BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem ini merupakan cakupan dari analisis yang berhubungan dengan *game* yang akan dibangun diantaranya analisis masalah terhadap *game* yang sejenis, analisis kebutuhan fungsional, dan analisis non fungsional.

3.2 Analisis Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, maka masalahnya adalah *game arcade* yang ada saat ini adalah dalam *gameplay*nya, hanya menembaki musuh tanpa memperlihatkan karakternya dan hanya melewati beberapa objek yang tidak bergerak sehingga *gameplay* kurang variatif. Selain itu, pada *game arcade* masih jarang mengimplementasikan cerita rakyat yang mengambil cerita dari Indonesia. Adapun masalah tersebut dapat dirumuskan dalam pernyataan sebagai berikut:

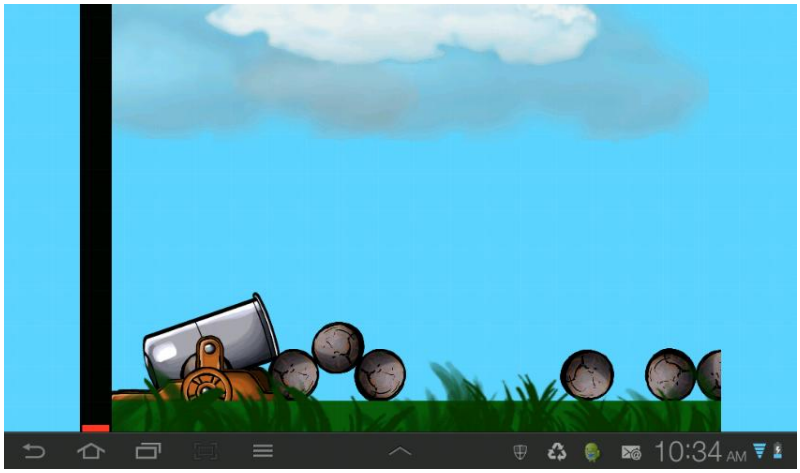
1) Masih kurangnya penyampaian kisah cerita Si Pitung kepada *player*.
2) Kurangnya jelasnya sosok tokoh Si Pitung yang dimunculkan dalam game sejenis.

3.3 Analisis *Game* Sejenis

Analisis yang dilakukan merupakan observasi dari *Game* sejenis untuk melihat cara bermain, tujuan dan ada tidaknya kekurangan atau kelebihan *Game* itu sendiri. Hasil observasi tersebut untuk membantu dalam penelitian. *Game* sejenis yang dianalisis yaitu *Meriam si Pitung.*

3.3.1 Meriam si Pitung

Meriam si pitung adalah sebuah *game arcade* berbasis android. *Game* ini hanya memiliki karakter meriam yang dapat dijalankan pemain, karena pada dasarnya *game* ini hanya menampilkan meriam dan musuh yang berdatangan sangat banyak. *Game* ini memiliki banyak level dengan kesulitan yang berbeda-beda.



Gambar 3.1 *Game* Meriam si Pitung

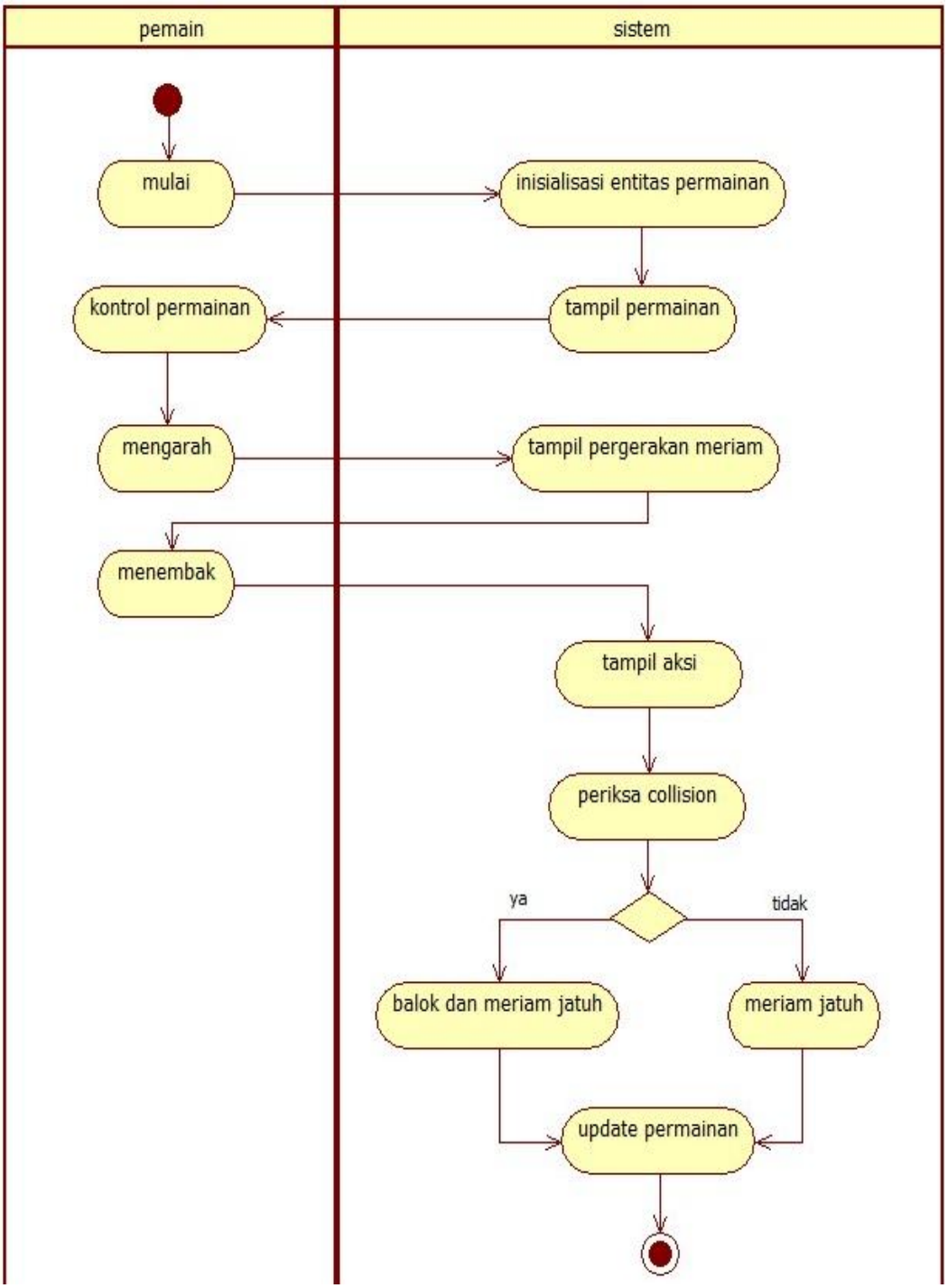
3.3.1.1 Alur Permainan Meriam Si Pitung

Alur permainan meriam si pitung merupakan sebuah alur dari aktivitas *game*, dimana alur tersebut dapat menggambarkan aktivitas yang berada dalam *game*. Penjelasan dari alur permainan Meriam si Pitung akan dijelaskan dalam table 3.1 berikut.

Tabel 3. 1 Alur Permainan Meriam Si Pitung

Alur Permainan
1. Pemain harus mengarahkan meriam untuk menembakan meriam ke balok. 2. Pemain harus menembakan meriam dengan jangkauan tertentu. 3. Pemain menggunakan meriam untuk menjatuh balok. 4. Pemain akan mendapatkan point jika dapat menjatuhkan semua balok. 5. Pemain akan kalah jika semua kesempatan menembak digunakan namun tidak mengenai balok. 6. Permainan ini tidak mempunyai level, hanya mengulang kembali permainan.

Adapun alur permainan dari *game* Meriam si pitung dapat dilihat pada gambar 3.1 berikut :



Gambar 3.2 alur permainan Meriam Si Pitung

Berdasarkan analisis yang dilakukan terhadap *game* meriam si pitung, dapat disimpulkan bahwa dalam *game* tersebut kurang sesuai dengan cerita rakyat si pitung sehingga kurang mengenalkan tokoh pitung. Selain itu kekurangan

dalam game ini meliputi beberapa hal yaitu, tidak adanya leveling dalam game, tidak memiliki *storyline*, tidak dimunculkannya karakter si pitung dalam game, dan tidak ada perbedaan susunan balok ketika permainan berulang.

3.4 Analisis *Game* yang akan dibangun

Analisis *Game* yang akan dibangun merupakan bagian yang mendeskripsikan *Game* yang akan dibangun. Pada bagian ini analisis terdiri dari arsitektur, *storyline*, tingkat kesulitan, *Gameplay*.

3.4.1 Pengenalan

Game yang akan dibangun berjudul si pitung yang bergenre *arcade*. *Game* ini dibangun untuk memperkenalkan tokoh si pitung dan mengintrepretasikannya ke dalam salah satu game yang sudah ada sejak lama kepada anak-anak masa kini.

Batasan-batasan pada game si pitung ini dijelaskan pada table 3.3 berikut.

Tabel 3.2 Batasan *Game*

Batasan *Game*
1. *Game* yang dibuat berbasis desktop dengan tampilan jenis 2 Dimensi. 2. *Single player* 3. Karakter dapat melawan musuh dengan tangan kosong dan menggunakan senjata. 4. Terdapat bermacam-macam musuh. 5. Terdapat life point yang dapat member kesempatan pemain dan member kesulitan pada musuh. 6. Terdapat beberapa warga yang harus diselamatkan.

3.4.2 *Storyline*

Game si Pitung menceritakan tentang seorang warga asli betawi yang membela warga yang selalu ditindas oleh para menir belanda. Pada awal cerita, si pitung sudah beberapa tahun belajar di sebuah pesantren. Suatu ketika dia telah lulus belajar dari pesantren tersebut, tidak lama si pitung pun berencana untuk pulang ke kampung halamannya. Sesampainya di kampong si pitung terkejut akan kejadian di kampong nya dimana para menir belanda selalu menyiksa warga-

warga dikampungnya dengan cara kerja paksa, di pajak dan lainnya. Si pitung pun tidak tinggal diam, dengan kemampuannya, dia berencana untuk melepaskan warga yang menjadi tawanan para menir belanda. Akan tetapi si pitung harus melalui penjagaan yang ketat dari prajurit-prajurit belanda[1].

3.4.3 *Gameplay*

Gameplay dilakukan untuk menggambarkan aturan-aturan dalam *Game*. *Gameplay* Si Pitung yang akan dibangun dijelaskan pada table 3.4 berikut.

Tabel 3. 3 *Gameplay* Si Pitung

Gameplay
1. Pada awal permainan pemain harus mengalahkan musuh dengan cara memukul. 2. Pemain menghindari serangan musuh dengan cara melompat 3. Jika lifepoin habis maka permainan akan selesai dan akan mengulang ke awal level tersebut. 4. Pemain akan berjumpa dengan prajurit1 yang masih lemah dan mampu dibunuh dengan pukulan. Level 1 pertempuran terjadi pada siang hari, selain itu pemain harus membunuh semua prajurit1. 5. Di level 2 ditambah prajurit lagi dimana pemain harus membunuh prajurit1 dan prajurit2, namun tingkat kesulitan pun berbeda dengan level 1 karena di level 2 prajuritnya bisa menembak dan memukul. 6. Level ini merupakan level akhir pada game petualangan Si Pitung. Misinya pun sama seperti level 1 dan level 2, namun pada level akhir ini akan mendapatkan senjata untuk membantu menghadapi prajurit dan 1 sosok Menir(boss) sekaligus yang harus dikalahkan oelh pemain. 7. Pemain harus menyelamatkan warga untuk dapat melanjutkan permainan ke level selanjutnya.

3.4.4 Misi

Misi yang harus diselesaikan yaitu dengan melakukan penyelamatan warga miskin dan mengalahkan prajurit kompeni hingga menir belanda melepaskan semua tawan tersebut. Di level kedua dan terakhir pemain harus menemukan item tertentu untuk dapat melewati dan mengalahkan prajurit kompeni tersebut.

3.4.5 Sasaran Pemain dan Kegunaan *Game*

Sasaran pemain menentukan layak tidaknya pemain memainkan permainan ini. Berikut adalah sasaran pemain yang layak memainkan permainan ini :

1. Berdasarkan umur pengguna yaitu 9 tahun ke atas.
2. Dapat mempergunakan komputer, karena *Game* berbentuk aplikasi java.

Kegunaan *Game* menunjukan kegunaan *Game* untuk pemain. Kegunaan dalam *Game* ini untuk memperkenalkan cerita rakyat dari betawi, sehingga pengguna dengan umur muda dapat bermain dan mengenal tokoh si pitung dari menyelesaikan permainan ini.

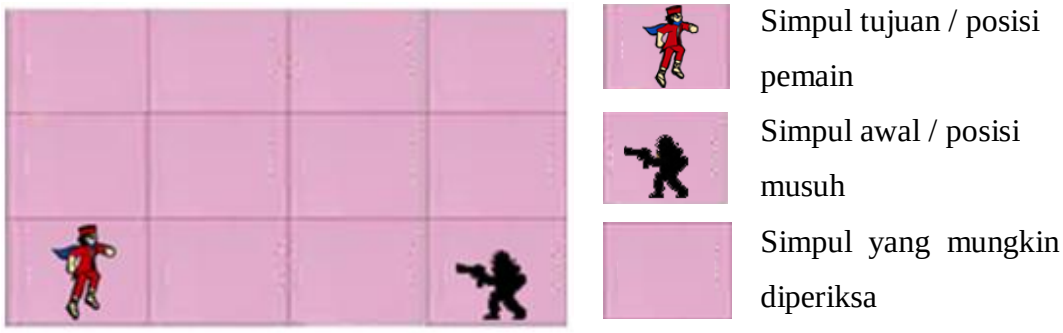
3.5 Analisis Algoritma

Analisis algoritma yang dilakukan dalam penelitian ini adalah meneliti bagaimana cara kerja algoritma *A** pada prajurit belanda dan menir di dalam *Game*. Kondisi dimana musuh dapat mencari dan menyerang pemain ketika muncul.

3.5.1 Analisis Algoritma *A** pada NPC

Arena permainan pada *game* si pitung terdiri dari ordo X dan Y, dimana dalam analisis ini akan dijelaskan dengan ordo minimal 3x4 dikarenakan perhitungan pada ordo berapa pun akan sama. Penentuan bobot setiap node akan diberikan nilai sesuai dengan jarak ke tujuan, misalnya node yang terjauh dari tujuan maka diberi bobot yang kecil sedangkan node yang terdekat dengan tujuan diberi bobot yang lebih besar. Pada kasus pencarian objek di arena permainanakan terdapat dua kondisi yaitu, kondisi pertama terjadi ketika pencarian yang dilakukan oleh musuh tidak terdapat penghalang, sedangkan kondisi kedua terjadi ketika pencarian yang dilakukan terdapat penghalang. Oleh karena itu, pada analisis ini akan dilakukan jenis perhitungan menggunakan algoritma A* tanpa penghalang.

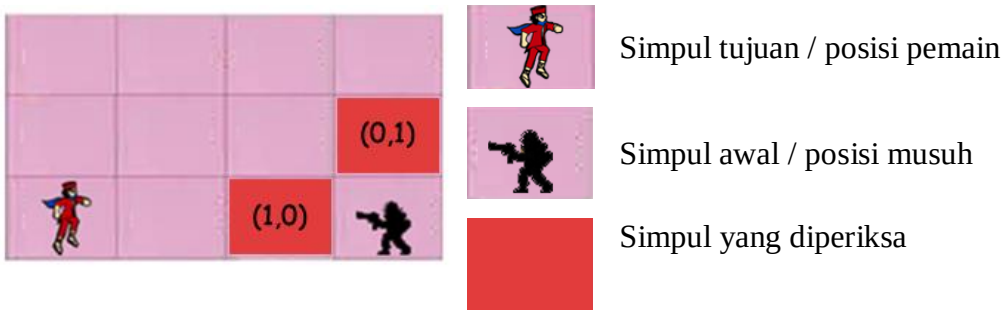
Pada kondisi awal, simpul awal atau posisi musuh dicontohkan berada pada koordinat (1,0) sedangkan simpul tujuan simpul tujuan atau posisi karakter pemain berada pada koordinat (3,0) dalam arena permainan.



