BAB III
ANALISIS DAN KEBUTUHAN ALGORITMA

3.1 Analisis Masalah

Pada analisis masalah *game* yang umumnya tingkat kesulitan berupa tahapan yang dimulai dari tingkatan termudah hingga tahapan tersulit. Namun ada pula *game* yang berada pada satu *level* dengan tingkat kesulitan yang berawal dari tingkat termudah hingga tingkat tersulit, hanya saja pada *game* dengan satu *level* ini, jika tingkat kesulitan berada dalam tingkat tersulit, maka *game* akan tetap pada tingkat kesulitan tersebut atau tidak dapat kembali ke tingkat awal (mudah). Dengan masalah tersebut, *game* terasa membosankan karena *game* mudah ditebak. Dengan mengimplementasikan logika *fuzzy* pada tingkat kesulitan *game* evakuasi korban banjir diharapkan tingkat kesulitan pada *game* ini dapat kembali ke tingkat termudah atau sedang saat tingkat kesulitan dalam *game* berada ditingkat tersulit.

3.2 Analisis *Game*

Dalam *game* ini terdapat 1 tahap permainan yaitu tahap pengevakuasian, yang dimana pemain akan mulai mengevakuasi korban-korban yang ada diatas rumah korban masing-masing. Pada tahap *game* pengevakuasian ini algoritma logika *fuzzy* dijalankan, dimana ada kondisi yang menentukan apakah saat pengevakuasian sudah dimulai tingkat kesulitan mudah, sedang, atau sulit. Kondisi itu berhubungan dengan jumlah korban yang dibawa pemain interval 0 hingga 6 korban dan cuaca mendung, gerimis, dan hujan deras yang akan terjadi secara acak dalam 10 detik.

Pada tahapan ini akan mendeskripsikan *game* yang akan dibangun. Pada bagian ini terdiri dari *storyline, gameplay,* jenis permainan, tingkat kesulitan, dan analisis *scoring.*

Tabel 3.1 *Game* Yang Dibangun

	***Game* Yang Dibangun**
Desain Grafis	2D
Karakter Utama	Karakter Pemuda
Pergerakan Karakter	Pergerakan karakter digerakan oleh pemain. Pergerakan hanya ke kanan dan ke kiri
Rintangan	Batang kayu, ular , dan arus listrik
Target	Mengumpulkan banyak korban untuk di evakuasi
Gameplay	Untuk menyelesaikan permainan pemain harus melewati halangan-halangan yang ada dan mengumpulkan banyak korban yang nantinya akan dibawa ke pos evakuasi.

3.2.1 Skenario *Game*

Skenario dalam *game tingkat kesulitan evakuasi korban banjir* dapat dilihat alur permainannya sebagai berikut:

1. Pemain akan langsung berada didaerah yang sudah terkena bencana banjir.
2. Pemain langsung berada disebuah perahu karet.
3. Pemain akan mengendalikan perahu karet dengan arah kanan dan kiri untuk mengevakuasi korban banjir dan juga untuk menghindar dari halangan-halangan yang ada di daerah banjir.
4. Pemain mendapatkan waktu 5 menit untuk mengevakuasi , dalam waktu tersebut pemain harus mengumpulkan banyak korban banjir
5. Jika pemain mengumpulkan sedikit korban dalam waktu 5 menit maka pemain diharuskan untuk mengulang ke awal pengevakuasian.
6. Jika mendapat banyak korban, pemain akan mendapatkan poin dari jumlah korban yang dibawa dan juga dari jumlah halangan yang terlewati.

3.2.2 Tahap Permainan

Dalam *game* yang akan dibangun terdapat 1 tahap permainan, yaitu:

1. Pengevakuasian Korban

   Pemain harus mengumpulkan banyak korban untuk di evakuasi ke pos

evakuasi korban banjir.

