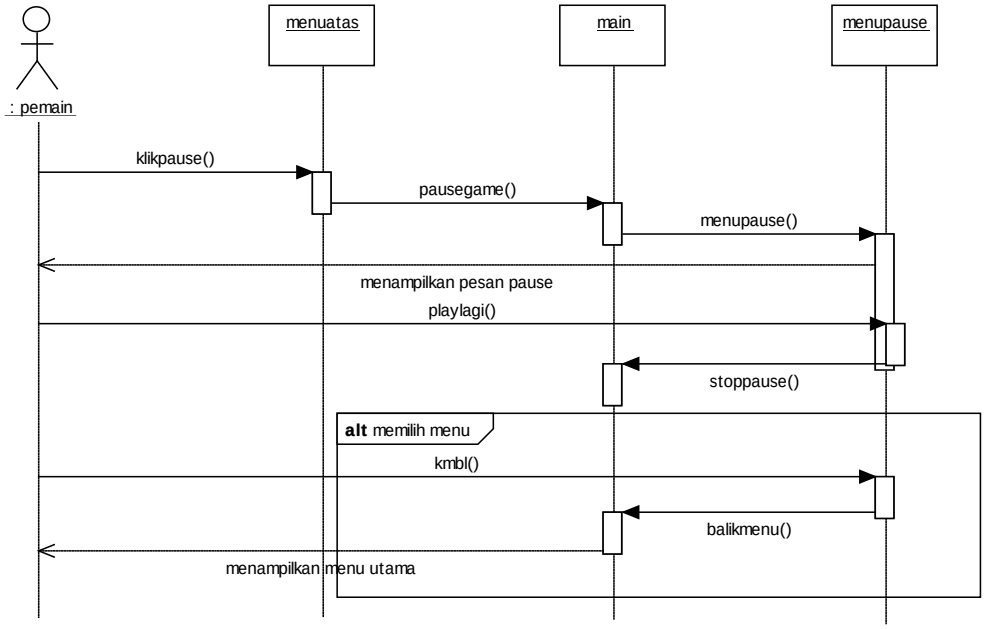
Gambar 3-23 *Sequence Diagram* Melawan Musuh

6. *Sequence Diagram* Membeli perlengkapan



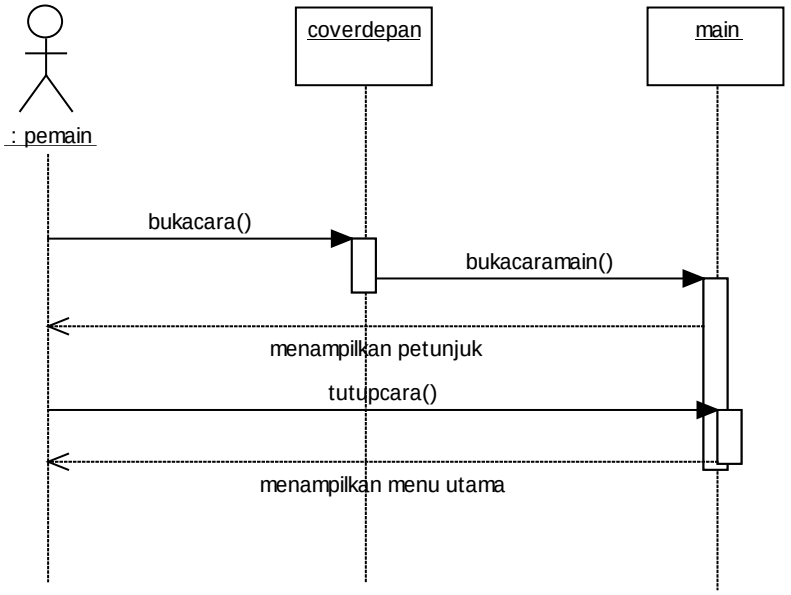
Gambar 3-24 *Sequence Diagram* Membeli Perlengkapan

7. Sequence Diagram Memberhentikan *Game*



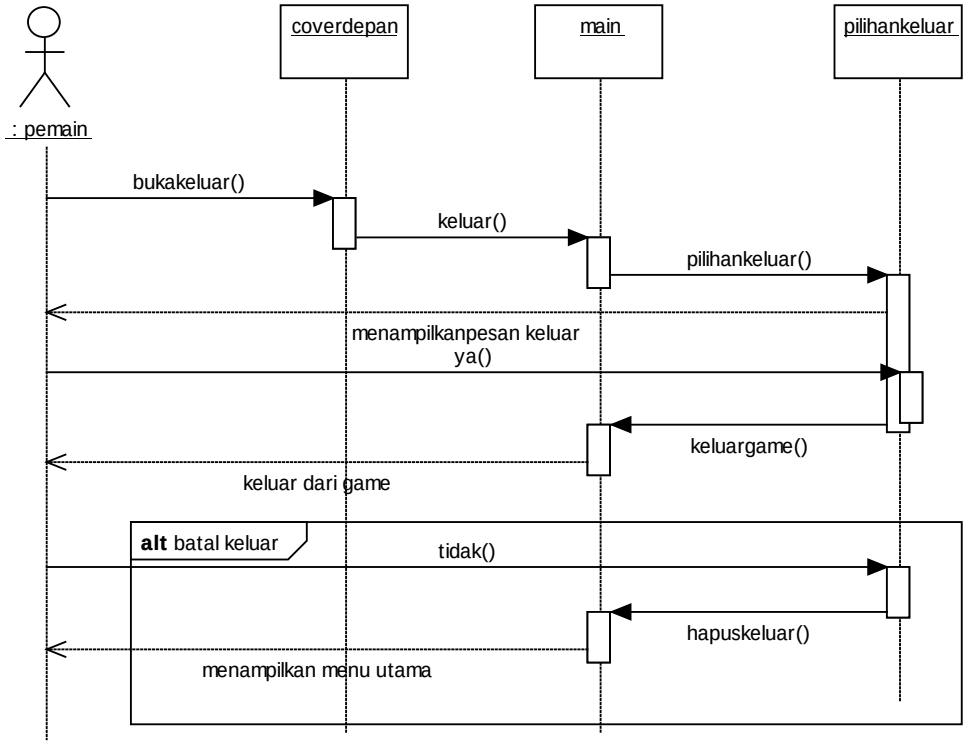
Gambar 3-25Sequence Diagram Memberhentikan *Game*