Gambar 3.3 kondisi awal pencarian A*

Langkah selanjutnya akan dilakukan pemeriksaan terhadap simpul-simpul yang bertetangga yang memiliki nilai *cost* / biaya terkecil sebagai *Best Node* sebagai rute yang akan dipilih peluru, dengan melakukan langkah perhitungan sebagai berikut:

Pada langkah pertama akan diperiksa simpul pada koordinat n(1,0) dan koordinat n(0,1).



Gambar 3.4 Penelusuran pada koordinat n(1,0), n(0,1)

a. Koordinat n(1,0)

Nilai n.x = 1

Nilai n.y = 0

Simpul tujuan(3,0) sehingga goal.x = 3, goal.y = 0

g (1,0) = 1

h_orthogonal(n) = (abs(n.x-goal.x) + abs(n.y-goal.y))

h_orthogonal(1,0) = (abs(1 - 3) + abs(0 - 0))

= (abs(-2) + abs(0))

= 2

h_diagonal(n) = min(abs(n.x-goal.x) + abs(n.y-goal.y))

h_diagonal(1,0) = min(abs(1 - 3)+abs(0 - 0))

= min(abs(-2)+abs(0))

= min 2

h(n) = h_diagonal(n) + (h_orthogonal (n) – (2 * h_diagonal(n)))

h(1,0) = (- 2) + (2-(2*(-2)))

= -2 + 6

= 4

f (1,0) = g (1,0) + h (1,0)

= 1 + 4

= 5

b. Koordinat n(0,1)

Nilai n.x = 0

Nilai n.y = 1

Simpul tujuan(3,0) sehingga goal.x = 3, goal.y = 0

g(n) = g (0,1) = 1

h_orthogonal(n) = (abs(n.x-goal.x) + abs(n.y-goal.y))

h_orthogonal(0,1) = (abs(0 - 2) + abs(1 - 0))

= (abs(-2) + abs(1))

= 3

h_diagonal(n) = min(abs(n.x-goal.x) + abs(n.y-goal.y))

h_diagonal(0,1) = min(abs(0 - 2)+abs(1 - 0))

= min(abs(-2)+abs(1))

= min 3

h(n) = h_diagonal(n) + (h_orthogonal (n) – (2 * h_diagonal(n)))

h(0,1) = (-3) + (3-(2*(-3)))

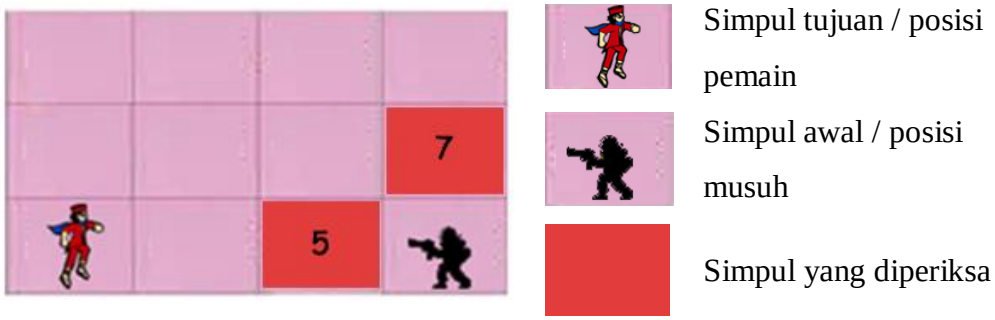
= -3 + 9

= 6

f (0,1) = g (0,1) + h (0,1)

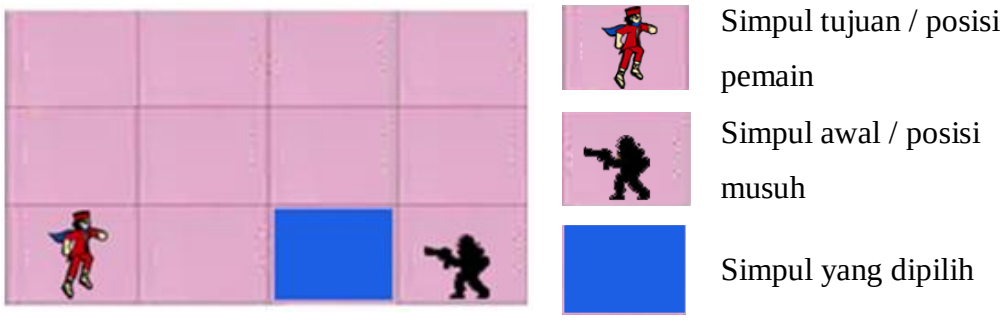
= 1 + 6

= 7



Gambar 3.5 Pencarian *BestNode* pertama

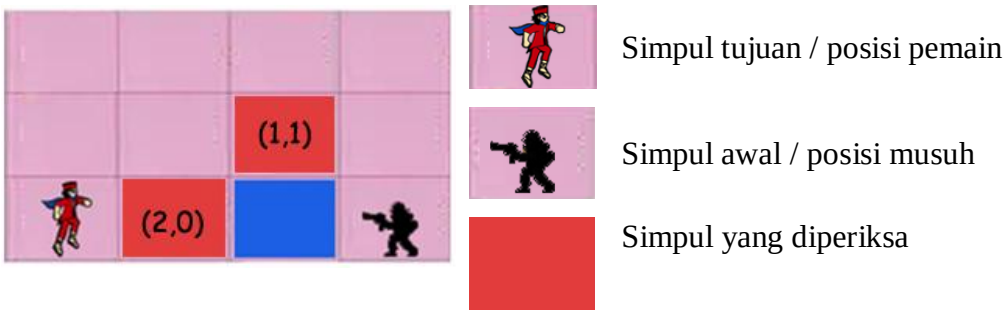
Dari hasil perhitungan diatas seperti terlihat pada gambar 3.7 terdapat tiga simpul yang mungkin menjadi *BestNode* yaitu (1,0) dengan f(n)=5 dan (0,1) dengan f(n)=7. Dari ke dua simpul yang mungkin maka dipilihlah simpul (1,0) sebagai *Best Node* karena memiliki biaya terkecil yaitu 5.



Gambar 3.6 Simpul yang terpilih sebagai *BestNode*

Langkah selanjutnya akan dilakukan pemeriksaan terhadap simpul-simpul yang bertetangga yang memiliki nilai *cost* / biaya terkecil sebagai *Best Node* sebagai rute yang akan dipilih peluru, dengan melakukan langkah perhitungan sebagai berikut:

Pada langkah pertama akan diperiksa simpul pada koordinat n(2,0) dan koordinat n(1,1).



Gambar 3.7 Penelusuran kedua pada koordinat n(2,0), n(1,1)

c. Koordinat n(2,0)

Nilai n.x = 2

Nilai n.y = 0

Simpul tujuan(3,0) sehingga goal.x = 3, goal.y = 0

g (2,0) = 2

h_orthogonal(n) = (abs(n.x-goal.x) + abs(n.y-goal.y))

h_orthogonal(2,0) = (abs(2 - 3) + abs(0 - 0))

= (abs(-1) + abs(0))

= 1

h_diagonal(n) = min(abs(n.x-goal.x) + abs(n.y-goal.y))

h_diagonal(2,0) = min(abs(2 - 3)+abs(0 - 0))

= min(abs(-1)+abs(0))

= min 1

h(n) = h_diagonal(n) + (h_orthogonal (n) – (2 * h_diagonal(n)))

h(2,0) = (- 1) + (1-(2*(-1)))

= -1 + 3

= 2

f (2,0) = g (2,0) + h (2,0)

= 2 + 2

= 4

d. Koordinat n(1,1)

Nilai n.x = 1

Nilai n.y = 1

Simpul tujuan(3,0) sehingga goal.x = 3, goal.y = 0

g(n) = g (1,1) = 2

h_orthogonal(n) = (abs(n.x-goal.x) + abs(n.y-goal.y))

h_orthogonal(1,1) = (abs(1 - 2) + abs(1 - 0))

= (abs(-1) + abs(1))

= 2

h_diagonal(n) = min(abs(n.x-goal.x) + abs(n.y-goal.y))

h_diagonal(1,1) = min(abs(1 - 2)+abs(1 - 0))

= min(abs(-1)+abs(1))

= min 2

h(n) = h_diagonal(n) + (h_orthogonal (n) – (2 * h_diagonal(n)))

h(1,1) = (-2) + (2-(2*(-2)))

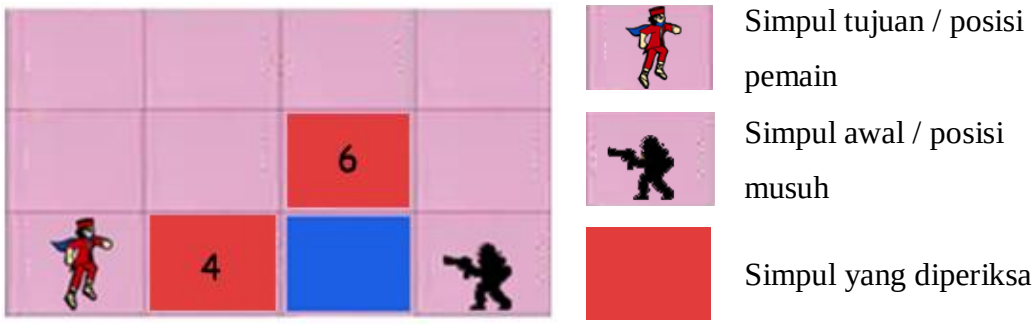
= -2 + 6

= 4

f (1,1) = g (1,1) + h (1,1)

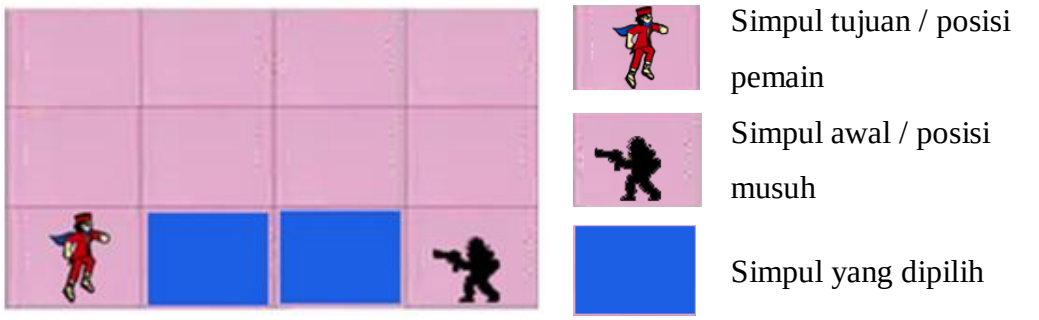
= 2 + 4

= 6



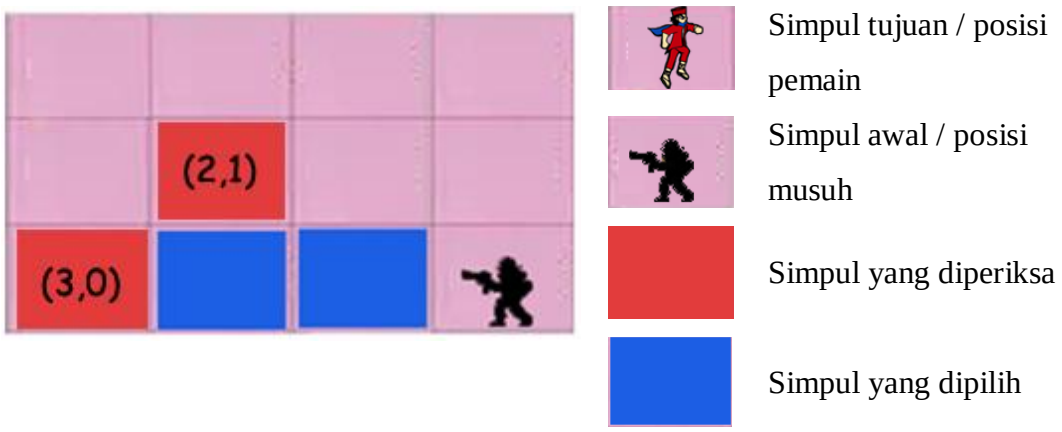
Gambar 3.8 Pencarian *BestNode* kedua

Dari hasil perhitungan diatas seperti terlihat pada gambar 3.17 terdapat tiga simpul yang mungkin menjadi *BestNode* yaitu (2,0) dengan f(n)=3 dan (0,0) dengan f(n)=5. Dari ke dua simpul yang mungkin maka dipilihlah simpul (2,0) sebagai *Best Node* karena memiliki biaya terkecil yaitu 3.



Gambar 3.9 Simpul yang terpilih sebagai *BestNode*

Setelah mendapatkan *BestNode* dilangkah pertama, dilakukan penelusuran lanjutan untuk mendapatkan *BestNode* selanjutnya dengan melakukan penelusuran simpul – simpul yang memiliki nilai *cost* terkecil. Seperti pada langkah pertama dilakukan lagi penelusuran pada koordinat n(3,0) dan n(2,1).



Gambar 3.10 penelusuran ketiga pada koordinat n(3,0), n(2,1)

e. Koordinat n(3,0)

Nilai n.x = 3

Nilai n.y = 0

Simpul tujuan(3,0) sehingga goal.x = 3, goal.y = 0

g(n) = g (3,0) = 3

h_orthogonal(n) = (abs(n.x-goal.x) + abs(n.y-goal.y))

h_orthogonal(3,0) = (abs(3 - 3) + abs(0 - 0))

= (abs(0) + abs(0))

= 0

h_diagonal(n) = min(abs(n.x-goal.x) + abs(n.y-goal.y))

h_diagonal(3,0) = min(abs(3 - 3)+abs(0 - 0))

= min(abs(0)+abs(0))

= min 0

h(n) = h_diagonal(n) + (h_orthogonal (n) – (2 * h_diagonal(n)))

h(3,0) = (-0) + (0-(2*(-0)))

= 0 + 0

= 0

f (3,0) = g (3,0) + h (3,0)

= 3 + 0

= 3

f. Koordinat n(2,1)

Nilai n.x = 2

Nilai n.y = 1

Simpul tujuan(3,0) sehingga goal.x = 3, goal.y = 0

g(n) = g (3,0) = 3

h_orthogonal(n) = (abs(n.x-goal.x) + abs(n.y-goal.y))

h_orthogonal(2,1) = (abs(2 - 3) + abs(1 - 0))

= (abs(-1) + abs(1))

= 2

h_diagonal(n) = min(abs(n.x-goal.x) + abs(n.y-goal.y))

h_diagonal(2,1) = min(abs(2 - 3)+abs(1 - 0))

= min(abs(-1)+abs(1))

= min 2

h(n) = h_diagonal(n) + (h_orthogonal (n) – (2 * h_diagonal(n)))

h(2,1) = (-2) + (2-(2*(-2)))

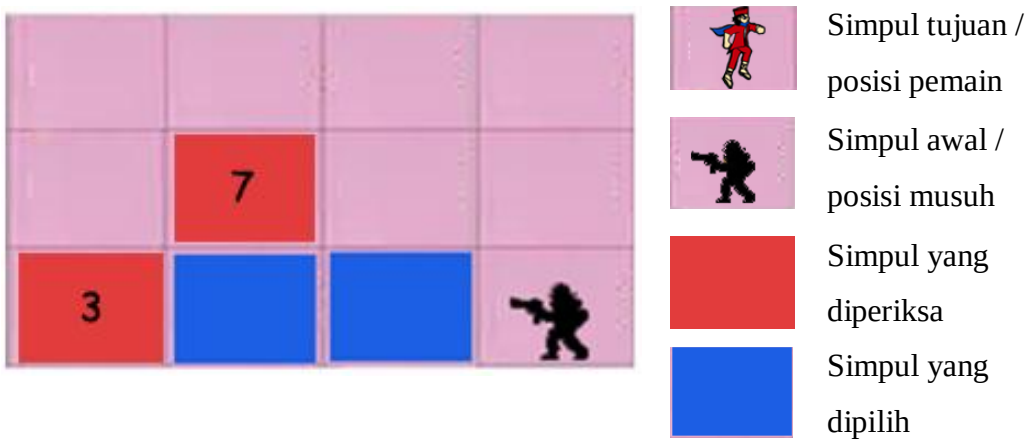
= -2 + 6

= 4

f (2,1) = g (2,1) + h (2,1)

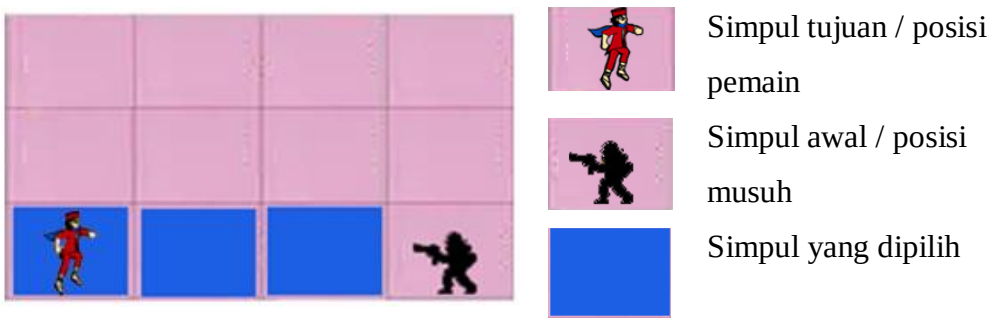
= 3 + 4

= 7



Gambar 3.11 Pencarian *BestNode* ketiga

Pada gambar 3.10 terdapat tiga simpul yang mungkin menjadi *BestNode* yaitu koordinat n(3,0) dengan f(n)=2. Koordinat n(3,0) dipilih sebagai *Best node* karena memiliki biaya *cost* terkecil dan koordinat n(3,0) dikenali sebagai simpul tujuan yang berarti solusi untuk menemukan rute terpendek menuju posisi pemain.