2. Tingkat kesulitan

Tantangan merupakan kejadian yang muncul dalam *game* dan dapat diselesaikan oleh pemain dengan berbagai macam cara. Terdapat tiga kondisi tingkat kesulitan yang terdapat pada tabel berikut :

Tabel 3.2 kondisi tingkat kesulitan

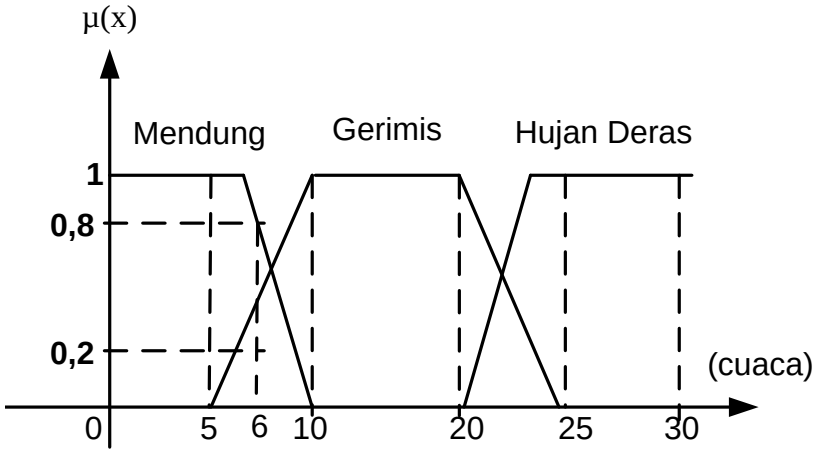
Aturan	Rintangan	Skenario Permainan
Mudah	Batang pohon	Pemain dihadapkan dengan batang pohon. Bila pemain terkena laju batang pohon maka health pemain akan berkurang.
Sedang	Batang pohon	Pemain dihadapkan dengan batang pohon. Bila pemain terkena laju batang pohon maka health pemain akan berkurang.
	Kemunculan ular	Pemain dihadapkan dengan adanya ular yang tiba-tiba muncul untuk mengganggu proses evakuasi , jika pemain terkena laju ular maka health pemain akan berkurang dan kuota korban akan berkurang satu.
Sulit	Batang pohon	Pemain dihadapkan dengan batang pohon. Bila pemain terkena laju batang pohon maka health pemain akan berkurang.
	Kemunculan ular	Pemain dihadapkan dengan adanya ular yang tiba-tiba muncul untuk mengganggu proses evakuasi , jika pemain terkena laju ular maka health pemain akan berkurang dan kuota korban akan berkurang satu.

	Kabel listrik yang terputus	Pemain dihadapkan dengan kabel listrik yang tiba-tiba terputus dan mengeluarkan arus listrik. Jika pemain terkena arus listrik maka akan mengurangi health pemain dan kuota korban akan berkurang dua

3.3 Analisis Masukan

Terdapat 2 variabel pada *game* ini, dimana variabel ini akan menentukan hasil *output* pada *game* . Adapun variabel yang digunakan adalah:

2. Cuaca yang muncul saat terjadinya pengevakuasian korban → Mendung, Gerimis, Hujan Deras. Cuaca ditentukan secara acak setiap 5 hingga 10 detik sekali, jadi ketika user memainkan *game*, tidak mesti cuaca sedang di mendung.
   a. Mendung (cuaca) → Kondisi dimana awan menghitam yang berkemungkinan akan terjadinya hujan.
   b. Gerimis (cuaca) → Kondisi dimana hujan turun dengan intensitas rendah
   c. Hujan Deras (cuaca) → Kondisi dimana hujan turun dengan intensitas tinggi.



Gambar 3.1 Grafik derajat keanggotaan penentuan kondisi cuaca

Fungsi Keanggotaan pada grafik derajat keanggotaan penentuan kondisi cuaca

a. $$\mu_{Mendung}(x) = \begin{cases} 1 & 0 \le x \le 5 \\ (10-x)/(10-5) & 5 \le x \le 10 \\ 0 & 10 \le x \le 30 \end{cases} \quad (3.1)$$

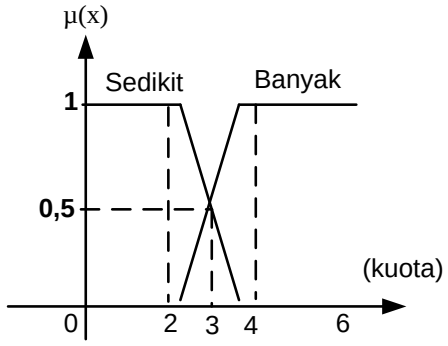
b. $$\mu_{Gerimis}(x) = \begin{cases} 0 & 0 \le x \le 5, 25 \le x \le 30 \\ (x-5)/(10-5) & 5 \le x \le 10 \\ 1 & 10 \le x \le 20 \\ (25-x)/(25-20) & 20 \le x \le 25 \end{cases} \quad (3.2)$$

c. $$\mu_{HujanDeras}(x) = \begin{cases} 0 & 0 \le x \le 20 \\ (x-20)/(25-20) & 20 \le x \le 25 \\ 1 & 25 \le x \le 30 \end{cases} \quad (3.3)$$