8. *Sequence Diagram* Menampilkan Petunjuk



Gambar 3-26 *Sequence Diagram* Menampilkan Petunjuk

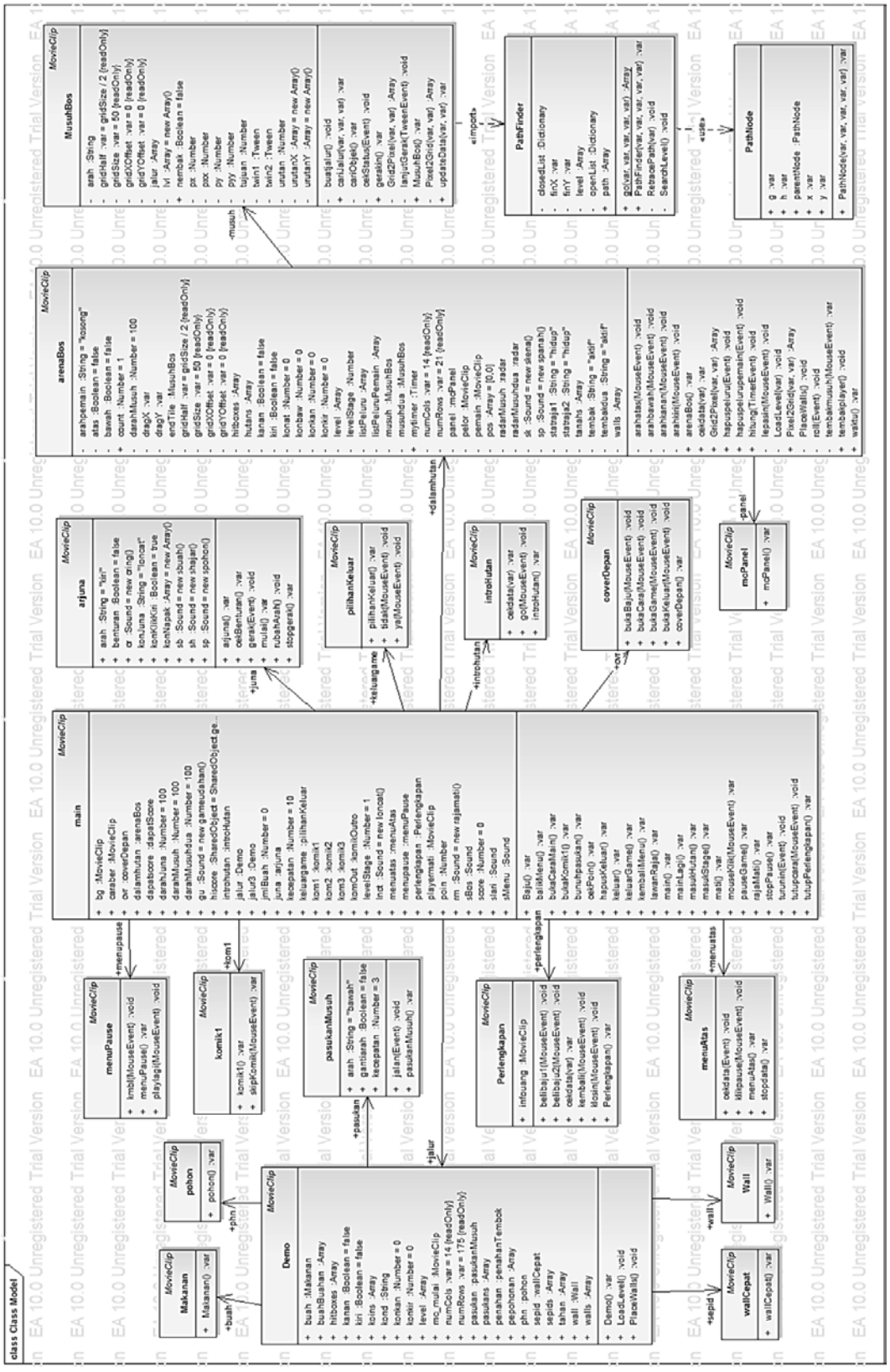
9. *Sequence Diagram* Keluar dari *game*



Gambar 3-27 *Sequence Diagram* Keluar dari *Game*

3.1.6.4 *Class Diagram*

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*. Hubungannya dari masing-masing *class* yang ada tersebut digambarkan seperti terlihat pada gambar 3-28.



Gambar 3-28 *Class* Diagram *Game Adventure* "Larilah Arjuna"

3.2 Perancangan Sistem

Perancangan sistem merupakan kegiatan untuk merancang aplikasi yang akan di bangun, tahapannya dimulai dari perancangan komponen *game*, perancangan antarmuka, perancangan pesan dan jaringan semantik dari *game adventure* Larilah Arjuna.

3.2.1 Peracangan Komponen *Game*

Di dalam bagian ini akan dijelaskan karakter-karakter dan *storyboard game adventure* Larilah Arjuna.