Gambar 3.12 Hasil pencarian jalur dengan Algoritma A*

Dari semua perhitungan yang telah dilakukan di pilihlah biaya/*cost* terkecil pada setiap langkahnya sehingga akan menghasilkan jalur terpendek yang akan dilalui musuh untuk melakukan pencarian posisi pemain seperti terlihat dari gambar sebelumnya.

3.6 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional dilakukan untuk mengetahui spesifikasi kebutuhan untuk sistem.Spesifikasi kebutuhan melibatkan analisis perangkat keras/*hardware*, analisis perangkat lunak/*software*, analisis perangkat pikir/*user*.

3.6.1 Analisis Kebutuhan Perangkat Keras

Analisis kebutuhan perangkat keras digunakan untuk mengetahui spesifikasi minimun perangkat keras dalam membangun *Game* si pitung akan dijelaskan pada tabel 3.4 berikut.

Tabel 3.4 Spesifikasi Perangkat Keras Pembangun

Nama Perangkat	Spesifikasi
Processor	1.80Ghz
RAM	4GB
Harddisk	50 GB
Display	Resolusi 1366 x 768
Keyboard	Standar
Mouse	Standar

Selain itu, untuk mengetahui spesifikasi minimum perangkat keras dalam menggunakan Game si pitung akan dijelaskan pada tabel 3.5 berikut.

Tabel 3.5 Spesifikasi Perangkat Keras Pengguna

Nama Perangkat	Spesifikasi
Processor	1.80Ghz
RAM	512MB
Harddisk	200MB
Display	Resolusi 800 x 600
Keyboard	Standar
Mouse	Standar

3.6.2 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak digunakan dalam sebuah sistem merupakan perintah yang diberikan kepada perangkat keras agar dapat saling berinteraksi diantaranya. Perangkat lunak yang dibutuhkan untuk pembangunan *Game* si pitung ini dijelaskan pada tabel 3.6 berikut.

Tabel 3.7 Spesifikasi Kebutuhan Perangkat Lunak Pembangun

Sistem operasi	Windows 7 *Ultimate* 32-bit
Compiler	Greenfoot
Development toolkit	Java
Design tools	Adobe Photoshop CS 3
Design analysis	StarUML

Selain itu, perangkat lunak yang dibutuhkan untuk pengguna Game si pitung dijelaskan pada tabel 3.7 berikut.

Tabel 3.8 Spesifikasi Kebutuhan Perangkat Lunak Pengguna

Operating System	Windows 7 *Ultimate* 32-bit
Tools	*Java vitual machine*

3.6.3 Analisis Kebutuhan Perangkat Pikir

Analisis kebutuhan perangkat pikir digunakan untuk mengetahui spesifikasi pengguna untuk dapat memainkan *Game* si pitung, sebagai berikut.

Tabel 3.8 Spesifikasi Pengguna

Peran	**Pemain**
Usia	9 tahun ke atas
Tingkat Keterampilan	Dapat menggunakan komputer
Kondisi Fisik	Tidak cacat seperti buta atau cacat tubuh

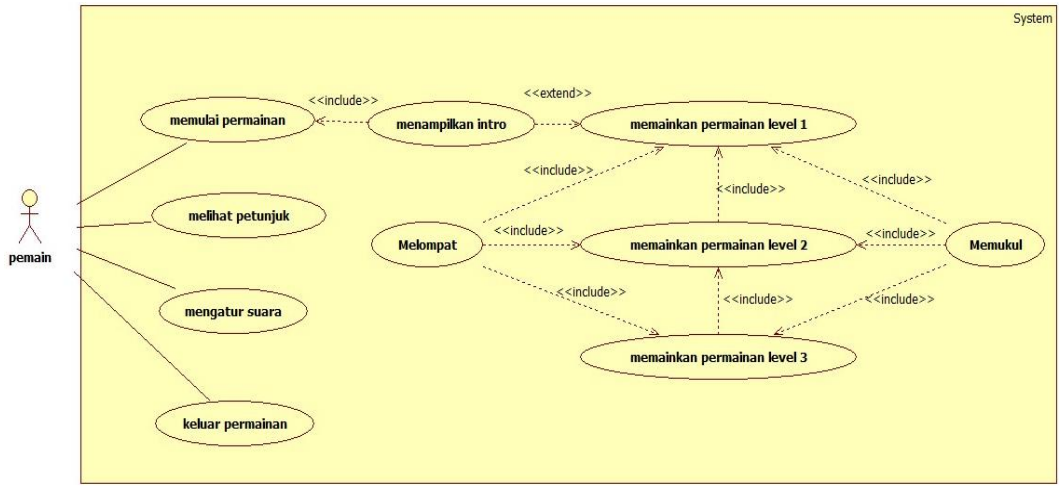
3.7 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional dilakukan untuk mengetahui gambaran sistem yang akan dibangun. Analisis kebutuhan fungsional terdiri dari pendefinisan fungsi *Game*, penggambaran alir aktivitas pada *Game* dan penggambaran interaksi antar objek pada *Game*.

Pendefinisian fungsi *Game* digunakan untuk menggambarkan fungsi-fungsi pada *Game*. Pendefinisian fungsi pada *Game* ini digambarkan dengan menggunakan *use case diagram*, definisi aktor,definisi*use case* dan skenario *use case*.

3.7.1 *Use Case* Diagram

Penggambaran fungsi *Game* berdasarkan interaksi antar aktor dan objek pada sistem digambarkan dengan menggunakan diagram use case seperti pada gambar 3.13.



Gambar 3.13 *Use Case* Diagram Petualangan Si Pitung

1.1. Definisi Aktor

Definisi aktor berfungsi menjelaskan peran setiap aktor yang terdapat pada use case. Berikut ini adalah definisi aktor.

Tabel 3.9 Definisi Aktor

No.	Aktor	Deskripsi
1.	Pemain	Pengguna/orang yang memainkan *Game* ini.

1.2. Definisi *Use Case*

Definisi *use case* berfungsi menjelaskan tiap-tiap proses dan objek yang terdapat pada *use case*. Berikut ini adalah definisi *use* case.

Tabel 3.10 Definisi *Use Case*

No.	Use Case	Deskripsi
1.	Memulai permainan	Proses memulai permainan
2.	Melihat petunjuk	Proses menampilkan petunjuk permainan
3.	Mengatur suara	Proses pengaturan suara
4.	Menampilkan intro	Proses cerita permainan
5.	Melompat	Proses aksi pemain dalam permainan
6.	Memukul	Proses aksi pemain dalam permainan
7.	Menampilkan permainan level 1	Proses memulai permainan level 1
8.	Menampilkan permainan level 2	Proses memulai permainan level 2
9.	Menampilkan permainan level 3	Proses memulai permainan level 3
10.	Keluar permainan	Proses keluar permainan

3.7.2 Skenario Use Case

Skenario pada diagram *Use Case* berisi tentang alur kejadian dari *Use Case* utama. Di dalamnya menjelaskan urutan interaksi antara aktor dengan *Use*

Case tersebut dari awal sampai akhir. Adapun skenario *Use Case* yang terdapat pada aplikasi permainan si pitung ini adalah sebagai berikut:

1. Skenario Use Case Memulai Permainan

Table 3.11 Skenario *Use Case* Mulai Permainan

Identifikasi	
Nomor	1
Nama	Memulai Permainan
Tujuan	Mulai permainan
Deskripsi	Proses untuk memulai permainan
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pemain memilih menu mulai saat berada di menu utama *Game*
Aksi Aktor	Reaksi Sistem
1. Memilih menu mulai	
	2. Menampilkan intro permainan
3. Tekan tombol lanjut	
	4. Tampil menu utama

2. Skenario Use Case Melihat Petunjuk

Table 3.12 Skenario *Use Case* Melihat Petunjuk

Identifikasi	
Nomor	2
Nama	Melihat Petunjuk
Tujuan	Menampilkan informasi permainan
Deskripsi	Proses untuk menampilkan informasi permainan
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pemain memilih menu petunjuk saat berada di menu utama *Game*

Aksi Aktor	Reaksi Sistem
1. Memilih menu petunjuk	
	2. Menampilkan halaman petunjuk
3. Menekan tombol kembali	
	4. Menampilkan halaman utama

3. Skenario *Use Case* Mengatur Suara

Table 3.13 Skenario Use Case Mengatur Suara

Identifikasi	
Nomor	3
Nama	Mengatur Suara
Tujuan	Mengatur suara permainan
Deskripsi	Proses untuk mengatur suara permainan
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pemain memilih menu suara saat berada di menu utama

Aksi Aktor	Reaksi Sistem
1. Memilih menu sound	
	2. Menampilkan pengaturan suara
3. Menekan tombol on	
	4. Menampilkan suara
5. Menekan tombol off	
	6. Tidak menampilkan suara

4. Skenario *Use Case* Menampilkan Intro

Table 3.14 Skenario Use Case Menampilkan Intro

Identifikasi	
Nomor	4
Nama	Menampilkan Intro

Tujuan	Menampilkan intro permainan
Deskripsi	Proses untuk menampilkan intro permainan
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pemain melihat intro permainan saat berada dalam halaman intro
Aksi Aktor	Reaksi Sistem
1. Menekan tombol lanjut	
	2. Menampilkan intro
3. Menekan tombol lanjut	
	4. Menampilkan permainan

5. Skenario *Use Case* Melompat

Table 3.15 Skenario Use Case Melompat

Identifikasi	
Nomor	5
Nama	Melompat
Tujuan	Menampilkan aksi pemain dalam permainan
Deskripsi	Proses untuk menampilkan aksi pemain dalam permainan
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pemain melompat saat berada dalam permainan.
Aksi Aktor	Reaksi Sistem
1. Menekan tombol w	
	2. Menampilkan aksi melompat

6. Skenario *Use Case* Memukul

Table 3.16 Skenario Use Case Memukul

Identifikasi

Nomor	6
Nama	Memukul
Tujuan	Menampilkan aksi pemain dalam permainan
Deskripsi	Proses untuk menampilkan aksi pemain dalam permainan
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pemain menampilkan aksi memukul saat berada dalam permainan.
Aksi Aktor	Reaksi Sistem
1. Menekan tombol k	
	2. Menampilkan aksi memukul
	3. Jika aksi memukul bertabrakan dengan musuh maka musuh akan mati.

7. Skenario *Use Case* Menampilkan Permainan level 1

Table 3.17 Skenario *Use Case* Menampilkan Permainan Level 1

Identifikasi	
Nomor	7
Nama	Menampilkan level 1
Tujuan	Menampilkan permainan level 1
Deskripsi	Proses untuk mulai memainkan permainan level 1
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pemain mulai memainkan permainan saat berada di tampilan permainan level 1
Aksi Aktor	Reaksi Sistem

1. Menggerakan karakter	
	2. Tampil gerakan karakter
	3. Cek tabrakan
	4. Jika karakter menabrak musuh, life poin berkurang
	5. Cek life point
	6. Jika life poin = 0, maka permainan akan diulang kembali
	7. Jika life poin != 0, maka karakter masih bisa dimainkan
8. Menekan tombol K	
	8. Karakter memukul
	9. Cek tabrakan pukulan
	10. Jika pukulan menabrak musuh, maka life poin musuh berkurang
	11. Jika life poin musuh = 0, maka musuh akan mati
	12. Jika life poin > 0, maka musuh hidup
	13. Cek jumlah warga
	14. Jika warga diselamatkan maka pemain mendapatkan life poin = 10
	15. Akhiri permainan
Kondisi akhir	Menampilkan permainan level 2

16. Skenario Use Case Menampilkan Permainan level 2

Table 3.18 Skenario Use Case Menampilkan Permainan Level 2

Identifikasi	
Nomor	8

Nama	Menampilkan level 2
Tujuan	Menampilkan permainan level 2
Deskripsi	Proses untuk mulai memainkan permainan level 2
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pemain mulai memainkan permainan saat berada di tampilan permainan level 2
Aksi Aktor	Reaksi Sistem
1. Menggerakan karakter	
	2. Tampil gerakan karakter
	3. Cek tabrakan
	4. Jika karakter menabrak musuh, life poin berkurang
	5. Cek life point
	6. Jika life poin = 0, maka permainan akan diulang kembali
	7. Jika life poin != 0, maka karakter masih bisa dimainkan
8. Menekan tombol B	
	8. Karakter memukul
	9. Cek tabrakan pukulan
	10. Jika pukulan menabrak musuh, maka life poin musuh berkurang
	11. Jika life poin musuh = 0, maka musuh akan mati
	12. Jika life poin > 0, maka musuh hidup
	13. Cek jumlah warga

	14. Jika warga diselamatkan maka pemain mendapatkan life poin = 10
	15. Akhir permainan
Kondisi akhir	Menampilkan permainan level 3

17. Skenario Use Case Menampilkan Permainan level 3

Table 3.19 Skenario Use Case Menampilkan Permainan Level 3

Identifikasi	
Nomor	9
Nama	Menampilkan level 3
Tujuan	Menampilkan permainan level 3
Deskripsi	Proses untuk mulai memainkan permainan level 3
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pemain mulai memainkan permainan saat berada di tampilan permainan level 3
Aksi Aktor	Reaksi Sistem
1. Menggerakan karakter	
	2. Tampil gerakan karakter
	3. Cek tabrakan
	4. Jika karakter menabrak musuh, life poin berkurang
	5. Cek life point
	6. Jika life poin = 0, maka permainan akan diulang kembali
	7. Jika life poin != 0, maka karakter masih bisa dimainkan

8. Menekan tombol B	
	8. Karakter mengeluarkan golok
	9. Cek tabrakan golok
	10. Jika golok menabrak musuh, maka life poin musuh berkurang
	11. Jika life poin musuh = 0, maka musuh akan mati
	12. Jika life poin > 0, maka musuh hidup
	13. Cek jumlah warga
	14. Jika warga diselamatkan maka pemain mendapatkan life poin = 10
	15. Jika life point boss = 0, maka permainan akan berakhir
	16. Jika life point boss!= 0, maka permainan tidak akan berakhir
	17. Akhir permainan
Kondisi akhir	Menampilkan intro

18. Skenario Use Case Keluar Permainan

Table 3.20 Skenario Use Case Keluar Permainan

Identifikasi	
Nomor	10
Nama	keluar Permainan
Tujuan	keluar permainan
Deskripsi	Proses untuk keluar permainan
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pemain tekan tombol exit yang berada di ujung

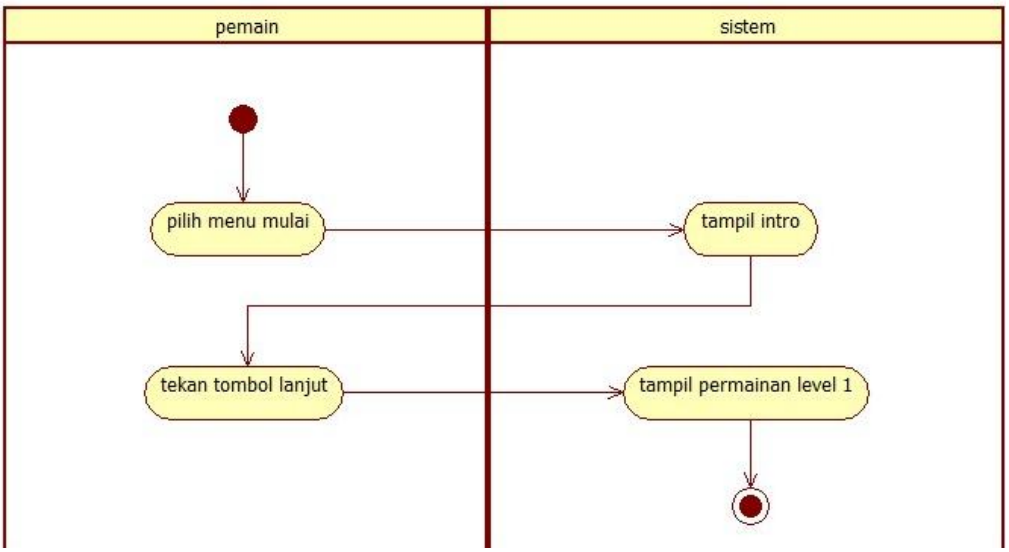
	kanan *Game*
Aksi Aktor	Reaksi Sistem
5. Memilih tombol exit	
	6. keluar permainan

3.7.2 *Activity* Diagram

Diagram ini berisi penjelasan secara lengkap mengenai urutan aktivitas yang terjadi antara pemain dengan *game* si Pitung.