3. Kuota korban yang dibawa oleh pengevakuasi → Sedikit, Banyak Penentuan korban, berdasarkan korban yang dibawa oleh pemain, jika pemain tidak membawa korban atau hanya membawa 2 korban saja maka akan dianggap kuota sedikit, sedangkan 3 hingga 6 korban dianggap Penuh.

a. Sedikit (kuota) → Korban evakuasi hanya sedikit, dalam kuota perahu 6 orang , yang terevakuasi hanya 2 orang.

b. Penuh (kuota) → Korban evakuasi memenuhi kuota perahu.



Gambar 3.2 Grafik derajat keanggotaan penentuan kuota korban

Fungsi Keanggotaan pada grafik derajat keanggotaan penentuan kuota korban

a. $$\mu_{Sedikit}(x) = \begin{cases} 1 & 0 \le x \le 2 \\ (x-2)/(4-2) & 2 \le x \le 4 \\ 0 & 4 \le x \le 6 \end{cases} \quad (3.4)$$

b. $$\mu_{Banyak}(x) = \begin{cases} 0 & 0 \le x \le 2 \\ (10-x)/(10-5) & 2 \le x \le 4 \\ 1 & 4 \le x \le 6 \end{cases} \quad (3.5)$$

3.3.2 Analisis *Scoring*

Pada *game* yang akan dibangun terdapat poin yang diberikan setiap menyelesaikan permainan. Poin akan hilang atau kembali ke keadaan awal jika pemain tidak menyelesaikan misi dalam setiap tahapnya. Jumlah *poin* yang didapat dari setiap misi yang dilakukannya sebagai berikut:

a. Melewati halangan yang ada = +50
b. Membawa korban =+100

3.4 Analisis Algoritma Logika *Fuzzy*

Algoritma logika *fuzzy* akan diterapkan pada kejadian-kejadian saat evakuasi korban banjir, dimana saat evakuasi akan terjadi cuaca yang berubah, halangan yang muncul, kuota korban dan tingkat kesulitan yang akan dihadapi pemain. Berikut adalah ilustrasi logika *fuzzy* yang dapat dilihat pada tabel *Input* dan *Output* Algoritma logika *fuzzy*:

Tabel 3.3 *Input* dan *Output* Algoritma logika *fuzzy* saat Perjalanan Evakuasi

Cuaca	Kuota	Aturan
Mendung	Sedikit	Mudah
Mendung	Banyak	Sedang
Gerimis	Sedikit	Mudah
Gerimis	Banyak	Sulit
Hujan Deras	Sedikit	Sedang
Hujan Deras	Banyak	Sulit

Pada tabel *Input* dan *output* yang digunakan untuk menentukan tingkat kesulitan yang akan dihadapi pemain. *Input* dan *Output* tersebut memiliki tingkat

kesulitan yang berbeda, berikut penjelasan dari masing-masing *Input dan output* tersebut Adapun himpunan *fuzzy* pada *game* ini adalah nya adalah:

1. Mudah
2. Sedang
3. Sulit

Rule logika *fuzzy* Saat Perjalanan Evakuasi 3 x 2 = 6

1. **IF** cuaca is Mendung **AND** kuota is Sedikit **THEN** Tantangan is Mudah
2. **IF** cuaca is Mendung **AND** kuota is Banyak **THEN** Tantangan is Sedang
3. **IF** cuaca is Gerimis **AND** kuota is Sedikit **THEN** Tantangan is Mudah
4. **IF** cuaca is Gerimis **AND** kuota is Banyak **THEN** Tantangan is Sulit
5. **IF** cuaca is Hujan Deras **AND** kuota is Sedikit **THEN** Tantangan is Sedang
6. **IF** cuaca is Hujan Deras **AND** kuota is Banyak **THEN** Tantangan is Sulit

Penentuan Kondisi Cuaca

Penjelasan cuaca merupakan nilai yang dihasilkan secara acak diperkirakan setiap 10 detik yang nantinya akan digunakan untuk menentukan kondisi hujan pada saat pengevakuasian , yang terdiri dari mendung, gerimis, dan hujan deras.