3.2.1.1 Karakter *Game*

Di dalam bagian ini akan dijelaskan karakter-karakter yang terdapat pada *game* Larilah Arjuna seperti yang bisa dilihat pada tabel 3-17.

Tabel 3-17 Karakter pada *Game* Larilah Arjuna

No	Karakter	Keterangan
1		Nama karakter : Arjuna, merupakan karakter utama dalam *game*
2		Nama karakter : Srikandi, Merupakan istri Arjuna yang akan membantu melawan bisma di level 2
3		Nama karakter: Prabu Malawapati, merupakan musuh Arjuna pada level 1
4		Nama karakter: Bisma, merupakan musuh yang akan dilawan arjuna dan srikandi pada level 2

No	Karakter	Keterangan
5		Nama karakter: Jayadarta, merupakan musuh yang akan dilawan arjuna pada level 3
6		Nama karakter: Karna, merupakan musuh yang akan dilawan arjuna pada level 3.

3.2.1.2 *Item Game*

Di dalam bagian ini akan dijelaskan *item-item* yang terdapat pada *game* Larilah Arjuna seperti yang bisa dilihat pada Tabel 3-18.

Tabel 3-18 Item pada *Game* Larilah Arjuna

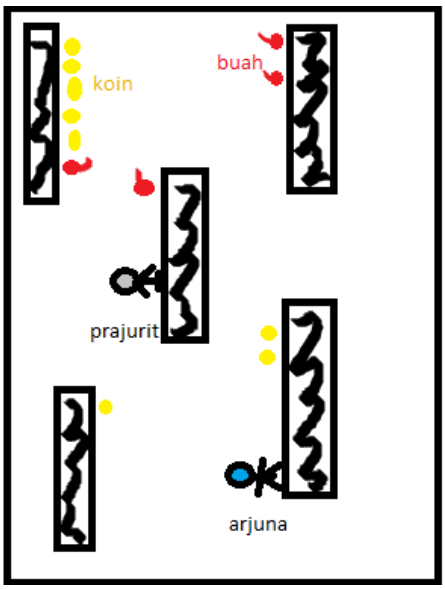
No	Karakter	Keterangan
1		Nama *item* : koin, merupakan koin yang akan diambil arjuna pada perjalanannya.
2		Nama *item* : Buah, merupakan item yang dapat menambah darah arjuna yang akan diambil pada perjalanannya.

3.2.1.3 *Storyboard*

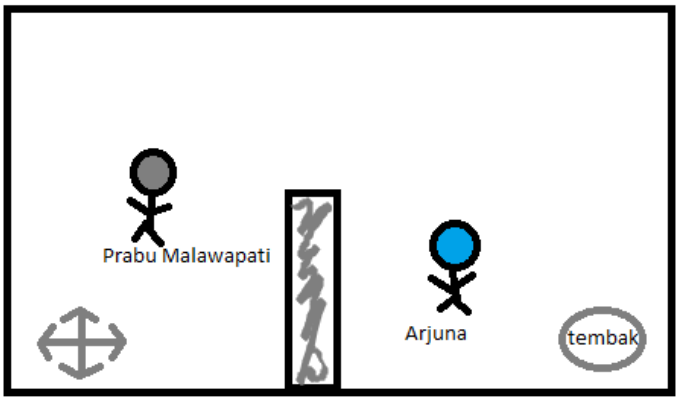
Storyboard adalah sketsa gambar yang disusun berurutan sesuai dengan naskah, dengan *storyboard* ide cerita dapat disampaikan kepada orang lain dengan lebih mudah, karena dapat menggiring khayalan seseorang mengikuti gambar-gambar yang tersaji sehingga menghasilkan persepsi yang sama. Berikut merupakan *storyboard* dari *game adventure* “Larilah Arjuna”.

1. *Storyboard* level 1

Cerita pada level 1 terjadi di hutan dimana arjuna dapat mengambil koin, buah dan membunuh prajurit dan juga akan melawan Prabu Malawapati. Gambar 3-29 adalah *storyboard* pada saat perjalanan arjuna menemui Prabu Malawapati dan pada gambar 3-30 pada saat di arena melawan Prabu Malawapati.



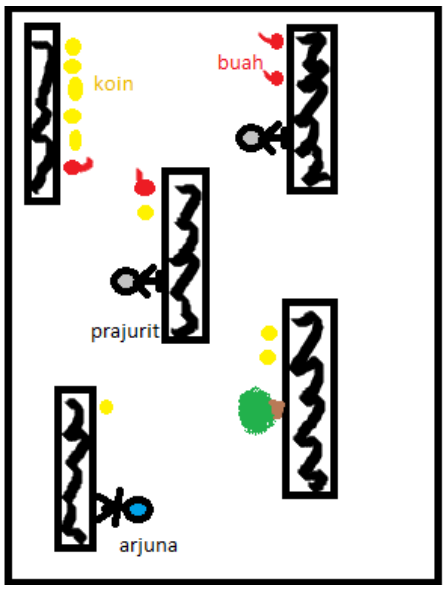
Gambar 3-29 *Storyboard* Level 1



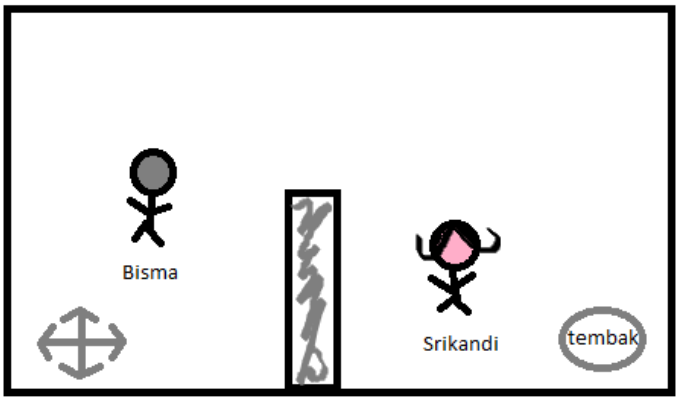
Gambar 3-30 *Storyboard* Melawan Prabu Malawapati

2. *Storyboard* level 2

Cerita pada level 2 terjadi di hutan dimana arjuna dapat mengambil koin, buah dan membunuh prajurit dan juga akan melawan Bisma. Gambar 3-31 adalah *storyboard* pada saat perjalanan arjuna menemui Bisma dan pada gambar 3-32 pada saat di arena melawan Bisma.