1. ***Activity* Diagram Memulai Permainan**

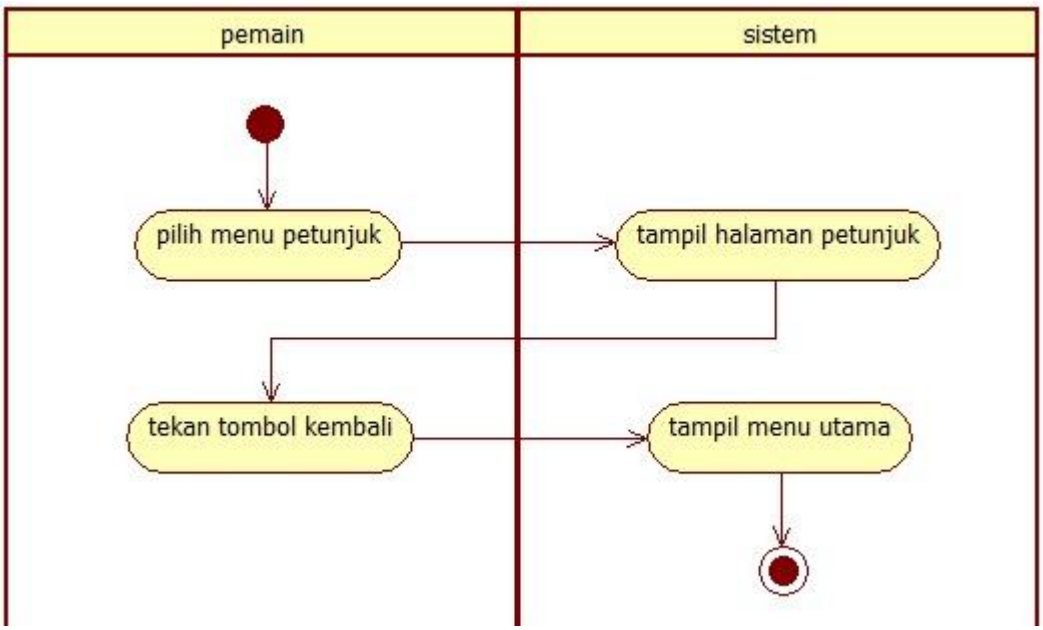
Proses ini terjadi saat pemain memulai permainan si Pitung. Sistem akan menampilkan halaman utama dari permainan si Pitung. *Activity* Diagram Mulai Permainan terdapat pada gambar 3.14 berikut:



Gambar 3.14 *Activity* Diagram Memulai Permainan

2. ***Activity* Diagram Melihat Petunjuk**

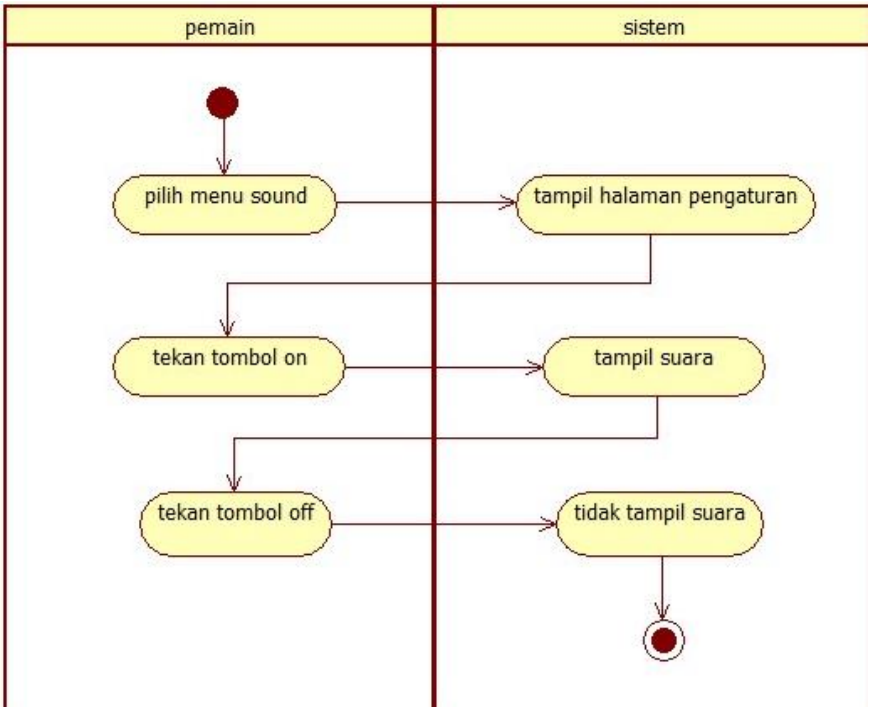
Proses ini terjadi saat pemain memulai memilih menu petunjuk pada *game* si Pitung. Sistem akan menampilkan petunjuk permainan dari permainan si Pitung. *Activity* Diagram petunjuk terdapat pada gambar 3.15 berikut:



Gambar 3.15 *Activity* Diagram Melihat Petunjuk

3. Diagram *activity* Mengatur Suara

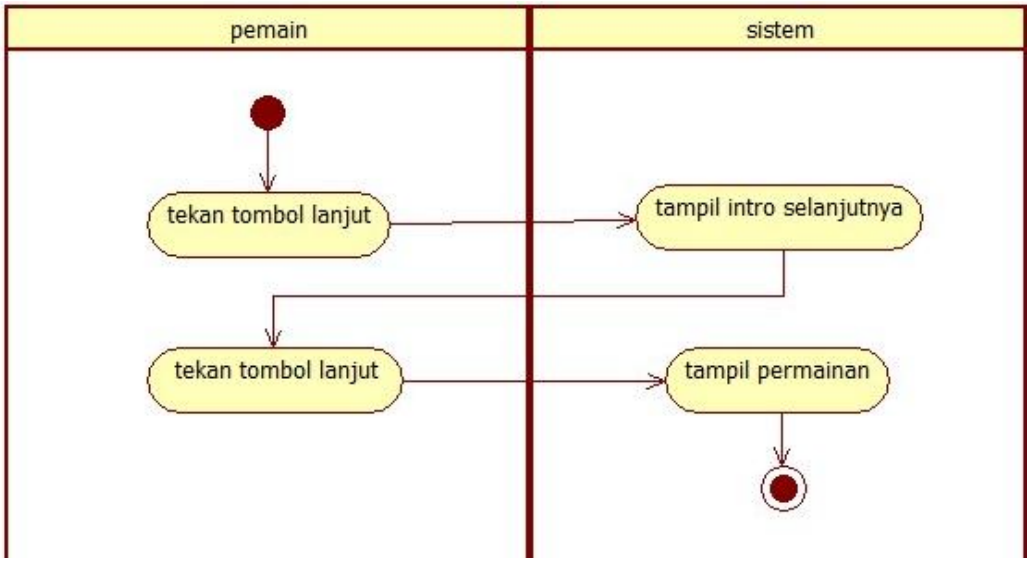
Proses ini terjadi saat pemain memulai memilih menu suara pada *game* si Pitung. Sistem akan menampilkan pengaturan suara dari permainan si Pitung. *Activity* Diagram petunjuk terdapat pada gambar 3.16 berikut:



Gambar 3.16 *Activity* Diagram Mengatur Suara

4. *Activity* Diagram Menampilkan *Intro*

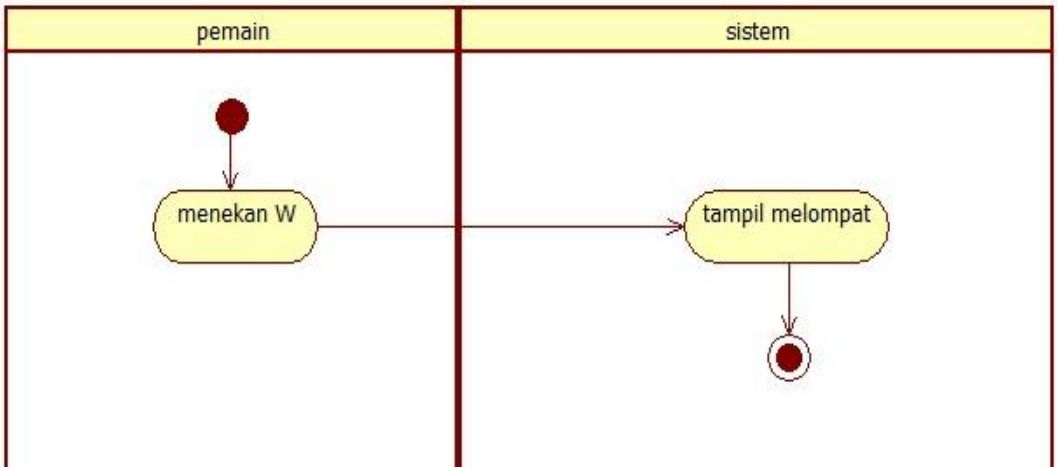
Proses ini terjadi saat pemain memulai masuk kedalam *game* si Pitung. Sistem akan menampilkan intro dari permainan si Pitung. *Activity* Diagram petunjuk terdapat pada gambar 3.17 berikut:



Gambar 3.17 Diagram *activity* Menampilkan *Intro*

5. *Activity* Diagram Melompat

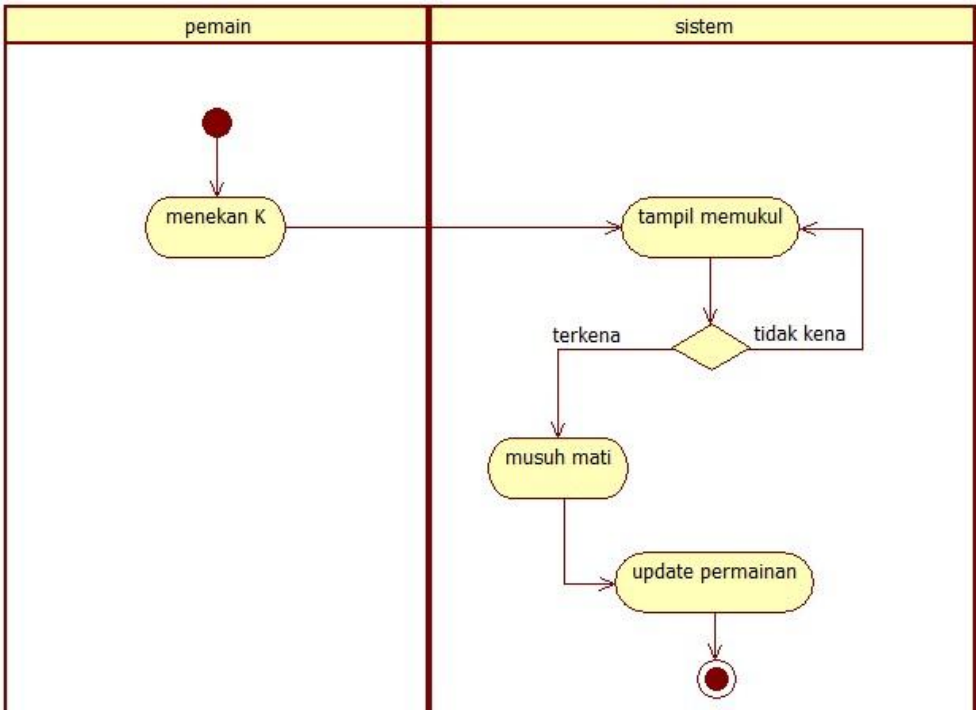
Proses ini terjadi saat pemain memulai menekan tombol w pada *game* si Pitung. Sistem akan menampilkan aksi melompat pada karakter si Pitung. *Activity* Diagram melompat terdapat pada gambar 3.18 berikut:



Gambar 3.18 *Activity* Diagram Melompat

6. Diagram *activity* Memukul

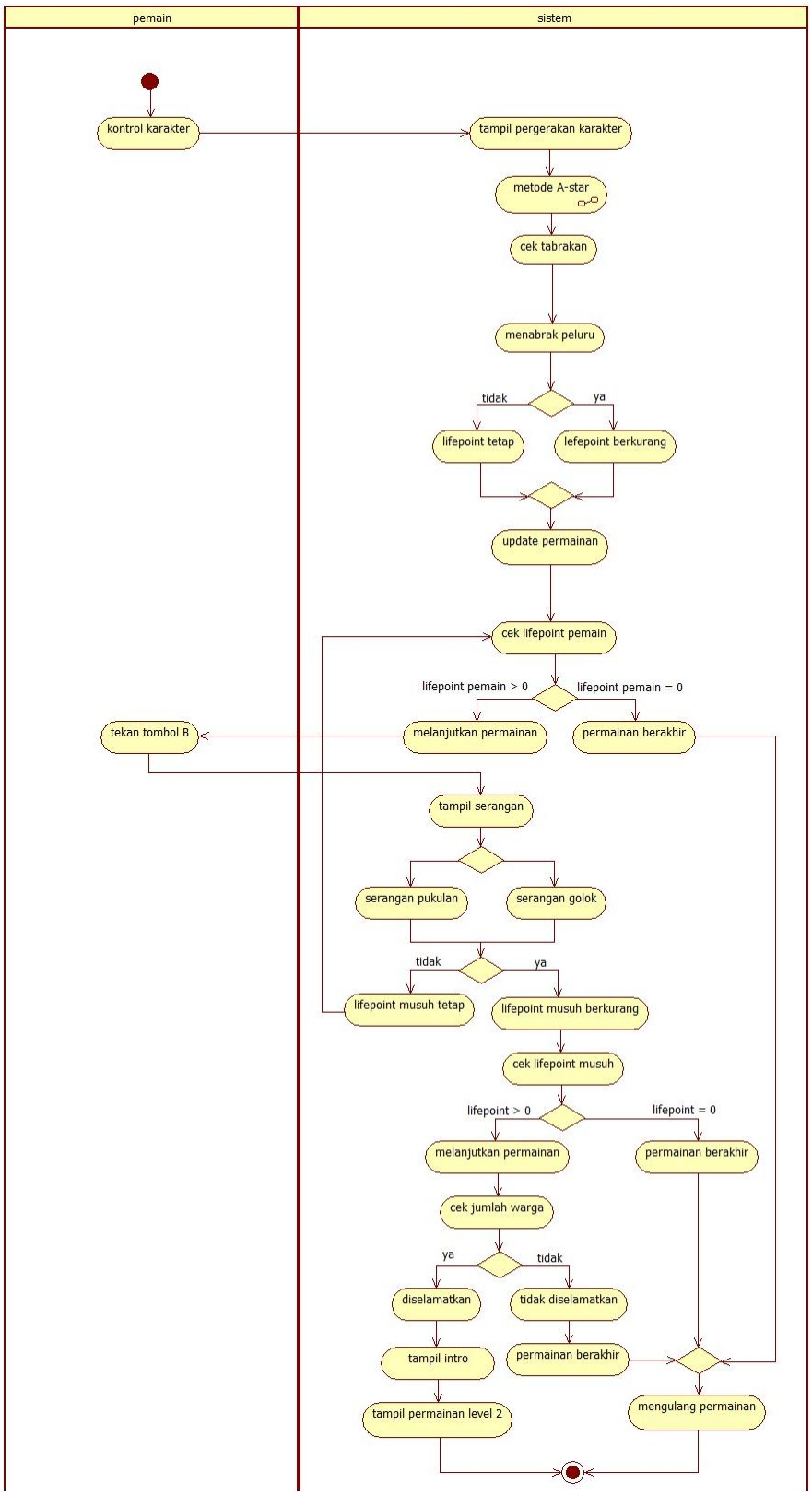
Proses ini terjadi saat pemain memulai menekan tombol k pada *game* si Pitung. Sistem akan menampilkan aksi memukul pada karakter si Pitung. *Activity* Diagram melompat terdapat pada gambar 3. 19 berikut:



Gambar 3.19 A*ctivity* **Diagram Memukul**

7. *Activity* Diagram Menampilkan Permainan Level 1

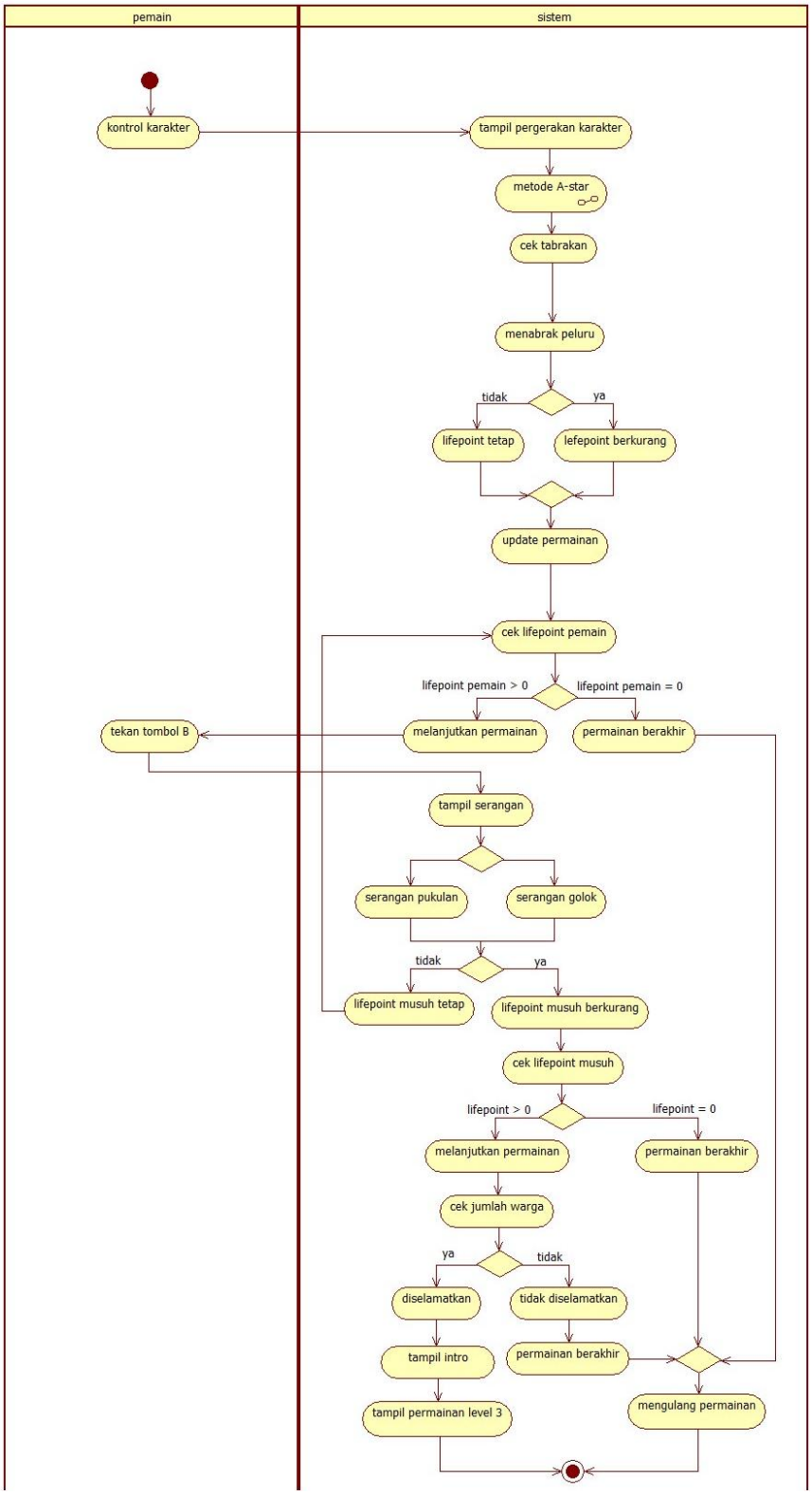
Proses ini terjadi saat pemain memulai masuk kedalam *game* si Pitung. Sistem akan menampilkan level 1 dari permainan si Pitung. *Activity* Diagram petunjuk terdapat pada gambar 3.20 berikut:



Gambar 3.20 *Activity* Diagram Menampilkan *Level 1*

8. *Activity* Diagram Menampilkan Permainan Level 2

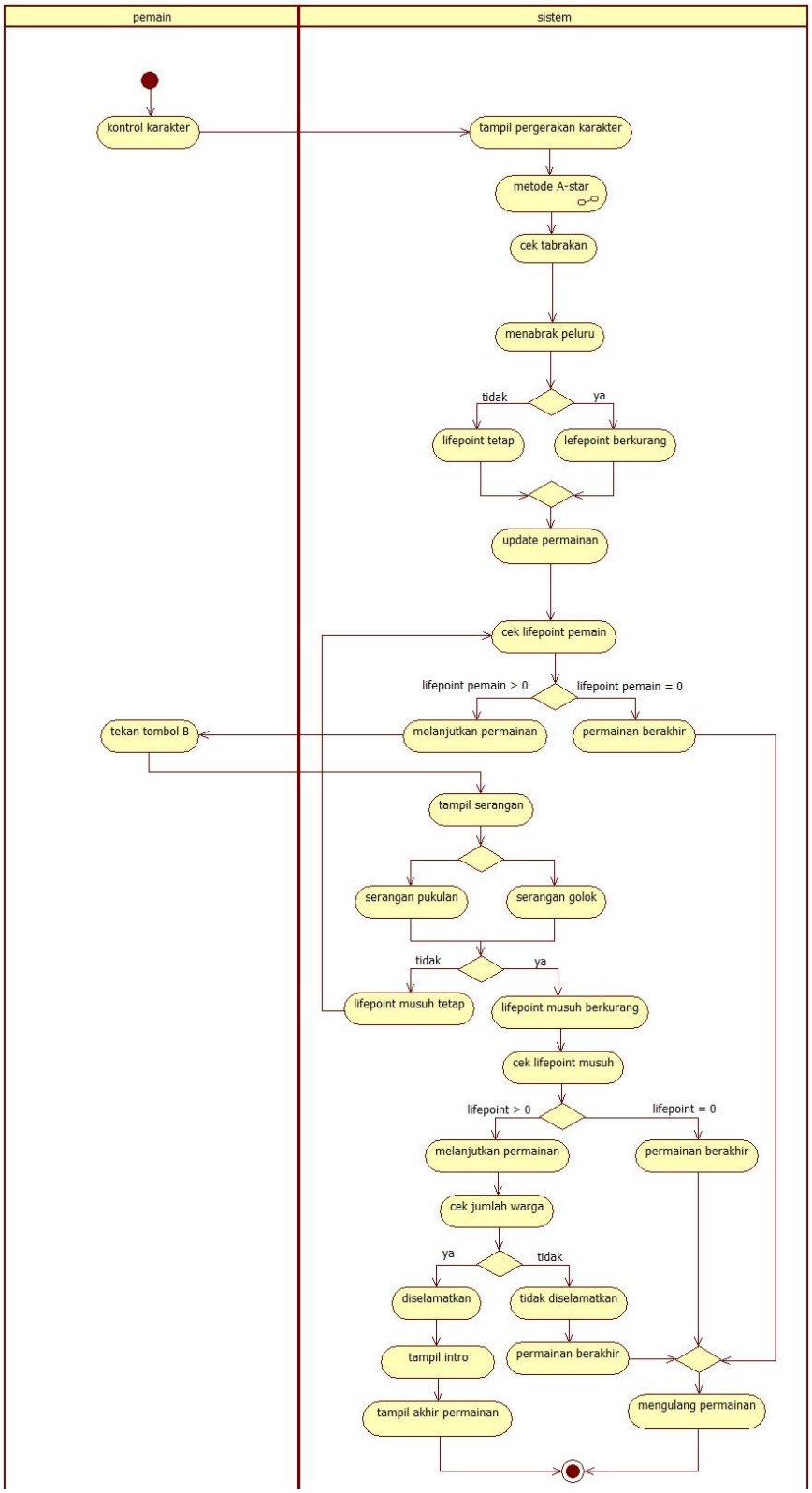
Proses ini terjadi saat pemain memulai masuk kedalam *game* si Pitung. Sistem akan menampilkan level 2 dari permainan si Pitung. *Activity* Diagram petunjuk terdapat pada gambar 3.21 berikut:



Gambar 3.21 *Activity* Diagram Menampilkan *Level 2*

9. *Activity* Diagram Menampilkan Permainan Level 3

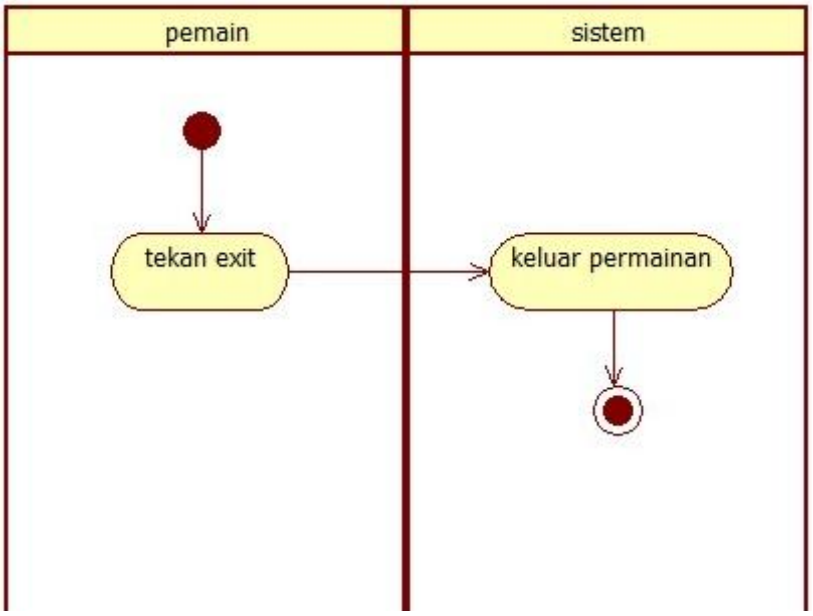
Proses ini terjadi saat pemain memulai masuk kedalam *game* si Pitung. Sistem akan menampilkan level 3 dari permainan si Pitung. *Activity* Diagram petunjuk terdapat pada gambar 3.22 berikut:



Gambar 3.22 *Activity* Diagram Menampilkan *Level 3*

10. *Activity* Diagram Keluar Permainan

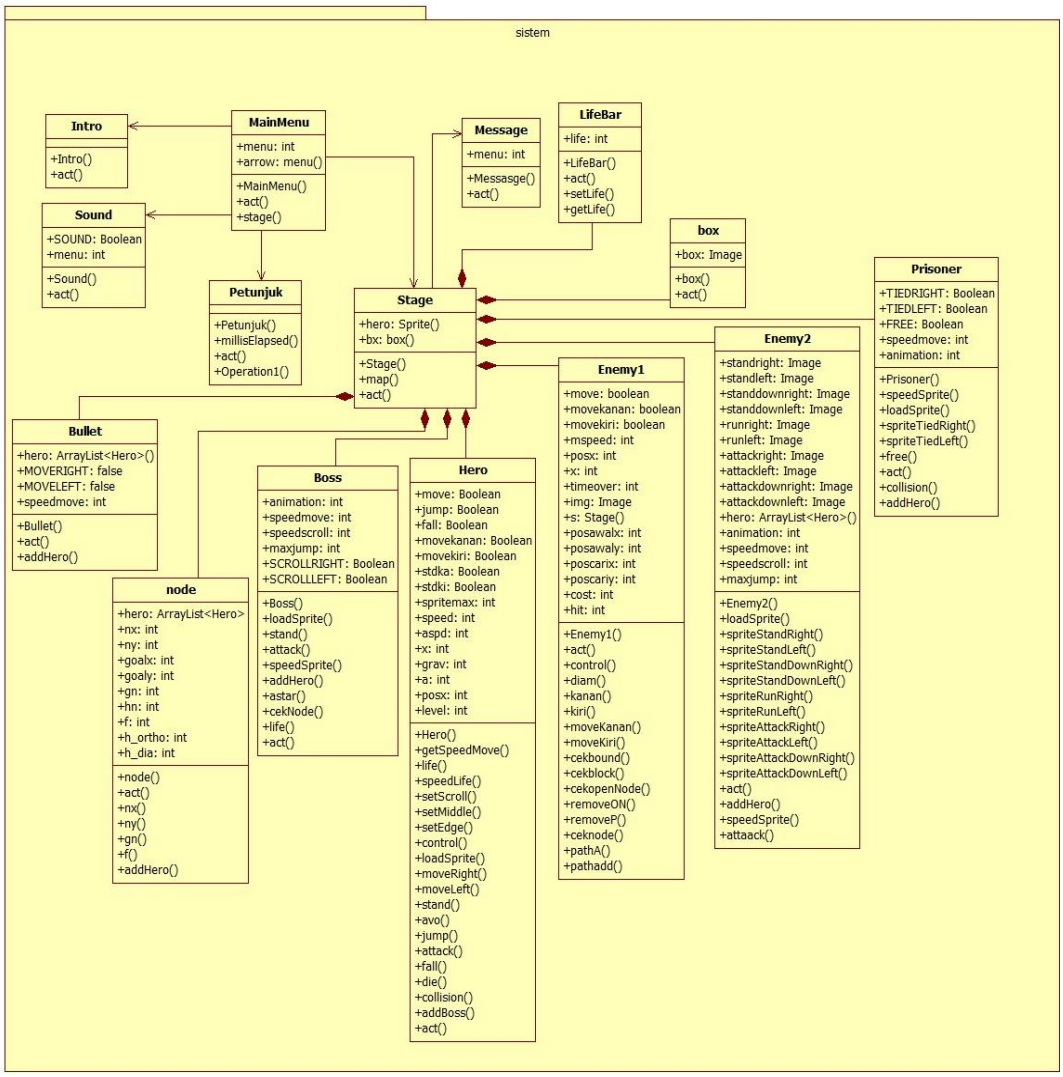
Proses ini terjadi saat pemain menekan tombol exit pada *game* si Pitung. Sistem akan keluar dari permainan. *Activity* Diagram petunjuk terdapat pada gambar 3.23 berikut:



Gambar 3.23 *Activity* Diagram Keluar Permainan

3.7.3 *Class* Diagram

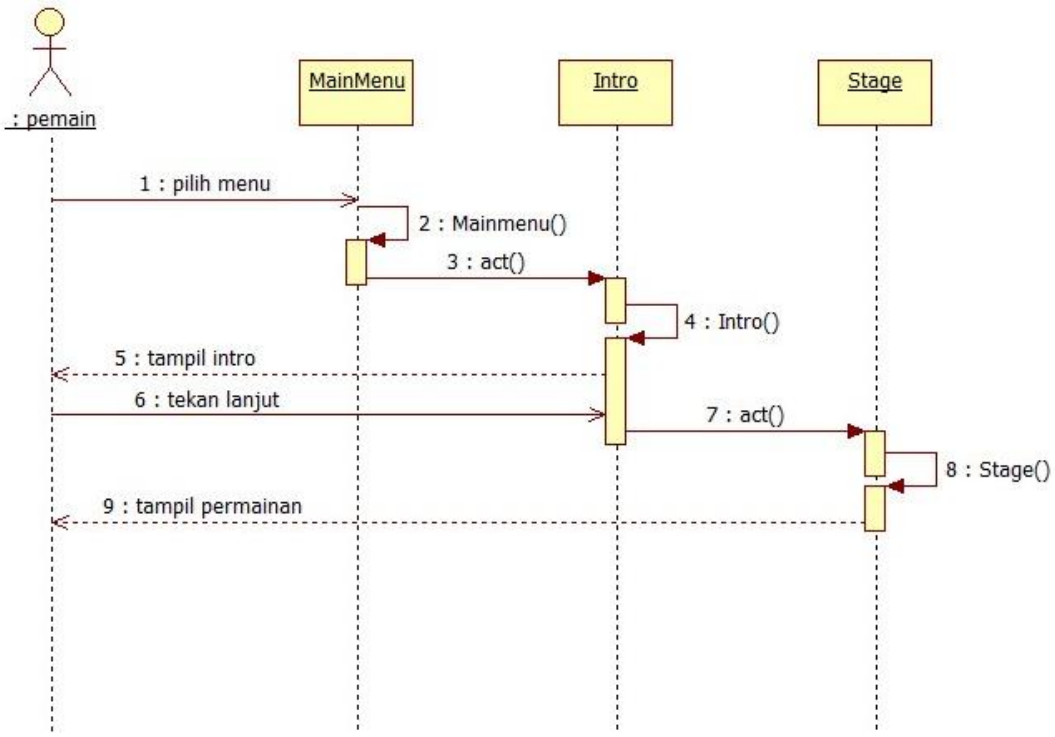
Class Diagram merupakan penggambaran struktur sistem dengan mendefinisikan kelas-kelas dari objek yang akan dibuat untuk membangun sistem. Berikut ini adalah gambar 3.24 *Class* Diagram pada *game* ini.



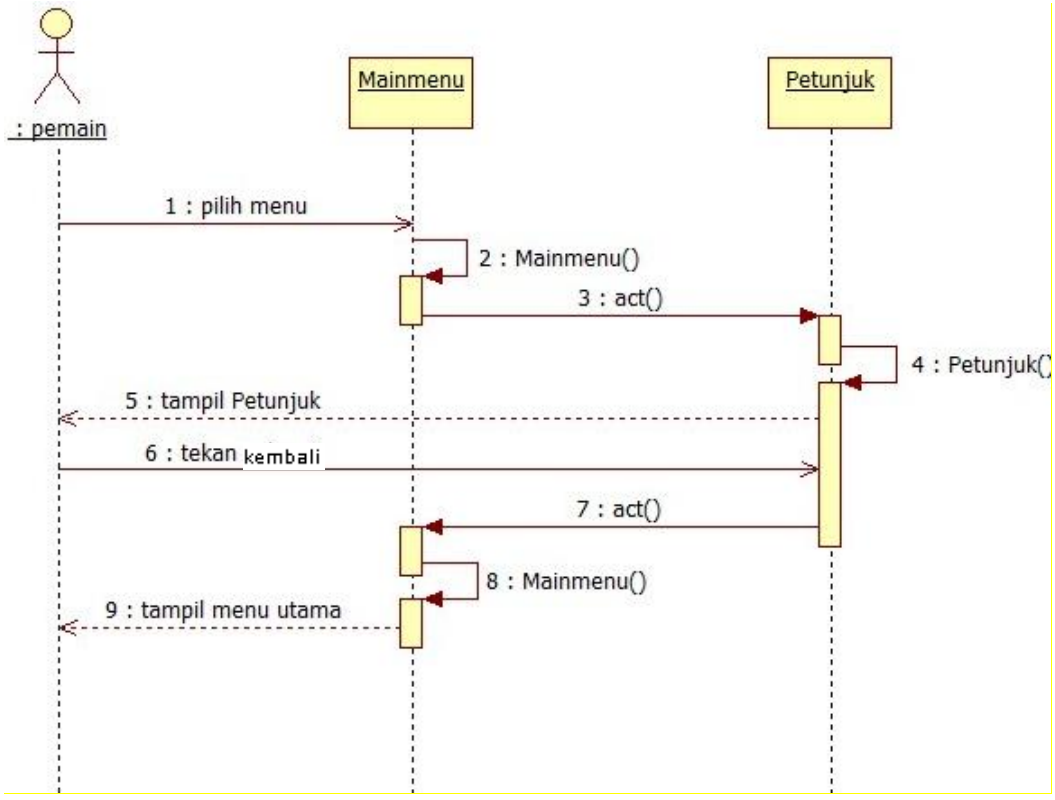
Gambar 3.24 *Class* Diagram Petualangan Si Pitung

3.7.4 *Sequence* Diagram

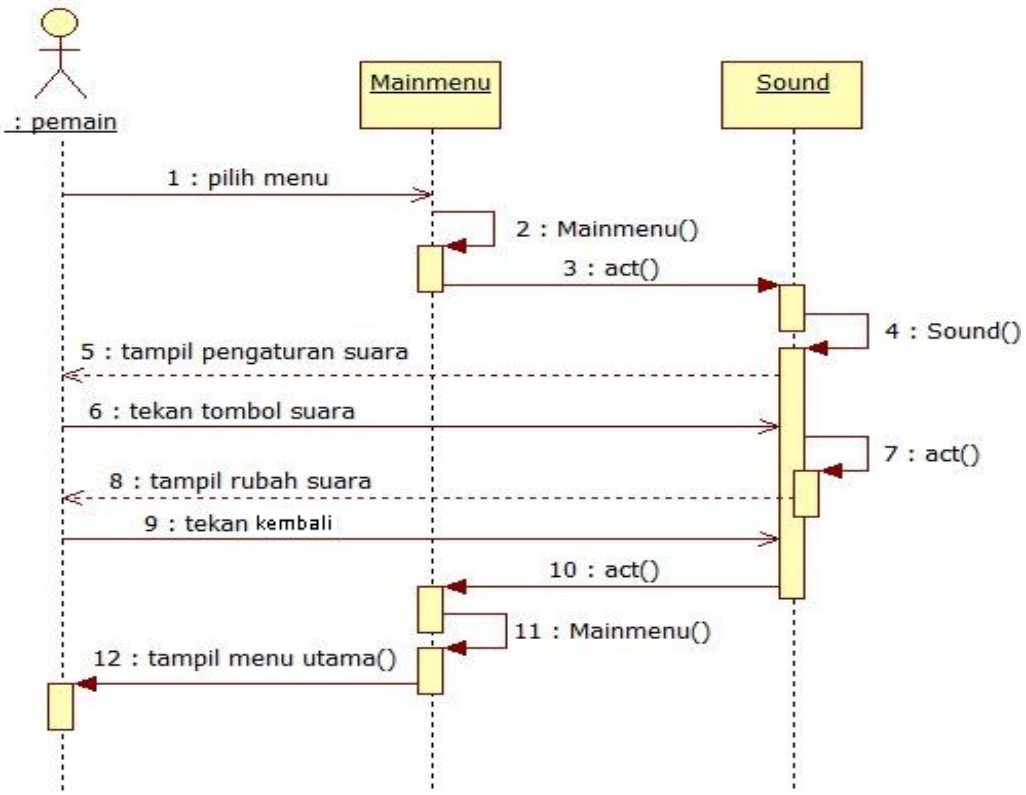
Sequence Diagram merupakan penggambaran interaksi antar objek yang terjadi di dalam dan di luar sistem pada permainan. *Sequence* Diagram terdiri atas dimensi *vertical* (waktu) dan dimensi *horizontal* (objek yang terkait). Pada *Sequence* Diagram ini, digambarkan skenario atau rangkaian langkah-langkah yang dilakukan oleh sitem sebagai respon dari sebuah *event* (kejadian) untuk menghasilkan output tertentu. Berikut ini adalah gambaran *Sequence* Diagram pada *game* ini.



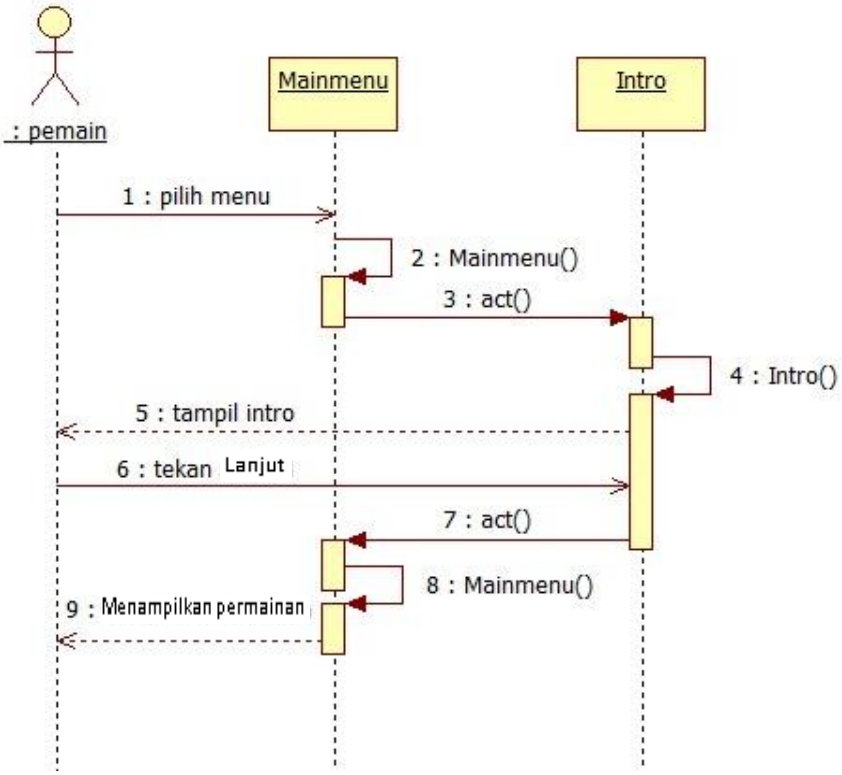
Gambar 3.25 *Sequence* Diagram Memulai Permainan



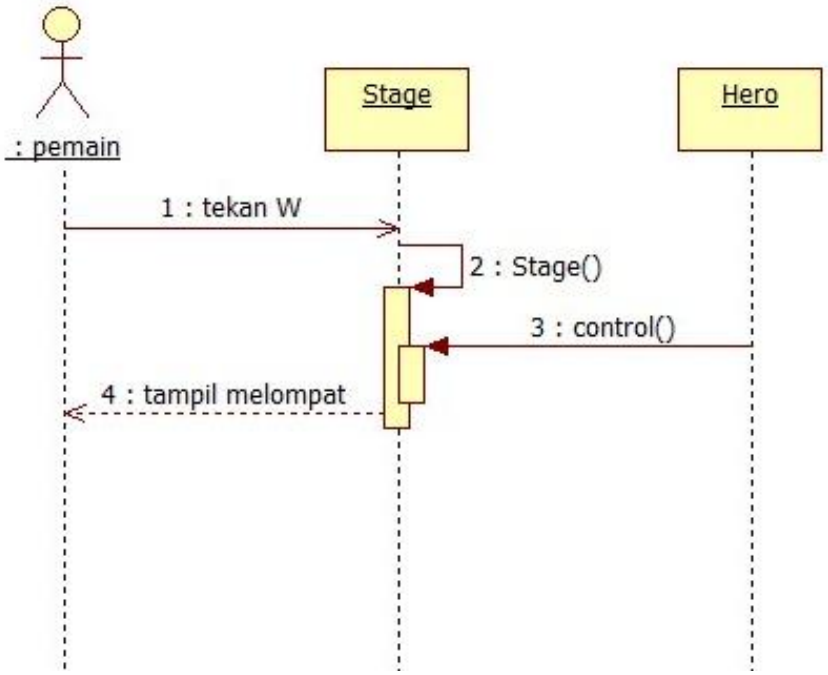
Gambar 3.26 *Sequence* Diagram Melihat Petunjuk



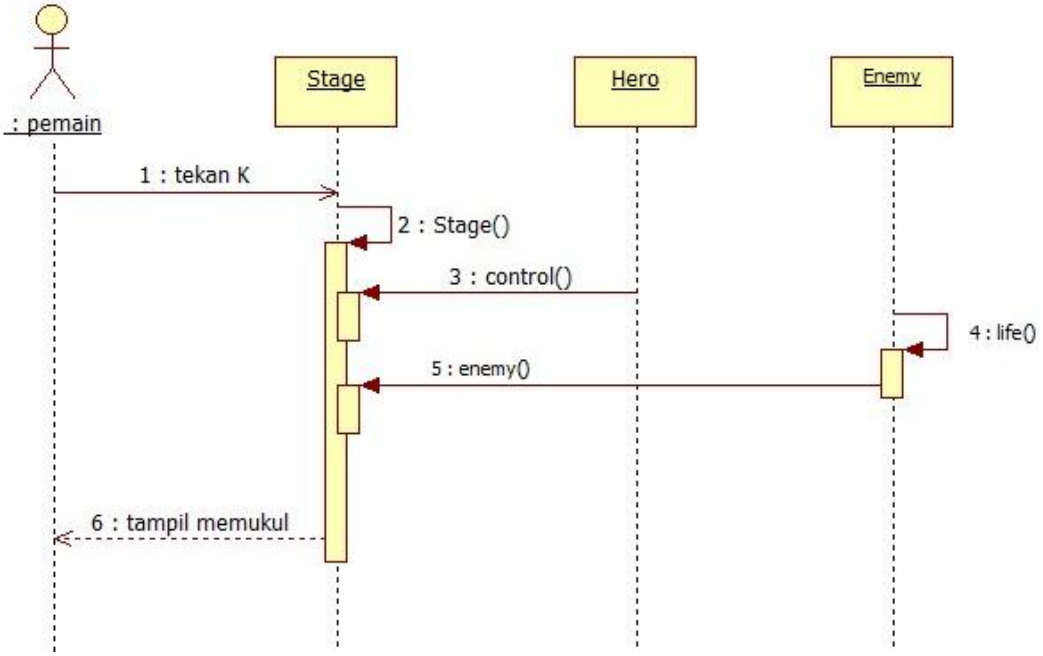
Gambar 3.27 *Sequence* Diagram Mengatur Suara