Penentuan Jumlah Korban yang Dibawa

Penjelasan kuota merupakan nilai yang dihasilkan dari banyaknya korban yang dibawa dimana korban yang dibawa terdiri dari Sedikit dan Banyak.

Contoh kasus

1. Misal didapatkan nilai cuaca sebesar 6. Pada proses ini maka nilai crips *Input* cuaca 6 dikonversi ke nilai *fuzzy* dengan cara, cuaca 6 berada pada nilai linguistik Mendung dan Gerimis. Derajat keanggotaan untuk gerimis dihitung menggunakan rumus yang dapat dilihat pada gambar 2.3. Maka didapat derajat keanggotaan cuaca mendung dengan rumus (gambar 2.3) dimana c = 5 dan d = 10 sehingga derajat keanggotaan Mendung = - (10 – 6) / (10 – 5) = 4/5. dimana a=5 dan b= 10 sehingga derajat keanggotaan Gerimis =(10-5) / (10-5) = 1/5. Dari hasil diatas akan didapat grafik derajat keanggotaan dari kasus Penentuan Kondisi Cuaca yang dapat dilihat pada

gambar 3.1

2. Misal didapatkan nilai kuota sebesar 3. Pada proses ini maka nilai crips *Input* kuota 3 dikonversi ke nilai *fuzzy* dengan cara, kuota 3 berada pada nilai linguistik sedikit dan Banyak. Derajat keanggotaan untuk Sedikit dihitung menggnakan rumus (gambar 2.3) dimana c = 2 dan d = 4 sehingga derajat keanggotaan Sedikit = - (4 – 3) / (4 – 2) = ½. Derajat keanggotaan untuk Penuh menggunakan rumus (gambar 2.3) dimana a=2 dan b= 4 sehingga derajat keanggotaan Banyak =(3-2) / (4-2) = ½. Maka akan didapat grafik derajat keanggotaan dari kasus Penentuan Jumlah Korban yang dibawa yang dapat dilihat pada gambar 3.2.

Dari hasil perhitungan tersebut maka terdapat aturan yang telah ada pada tabel *Input* dan *output* menggunakan aturan *Conjunction* (∧) dengan memilih derajat keanggotaan minimum sehingga diperoleh **:**

1. **IF** cuaca is Mendung (4/5) **AND** kuota is Sedikit (1/2) **THEN** Tantangan is Mudah (1/2)
2. **IF** cuaca is Mendung (4/5) **AND** kuota is Banyak (1/2) **THEN** Tantangan is Sedang (1/2)
3. **IF** cuaca is Gerimis (0) **AND** kuota is Sedikit (1/2) **THEN** Tantangan is Mudah (1/5)
4. **IF** cuaca is Gerimis (0) **AND** kuota is Banyak (1/2) **THEN** Tantangan is Sulit (1/5)
5. **IF** cuaca is Hujan Deras (0)**AND** kuota is Sedikit (1/2) **THEN** Tantangan is Sedang (1/2)
6. **IF** cuaca is Hujan Deras (0) **AND** kuota is Banyak (1/2) **THEN** Tantangan is Sulit (1/2)

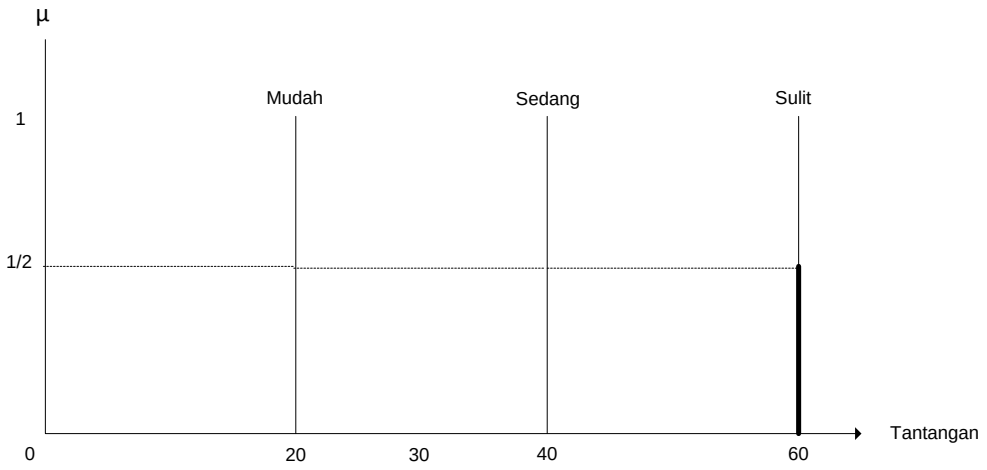
Dari hasil diatas maka akan melanjutkan menggunakan proses inference menggunakan model Sugeno dengan Aturan *Disjunction* (v) dimana memilih derajat keangotaan maksimum Sehingga dihasikan , yaitu :