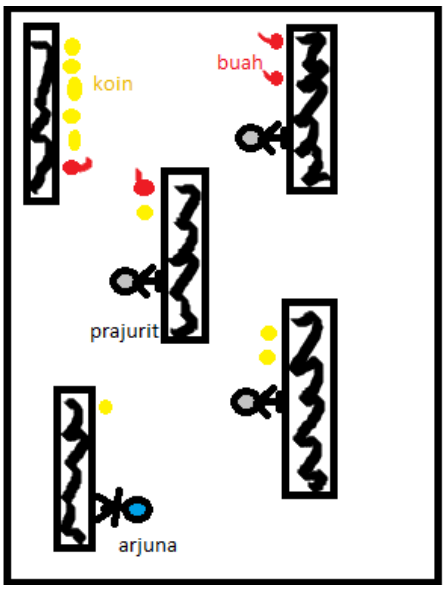
Gambar 3-31 *Storyboard* Level 2



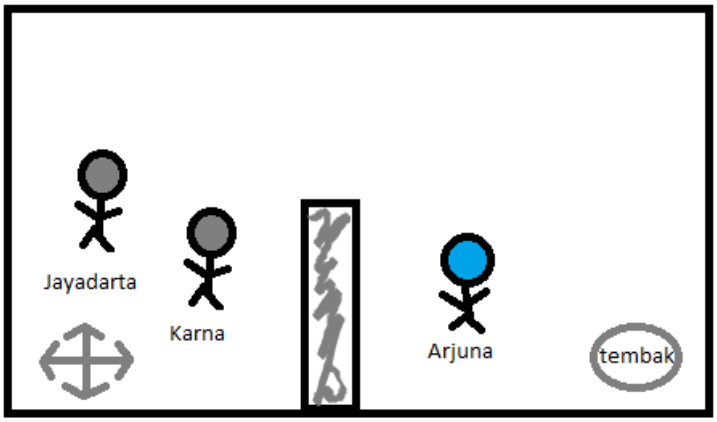
Gambar 3-32 *Storyboard* Melawan Bisma

3. *Storyboard* level 3

Cerita pada level 3 terjadi di desa dimana arjuna dapat mengambil koin, buah dan membunuh prajurit dan juga akan melawan Karna dan Jayadarta. Gambar 3-33 adalah *storyboard* pada saat perjalanan arjuna menemui Karna dan Jayadarta dan pada gambar 3-34 pada saat di arena melawan Karna dan Jayadarta.



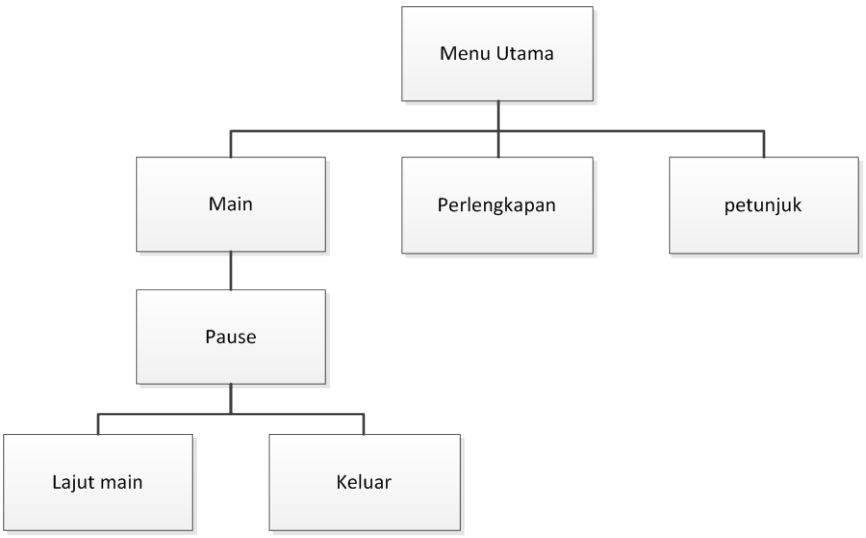
Gambar 3-33 *Storyboard* Level 3



Gambar 3-34 *Storyboard* Melawan Karna dan Jayadarta

3.2.2 Perancangan Menu

Menu digunakan oleh pemain (*user*) untuk berinteraksi dengan sistem. Munu yang dipilih akan menentukan respon dari sistem. Perancangan menu yang ada dalam *game* Larilah Arjuna dapat dilihat pada gambar 3-35.