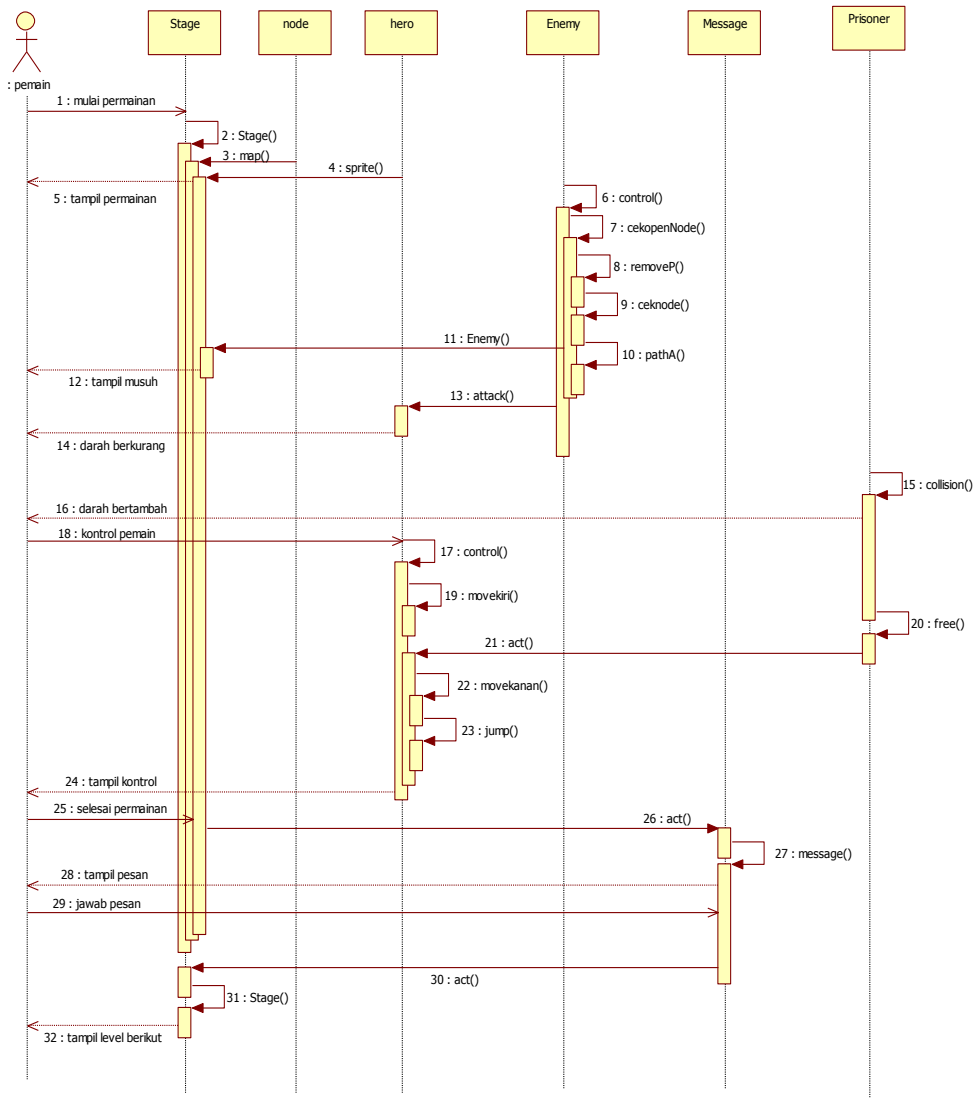
Gambar 3.28 *Sequence* Diagram Menampilkan Intro



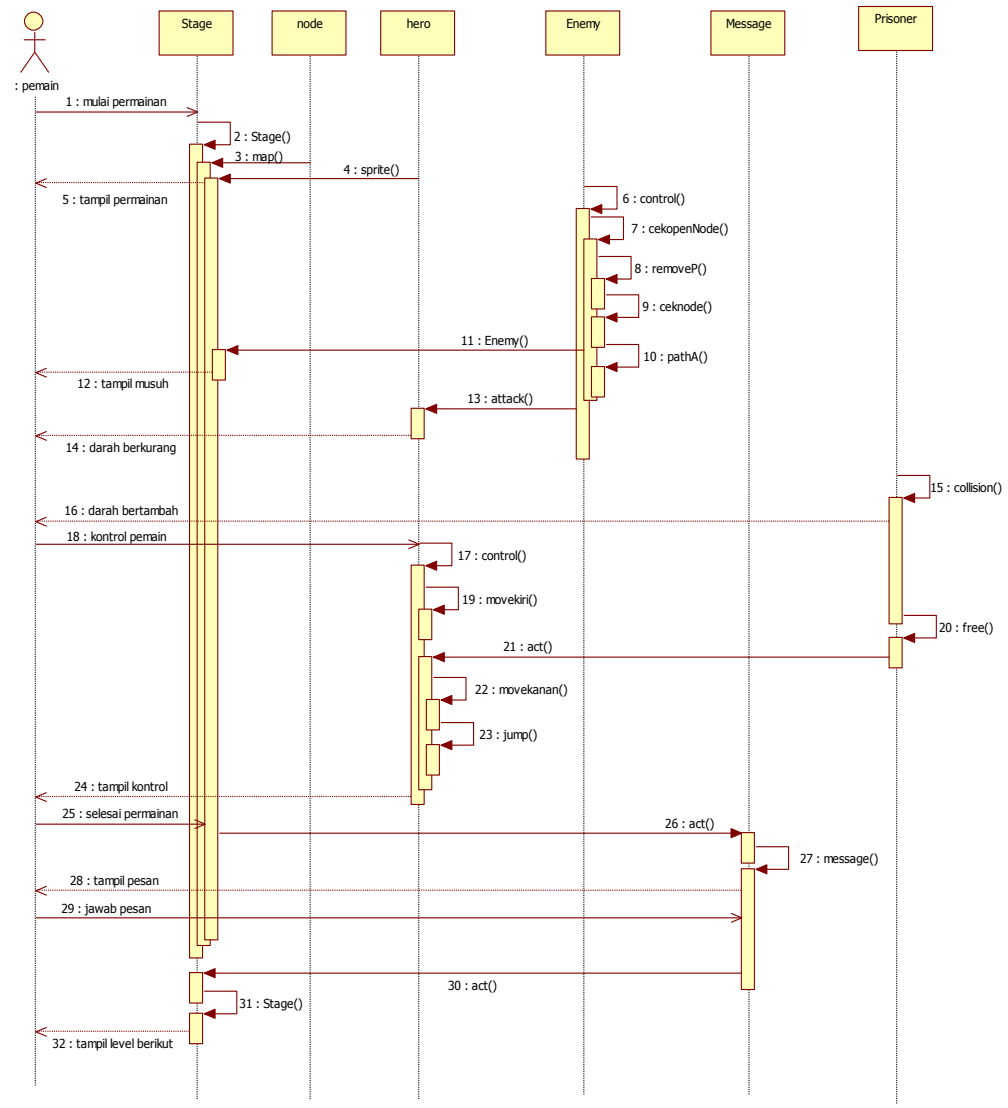
Gambar 3.29 *Squence* Diagram Melompat



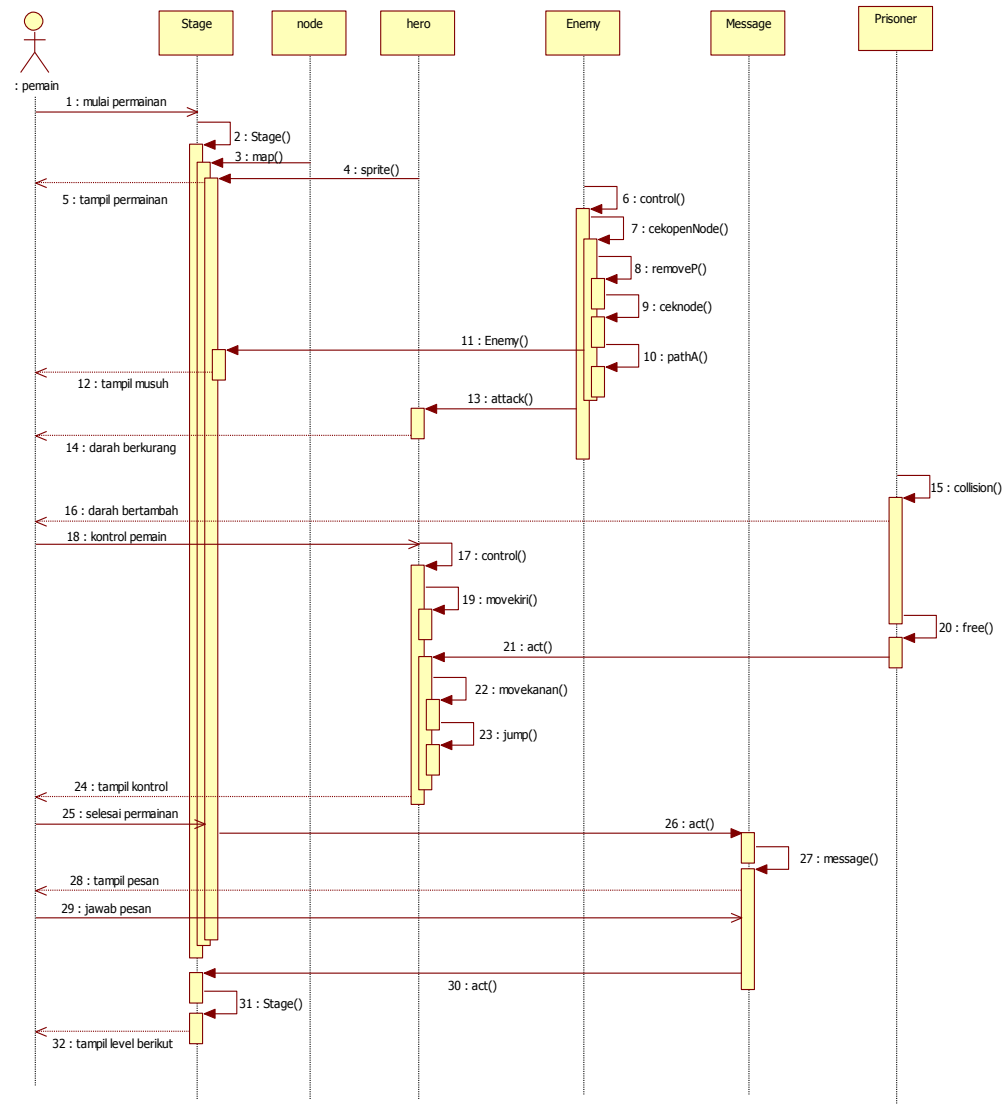
Gambar 3.30 *Sequence* Diagram Memukul



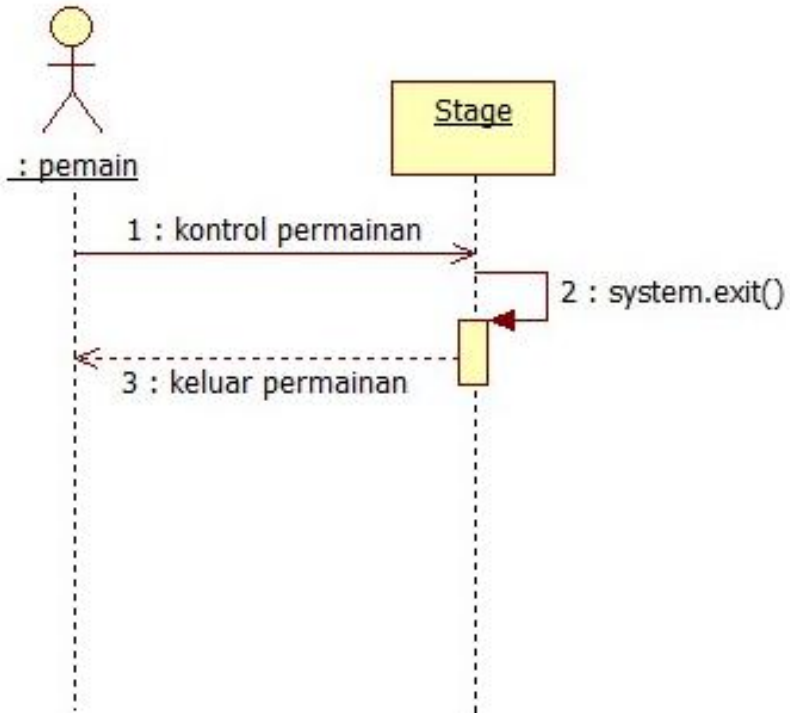
Gambar 3.31 *Sequence* Diagram Menampilkan Permainan *level* 1



Gambar 3.32 *Sequence* Diagram Menampilkan Permainan *level* 2



Gambar 3.33 *Sequence* Diagram Menampilkan Permainan *level* 3



Gambar 3.34 *Sequence* Diagram Keluar Permainan

3.8 Perancangan Sistem

Tahap perancangan merupakan kelanjutan dari proses analisis dimana dilakukan perubahan-perubahan terhadap sistem yang sedang berjalan. Hal ini dilakukan untuk mengatasi kekurangan yang ada, memudahkan pekerjaan yang dilakukan oleh orang yang terlibat dan menghemat waktu pekerjaan.

Perancangan *game* ini meliputi perancangan karakter, perancangan *storyboard*, perancangan struktur menu, perancangan antarmuka, perancangan pesan, jaringan semantik, perancangan prosedural. Semua perancangan yang telah disebutkan diatas harus saling berhubungan agar menciptakan suatu sistem yang baik.

3.8.1 Perancangan Karakter

Perancangan karakter adalah bagian dimana karakter pada *game* dijelaskan cirri-cirinya baik itu secara fungsionalitas peran maupun secara desain. Pada *game* ini karakter merupakan setiap unit pada permainan, sehingga pada perancangan

karakter akan mendeskripsikan setiap kelas unit pada setiap fraksi. Perancangan karakter pada *game* ini dijabarkan seperti berikut.

Tabel 3.21 Perancangan Karakter

No	Nama	Gambar	Keterangan
1	pitung		Ini adalah tokoh pahlawan yang karakternya dimainkan oleh pemain. Karakter ini dapat memukul dan melempar golok.
2	Prajurit 1		Ini adalah musuh yang berada dalam permainan. Karakter ini dapat menembak kepada pemain
3	box		Ini adalah kotak penghalang dalam permainan.
4	warga		Ini adalah warga yang ditawan oleh kompeni belanda
5	Prajurit 2		Ini adalah musuh yang berada dalam permainan. Karakter ini dapat memukul kepada pemain.
6	Menir		Ini adalah karakter boss yang berada pada akhir permainan. Karakter boss ini dapat menembak pemain dengan demage yang berbeda.
7	Life Point		Ini adalah bar life point untuk si pitung. Bar ini menampilkan informasi si pitung.

8	golok		Ini adalah senjata yang dipakai oleh pemain dalam permainan.

3.8.2 Perancangan Level

Pada setiap level, akan ada nusuh yang diterapkan algoritma A* untuk tingkat kesulitan pemain. Pemain dapat bergerak dan menyerang dengan cara memukul atau memakai senjata. Kemudian selain mengalahkan musuh, pemain harus menyelamatkan warga dari kompeni. Dalam permainan ini dibagi menjadi 3 tingkat kesulitan yaitu, jumlah musuh dalam setiap level berbeda-beda, memilki 3 musuh yang berbeda-beda dalam menyerang pemain dalam setiap level, dan memliki boss akhir yang kuat di level terakhir.

3.8.3 Perancangan *Storyboard*

Storyboard merupakan sketsa gambar yang disusun secara berurutan berdasarkan naskah untuk menggambarkan lingkungan pada permainan. Pada *game* ini, *storyboard* merupakan sebuah peta atau map yang akan digunakan untuk bermain dan diterapkan pada sebuah *stage*. Map pada *game* ini digambarkan seperti berikut.



Gambar 3.35 Permainan *Level* 1



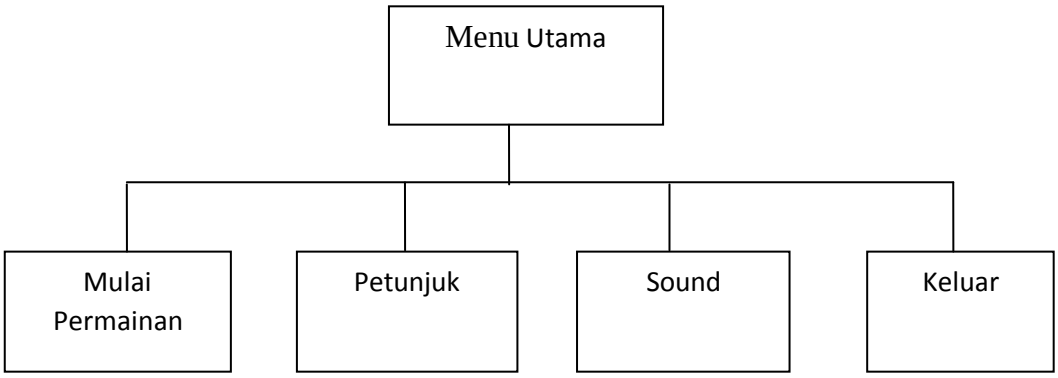
Gambar 3.36 Permainan *Level* 2



Gambar 3.37 Permainan *Level* 3

3.8.4 Perancangan Struktur Menu

Dalam pembangunan permainan diperlukan sebuah struktur menu untuk menunjukan keterkaitan dan tingkatan antar menu utama dan sub menu, sehingga dapat lebih mudah di mengerti. Perancangan struktur menu dapat dilihat dalam gambar berikut.