1. Tantangan is Mudah (1/2) ∨ tantangan is Mudah (1/2) Dihasilkan tantangan is Mudah (1/2)
2. Tantangan is Sedang (1/2) ∨ tantangan is Sedang (1/5) dihasilkan tantangan

is Sedang (1/2)

3. Tantangan is Sulit (0) ∨ tantangan is Sulit (1/2) dihasilkan tantangan is Sulit (1/2)

Maka akan di dapat grafik derajat keanggotaan model sugeno pada gambar 3.3 :



Gambar 3.3 Grafik Derajat keanggotaan model sugeno

Dilanjutkan dengan proses *defuzzyfication* menggunakan model sugeno Dengan menggunakan weighted average method untuk proses *defuzzyfication* maka didapat hasil dengan rumus 2.1. Maka akan didapatkan hasil perhitungan sebagai berikut :

$$y* = \frac{\frac{1}{2}(20)+\frac{1}{2}(40)+\frac{1}{2}(60)}{\frac{1}{2}+\frac{1}{2}+\frac{1}{2}} = 60 \qquad (3.6)$$

dari kesimpulan proses de*fuzzy*fication menggunakan model sugeno

1. Nilai yang didapat dari proses *defuzzyfication* tersebut selanjutnya digunakan untuk penentuan keputusan dengan aturan sebagai berikut
2. Mudah 0-20
3. Sedang 21-40
4. Sulit 41-60

Hasil nilai dari keputusan = 60 adalah Sulit.

Dari hasil tingkat tantangan Sulit maka akan muncul kejadian-kejadian pada tabel 3.2.

3.5 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan dalam sebuah sistem merupakan perintah-perintah yang diberikan kepada perangkat keras agar bisa saling berinteraksi diantara keduanya.

1. Pengembang

Spesifikasi perangkat lunak yang digunakan pengembang untuk membangun *game* ini dapat dilihat pada tabel berikut :

Tabel 3.4 Spesifikasi Perangkat Lunak Pengembang

Nama Perangkat Lunak	Spesifikasi
Sistem Operasi	Microsoft Windows 7 SP1
Tools	- Unity 3D 4.3 - Adobe Photoshop CS6 - DirectX 11

2. Pengguna

Spesifikasi minimum perangkat lunak yang dibutuhkan pengguna untuk memainkan *game* ini dapat dilihat pada tabel berikut :

Tabel 3.5 Spesifikasi Minimum Perangkat Lunak Pengguna

Nama Perangkat Lunak	Spesifikasi
Sistem Operasi	Microsoft Windows XP SP2
Tools	- Unity Web Player - DirectX 9

3. Pengembang

Spesifikasi perangkat keras yang digunakan pengembang untuk membangun *game* ini dapat dilihat pada tabel berikut:

Tabel 3.6 Spesifikasi Perangkat Keras Pengembang

Nama Perangkat	Spesifikasi
Processor	Intel Core 2 Duo 2.20 GHz
RAM	4 GB
VGA Card	250 MB

Harddisk	500 GB
Monitor	Resolusi 1366 x 768
Mouse dan *keyboard*	Standar

4. Pengguna

Berdasarkan *system requirement* yang ada pada *website* Unity3D spesifikasi minimum perangkat keras bagi pengguna untuk dapat memainkan *game* ini dengan baik dapat dilihat pada tabel berikut :

Tabel 3.7 Spesifikasi Minimum Perangkat Keras Pengguna

Nama Perangkat	**Spesifikasi**
Processor	1.80 GHz
RAM	2 GB
VGA Card	250 MB
Harddisk	Tersedia 100 MB
Monitor	Resolusi 800 x 600
Mouse dan *keyboard*	Standar

Dengan spesifikasi yang lebih rendah maka pengguna tidak akan memainkan aplikasi *game* dengan maksimal seperti mengalami ketersendatan atau gambar yang kabur.

3.5.1 Analisis Pengguna

Pada analisis pengguna ini akan mencakup analisis beberapa parameter terhadap calon pengguna dari aplikasi.