Gambar 3-35 Perancangan Menu

a. Menu Main : Merupakan menu untuk memulai permainan
b. Menu Perlengkapan : Merupakan menu untuk menampilkan perlengkapan yang bisa dibei oleh pemain
c. Menu Petunjuk : Merupakan menu untuk menampilkan petunjuk bermain *game*
e. Menu Pause : Merupakan menu untuk mengisirahatkan permainan
f. Menu Lanjut main : Merupakan menu untuk melanjutkan bermain dari istirahat permainan
g. Menu Keluar : Merupakan menu untuk menghentikan permainan dan kembali ke menu utama

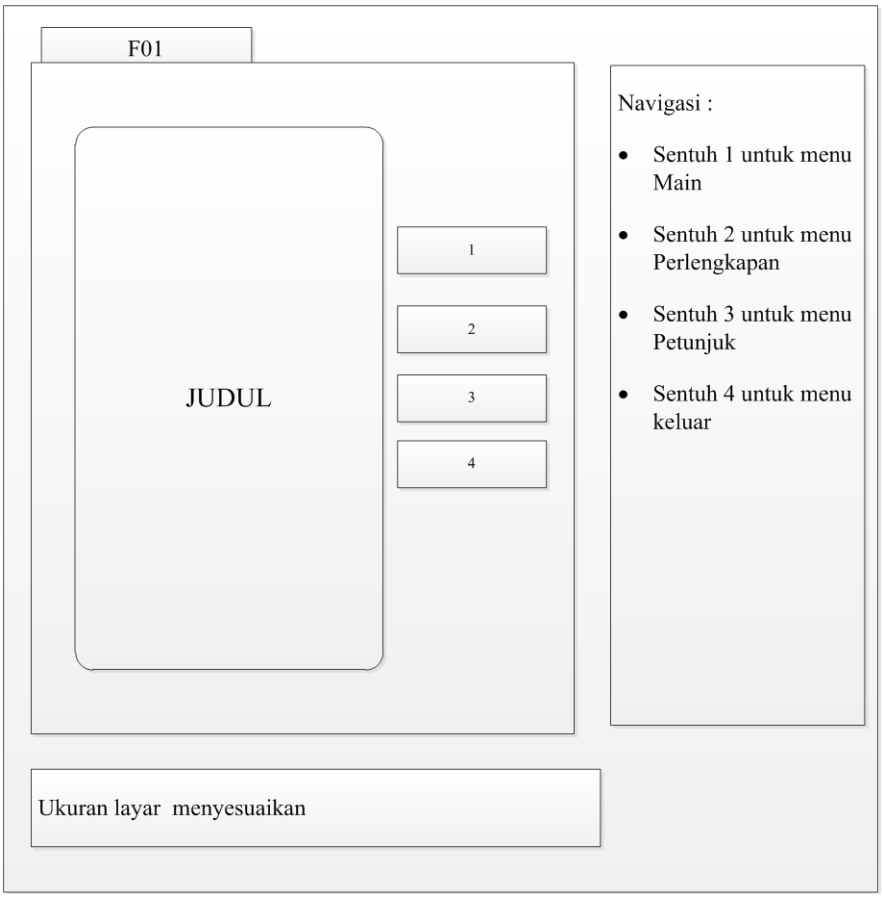
3.2.3 Perancangan Antarmuka

Perancangan antarmuka bertujuan untuk memberikan gambaran tentang aplikasi yang akan dibangun, sehingga akan mempermudah dalam mengimplementasikan aplikasi serta akan memudahkan pembuatan aplikasi.

Pada bagian ini akan dijelaskan perancangan antarmuka menu utama, perancangan antarmuka menu main, perancangan antarmuka menu perlengkapan, dan perancangan antarmuka menu petunjuk.

3.2.3.1 Perancangan Antarmuka Menu Utama

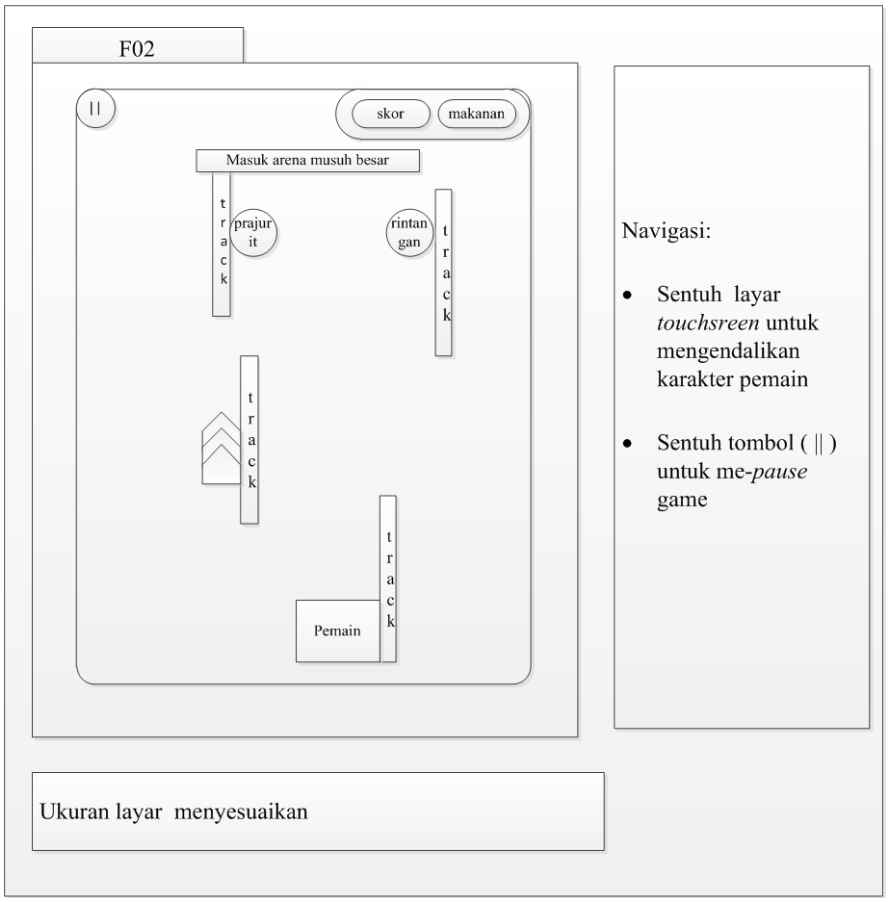
Menu utama merupakan menu yang akan muncul pertama kali pada saat pengguna membuka *game*. Gambar 3-36 menunjukan perancangan antarmuka menu utama.