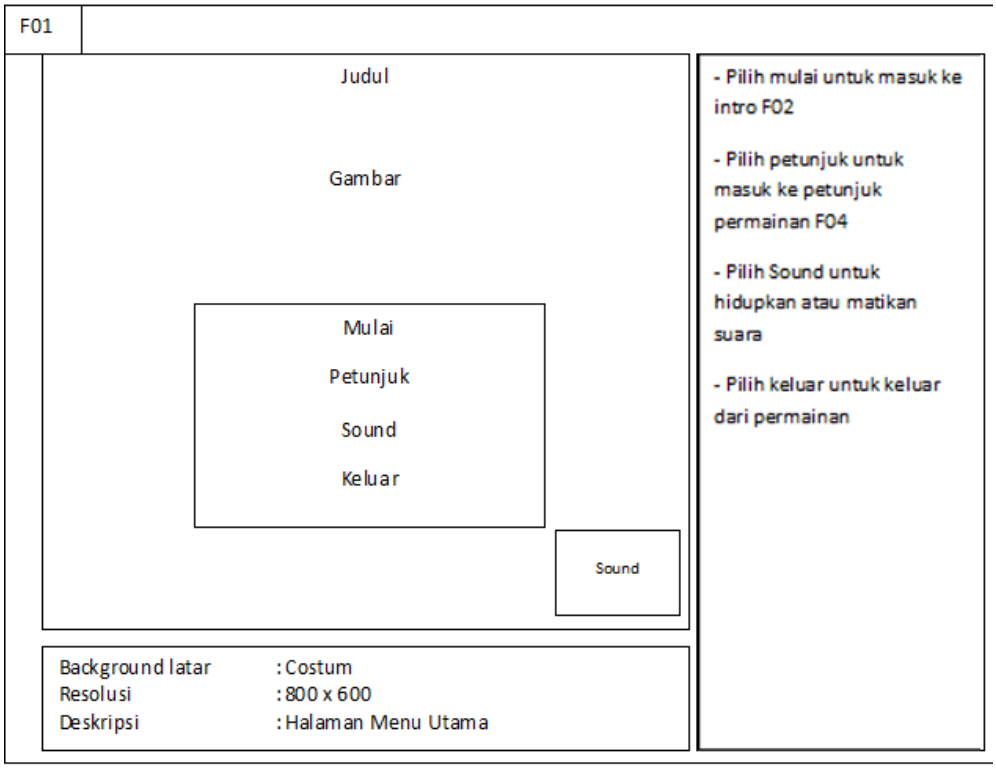


Gambar 3.38 Struktur Menu *Game* Petualangan Si Pitung

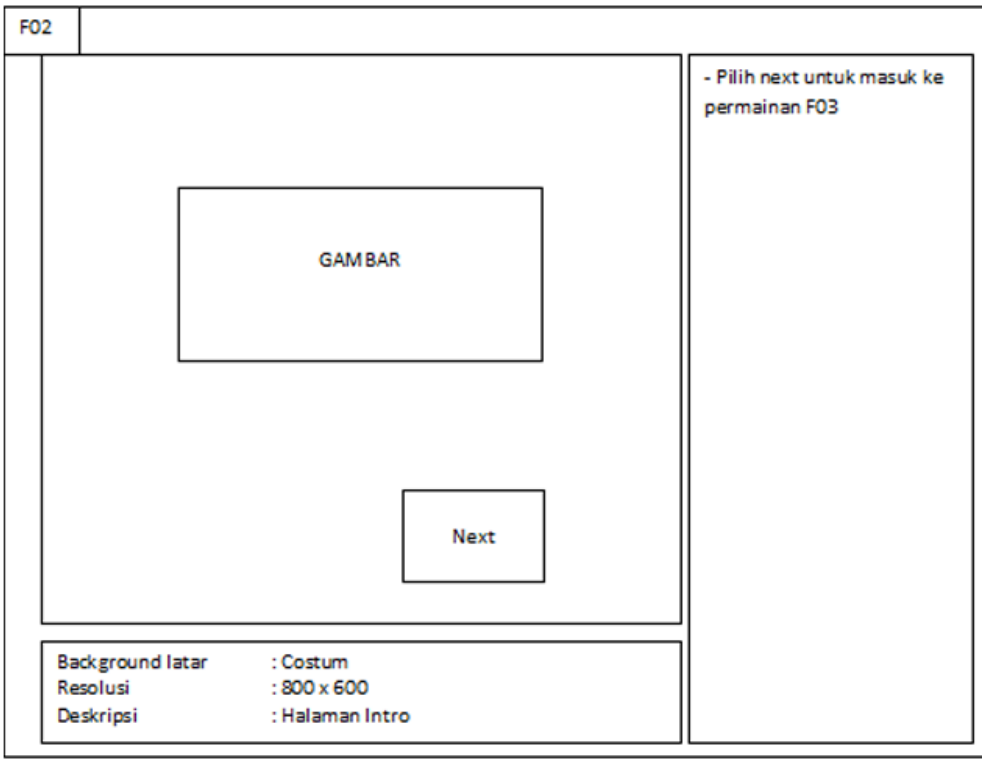
Mulai Permainan	:	merupakan menu untuk memulai permainan
Petunjuk	:	merupakan menu untuk menampilkan informasi tentang cara memainkan.
Sound	:	merupakan menu untuk mengatur suara dari permainan.
Keluar	:	merupakan menu untuk keluar dari permainan

3.8.5 Perancangan Antar Muka

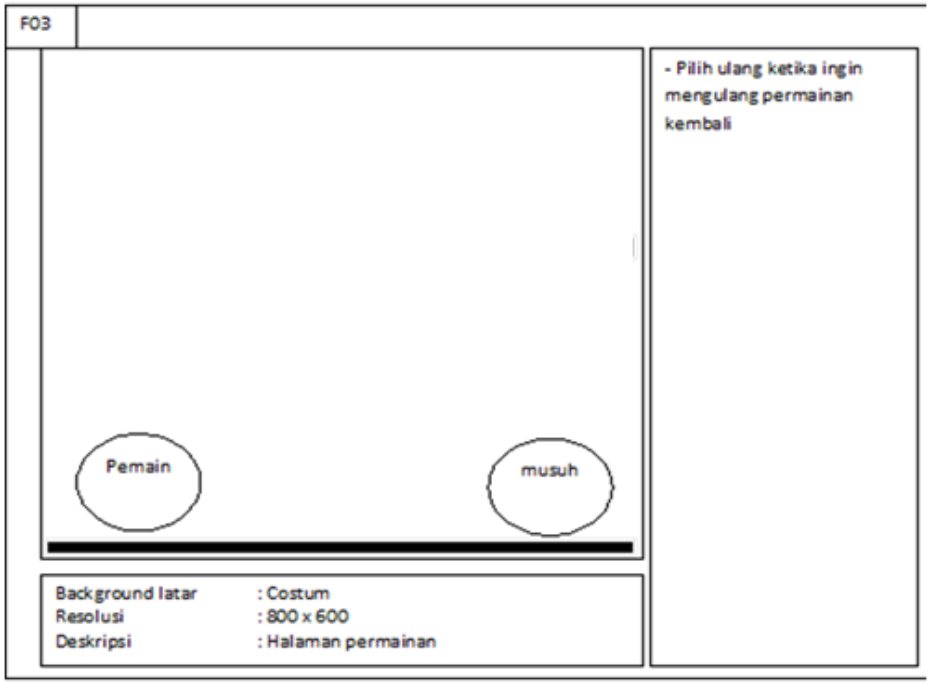
Perancangan antarmuka menu main pada permainan si pitung dapat dilihat pada gambar berikut.



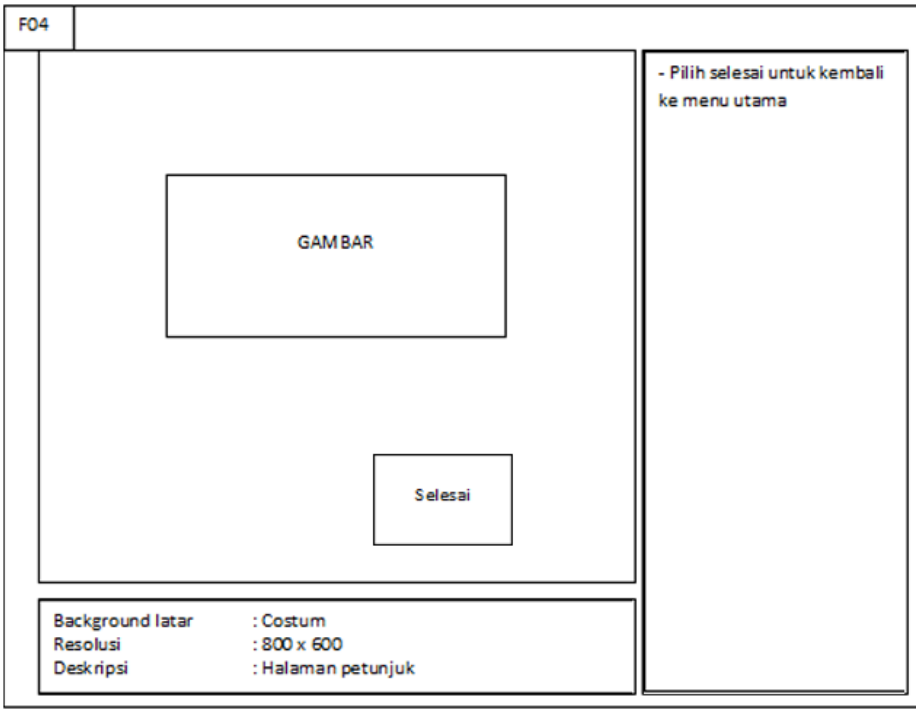
Gambar 3.39 Perancangan Antarmuka Menu Utama



Gambar 3.40 Perancangan Antarmuka Intro Permainan



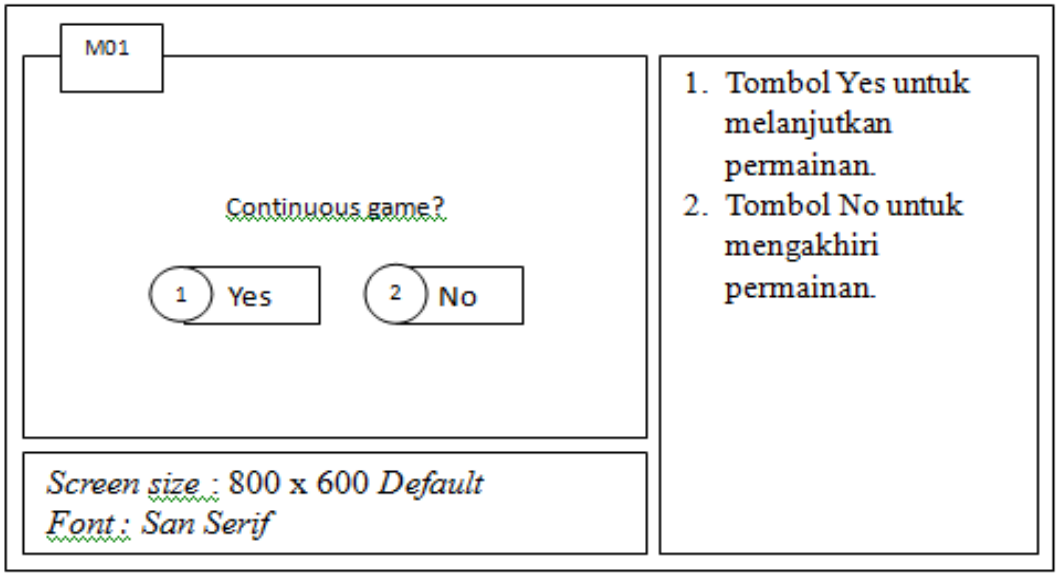
Gambar 3.41 Perancangan Antarmuka Halaman Permainan



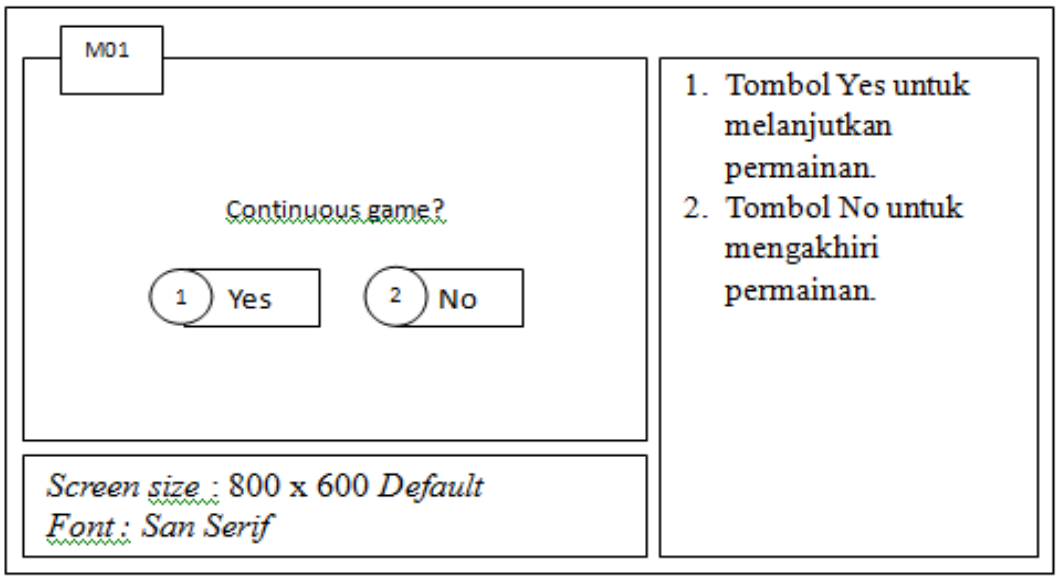
Gambar 3.42 Perancangan Antarmuka Petunjuk Permainan

3.8.6 Perancangan Antar Muka Pesan

Perancangan antar muka pesan pada aplikasi ini dapat dilihat sebagai berikut :



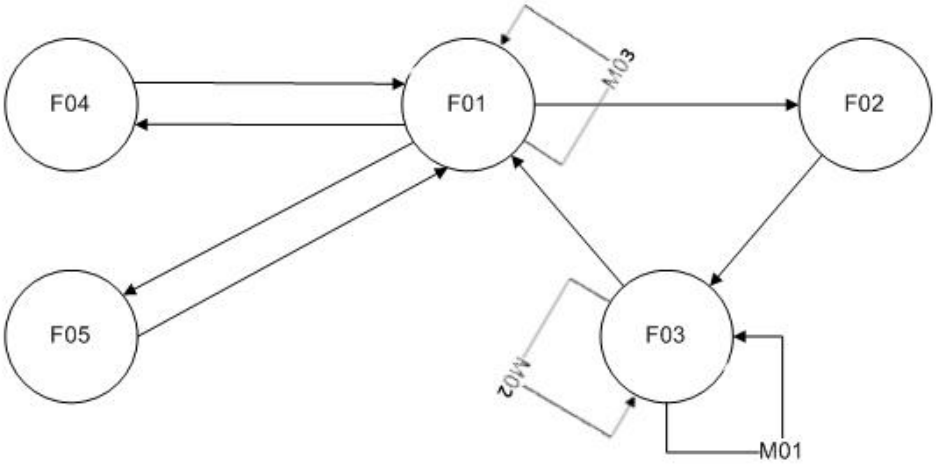
Gambar 3.43 Perancangan Antarmuka Peringatan Lanjut Permainan



Gambar 3.44 Perancangan Antarmuka Exit Permainan

3.8.7 Jaringan Semantik

Jaringan semantik pada permainan si pitung dapat dilihat pada gambar berikut.



Gambar 3.44 Jaringan Semantik