1. *User Knowledge* and *Experience* dari target pengguna yang akan menggunakan *game*

Game yang akan dibangun ini bisa digunakan oleh kalangan apapun, tetapi pengetahuan dan pengalaman akan memudahkan pengguna dalam penggunaannya. Berikut ini klasifikasi *knowledge and experience* dari aplikasi:

Tabel 3.8 Analisis klasifikasi *knowledge and experience*

Educational Level	*Game* ini bisa digunakan oleh berbagai kalangan seperti pelajar, mahasiswa hingga masyarakat awam
Reading Level	*Game* ini bisa digunakan oleh berbagai *level* pendidikan dengan *reading level* rendah
Computer Literacy	*Game* ini bisa digunakan oleh pengguna yang memiliki kemampuan komputer yang rendah
Task Experience	*Game* ini bisa digunakan oleh pengguna dengan pengalaman *game* yang rendah
System Experience	*Game* ini bisa digunakan oleh pengguna dengan pengalaman komputer yang rendah
Application Experience	*Game* ini bisa digunakan dalam semua sistem operasi Windows minimal XP
Native Language	*Game* ini menggunakan bahasa Indonesia dan bahasa inggris
Use of Other System	*Game* ini bisa dijalankan tanpa perlu install aplikasi lain

2. *User Physical Characteristic*

Keadaan fisik seseorang mungkin akan berpengaruh pada penggunaan aplikasi *game* ini. Ada hal-hal yang harus diperhatikan juga terhadap pengguna dari karakteristik fisiknya untuk dapat menggunakan aplikasi ini yaitu, *color blind, handedness,* dan *gender*.

Tabel 3.9 Analisis Users Physical Characteristic

Age	9 tahun ke atas
Gender	Pria dan wanita
Handedness	Kanan dan kiri
Disabilities	Pengguna yang buta tidak akan bisa menggunakan *game* ini, untuk yang penglihatannya kurang dan tuli masih dapat memainkan tetapi tidak akan optimal.

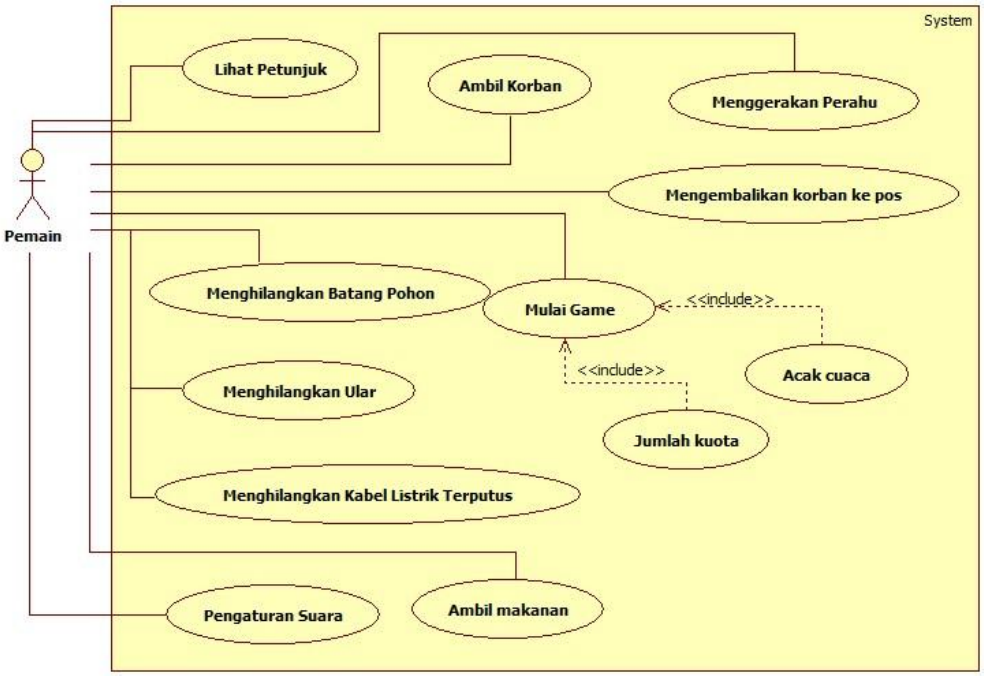
3.5.2 Analisis Kebutuhan Fungsional

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. Analisis

kebutuhan fungsional dimodelkan dengan menggunakan UML *(Unified Modeling Language)*.