Gambar 3-36 Perancangan Antarmuka Menu Utama

3.2.3.2 Perancangan Antamuka Menu Main

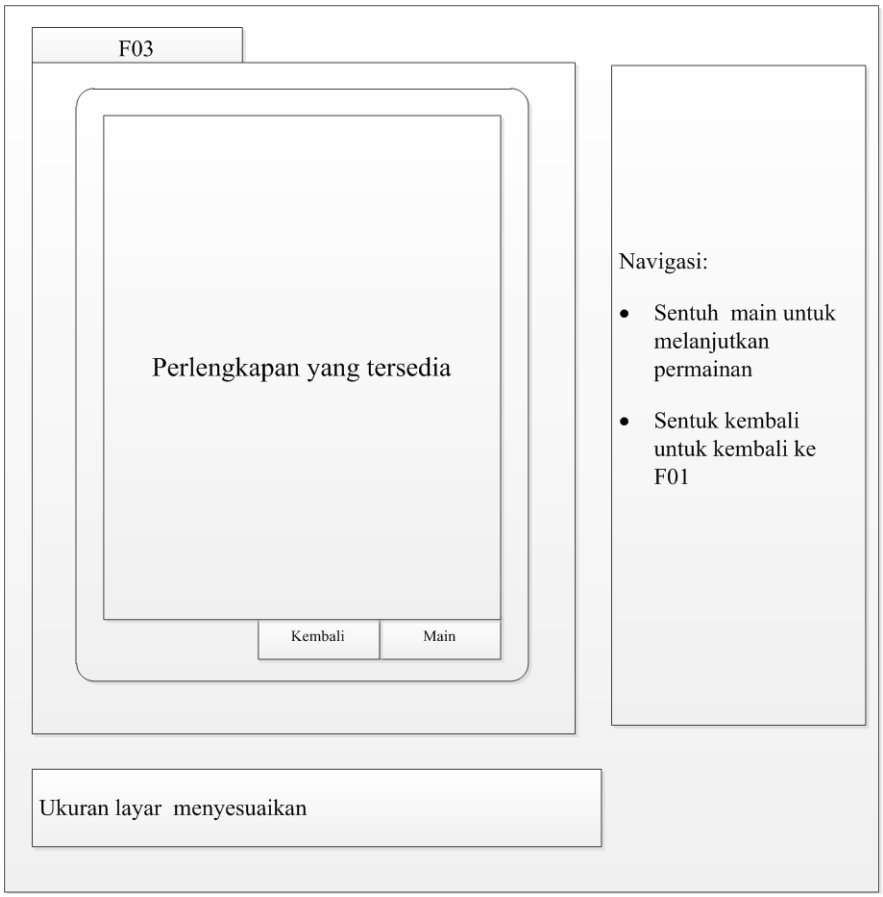
Antarmuka menu main merupakan perancangan antarmuka yang menampilkan permainan *game adventure* Larilah Arjuna. Gambar 3-37 merupakan perancangan antarmuka menu main.



Gambar 3-37 Perancangan Antarmuka Menu Main

3.2.3.3 Perancangan Antarmuka Menu Perlengkapan

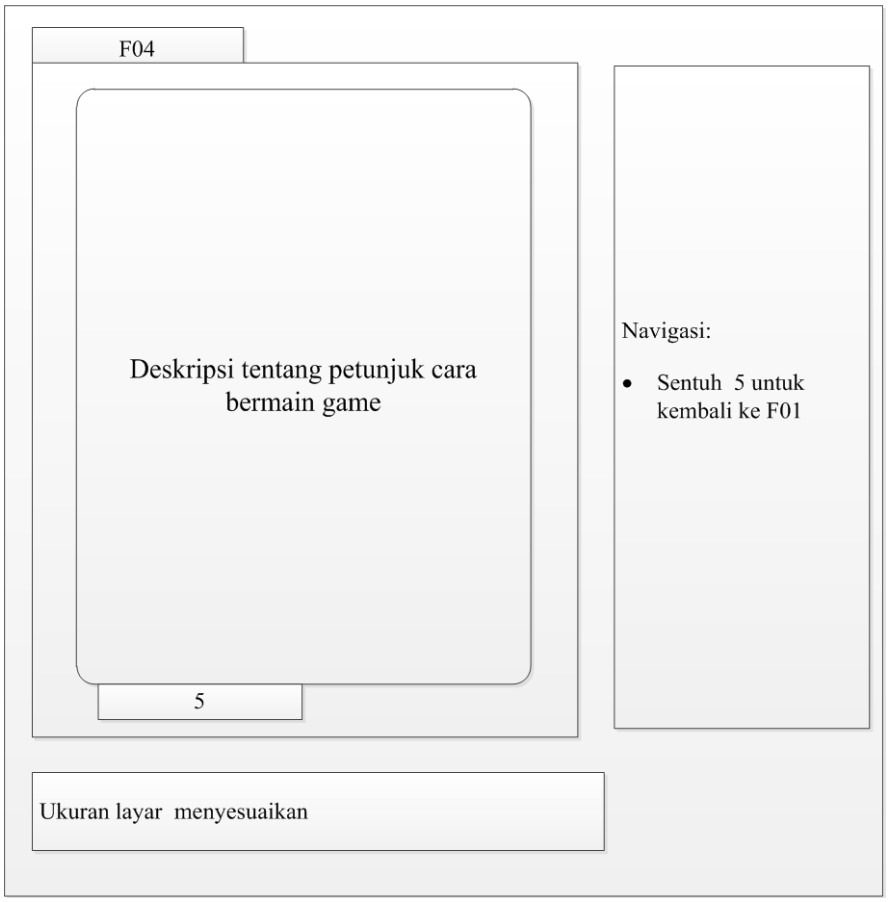
Menu perlengkapan merupakan menu menampilkan perlengkapan yang bisa dibeli oleh pemain dalam *game adventure* Larilah Arjuna. Gambar 3-38 merupakan perancangan antarmuka menu perlengkapan.



Gambar 3-38 Perancangan Antarmuka Menu Perlengkapan

3.2.3.4 Perancangan Antarmuka Menu Petunjuk

Menu petunjuk merupakan menu yang menampilkan petunjuk cara bermain dalam *game*. Gambar 3-39 merupakan perancangan antarmuka menu petunjuk.



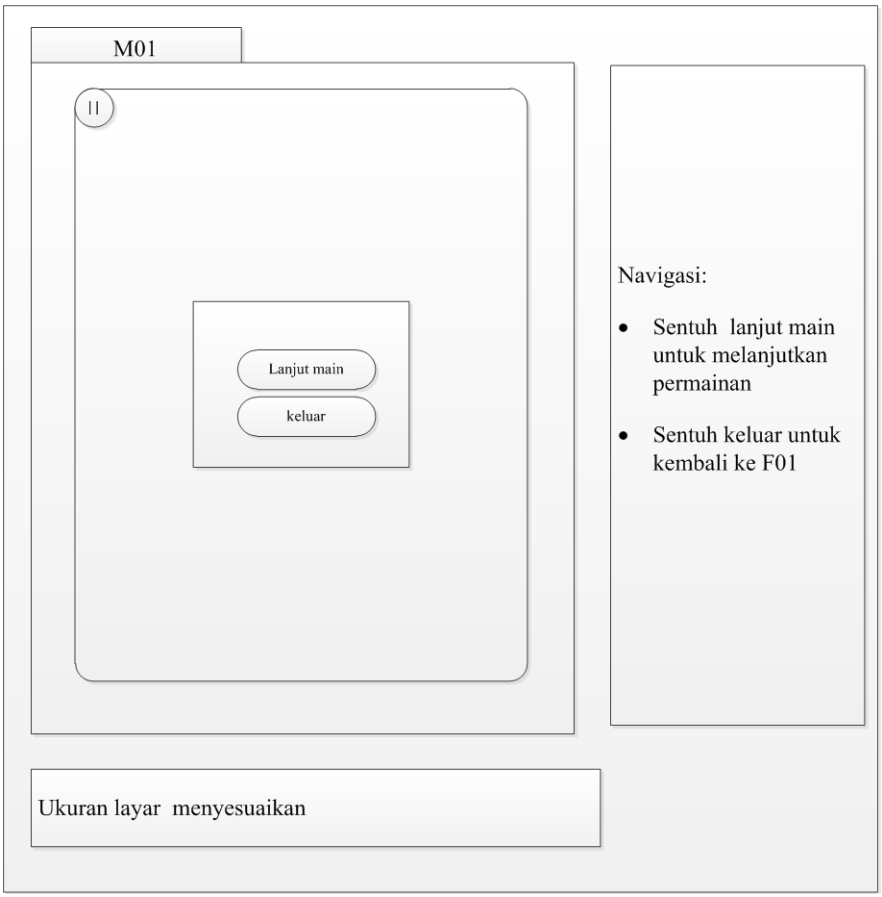
Gambar 3-39 Perancangan Antarmuka Menu Petunjuk

3.2.4 Perancangan Pesan

Pada bagian ini akan dijelaskan perancangan pesan yang terdapat pada *game adventure* "larilah Arjuna" yaitu perancangan pesan *pause menu*, perancangan pesan perlengkapan dan perancangan pesan keluar *game*.

3.2.4.1 Perancangan Pesan Pause Menu

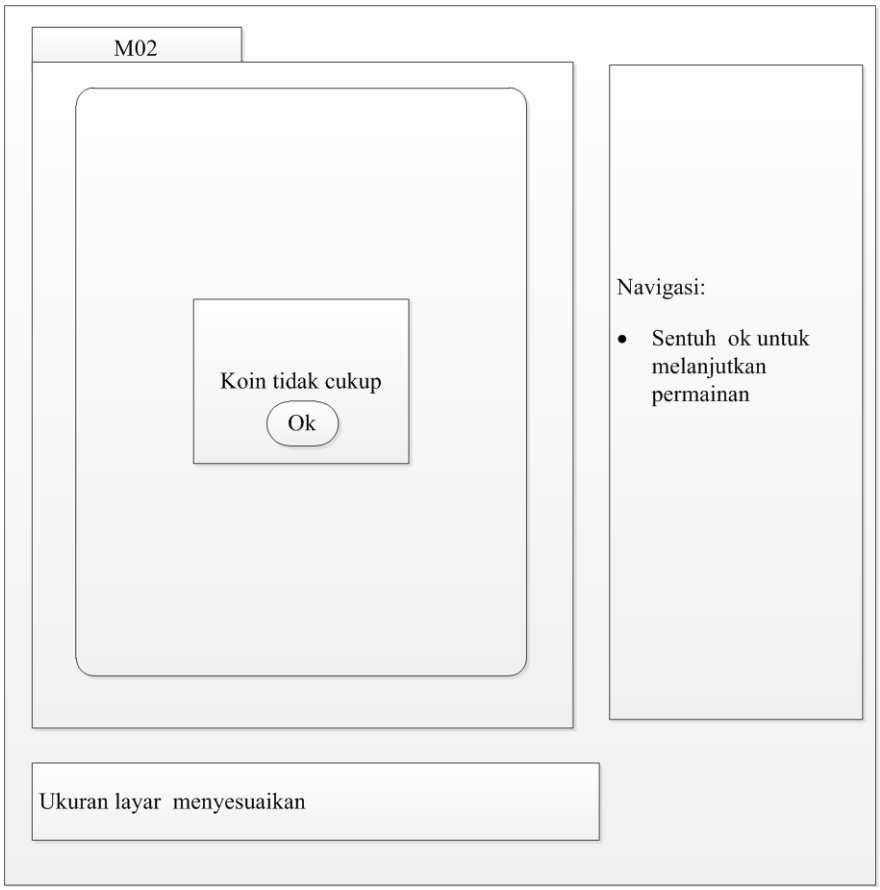
Pesan pause menu merupakan pesan yang muncul ketika pemain menampilkan pause menu. Perancangan pesan pause lebih jelasnya dapat dilihat pada gambar 3-40.



Gambar 3-40 Perancangan Pesan Pause Menu

3.2.4.2 Perancangan Pesan Perlengkapan

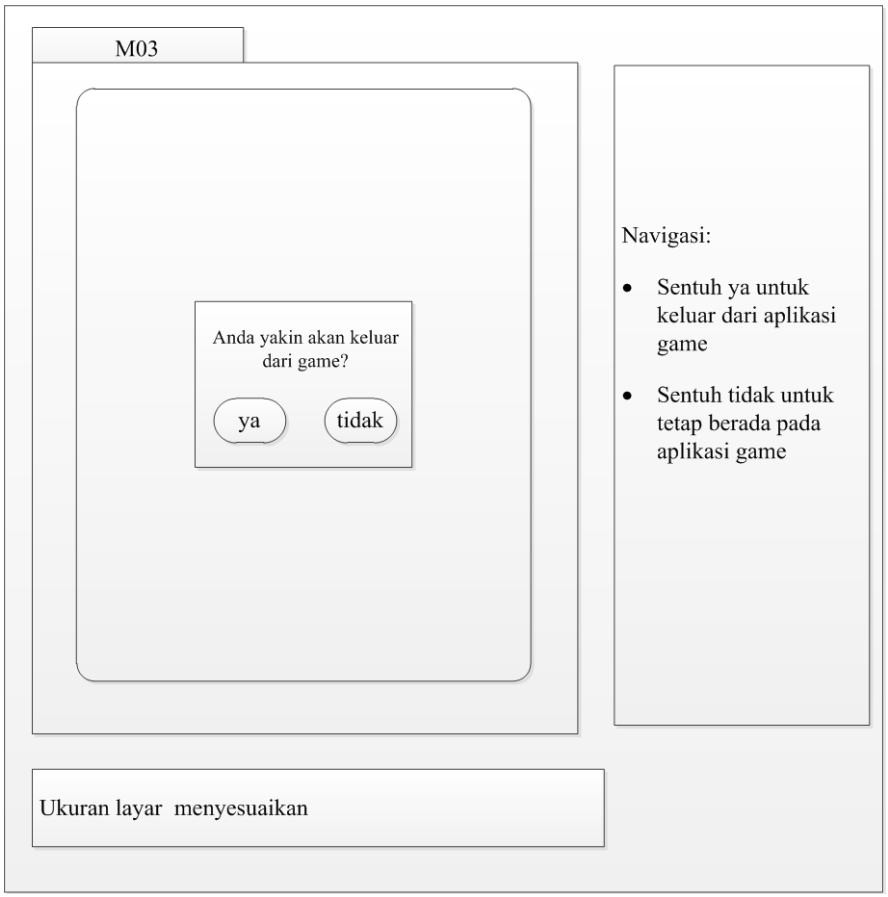
Pesan perlengkapan merupakan pesan yang muncul ketika pengguna membeli perlengkapan. Perancangan pesan perlengkapan lebih jelasnya dapat dilihat pada gambar 4-7.



Gambar 3-41 Perancangan Pesan Perlengkapan

3.2.4.3 Perancangan Pesan Keluar *Game*

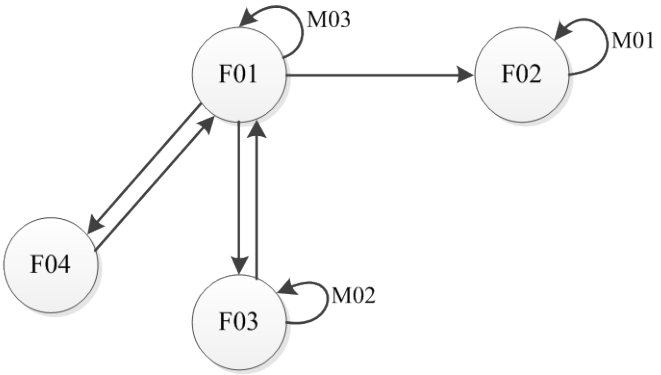
Pesan perlengkapan merupakan pesan yang muncul ketika pengguna akan keluar dari aplikasi *game*. Peracangan pesan keluar lebih jelasnya dapat dilihat pada gambar 3-42.



Gambar 3-42 Perancangan Pesan keluar *Game*

3.2.5 Jaringan Semantik

Untuk mengetahui keterhubungan dari setiap tampilan yang dibuat dapat menggunakan jaringan semantik. Gambar 3-43 adalah jaringan semantik pada *game adventure* Larilah Arjuna.



Gambar 3-43 Jaringan Semantik