3.5.3 Use Case Diagram

Diagram *use case* menggambarkan hubungan antara aktor dan sistem yang dibangun. Perancangan proses-proses yang terdapat pada *game* yang akan dibangun akan digambarkan dengan diagram *use case*. Interaksi antara aktor dengan aplikasi *game* yang akan dibangun digambarkan pada *diagram use case* pada Gambar 3.4



Gambar 3.4 *Use case Diagram Game* yang akan dibangun

3.5.4 Definisi Aktor

Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem sistem yang akan dibangun [19]. Definisi aktor diterangkan pada tabel berikut :

Tabel 3.10 Definisi Aktor

No	Aktor	Deskripsi
1.	Pemain	Pemain atau orang yang memainkan *game*

3.5.5 Definisi *Use case*

Use case merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor [19]. Definisi *use case* diterangkan pada tabel 3.11 :

Tabel 3.11 Definisi *Use case*

No.	*Use case*	Deskripsi
1.	Mulai *Game*	Proses untuk masuk dalam game evakuasi
2.	Acak Cuaca	Proses untuk mengacak (*random*) kondisi cuaca yang akan muncul pada *game*
3.	Jumlah Kuota	Proses untuk menghitung jumlah kuota yang ada dalam *game*
4.	Menggerakan Perahu	Proses untuk menggerakan perahu dalam *game*
5.	Ambil Korban	Proses untuk mengambil korban dalam *game*
6.	Mengembalikan korban ke pos	Proses untuk mengembalikan korban ke pos dalam *game*
7.	Ambil Makanan	Proses untuk mengambil makanan dalam *game*
8.	Menghilangkan Batang Pohon	Proses untuk mengklik halangan berupa batang pohon yang ada dalam *game*
9.	Menghilangkan Ular	Proses untuk mengklik halangan berupa ular yang ada dalam *game*
10.	Menghilangkan Kabel Listrik yang Terputus	Proses untuk mengklik halangan berupa kabel listrik yang terputus yang ada dalam *game*

11.	Lihat Petunjuk	Proses untuk melihat petunjuk atau panduan cara bermain
12.	Pengaturan Suara	Proses untuk mengatur suara menjadi ON atau OFF

3.5.6 Skenario *Use case*

Setiap *use case* dilengkapi dengan skenario. Skenario *use case* adalah alur jalannya proses *use case* dari sisi aktor dan sistem [19]. Berikut adalah skenario jalannya masing-masing *use case* yang telah didefinisikan sebelumnya :

1. Skenario *Use case* Mulai *Game*

Tabel 3.12 Scenario *Use case* Mulai *Game*

Nama *Use case*	Mulai Game	
Aktor	*User*	
Type	*Primary, Essential*	
Tujuan	Menampilkan layar utama pada *game* dan *check Fuzzy*	
Kondisi Awal	Aplikasi *Game* yang belum dijalankan	
Aksi Aktor		Reaksi Sistem
Skenario Normal		
1. Mulai Aplikasi		
		2. Menampilkan Menu Utama
3. Pilih Menu *Play*		
		4. Menampilkan Layar Utama *Game*
		5. Acak Cuaca dan Jumlah Kuota
		6. Hitung Nilai Derajat Keanggotaan *Fuzzy*
		7. Tantangan Muncul

2. Skenario *Use case* Acak Cuaca

Tabel 3.13 Scenario *Use case* Acak Cuaca

Nama *Use case*	Acak Cuaca
Aktor	*User*
Type	*Primary, Essential*
Tujuan	Menampilkan Acak Cuaca pada Game
Kondisi Awal	Cuaca yang muncul pada saat *game* dimulai

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Pilih menu *Play*	
	2. Menampilkan Layar Utama *Game*
	3. Acak Cuaca
	4. Cuaca Muncul

3. Skenario *Use case* Jumlah Kuota

Tabel 3.14 Scenario *Use case* Jumlah Kuota

Nama *Use case*	Jumlah Kuota
Aktor	*User*
Type	*Primary, Essential*
Tujuan	Menampilkan Jumlah Kuota yang ada pada *game*
Kondisi Awal	Pemain tidak memiliki kuota dalam perahu

Aksi Aktor	Reaksi Sistem
Skenario Normal	

1. Pilih menu Play	
	2. Menampilkan Layar Utama *Game*
	3. Check Kuota pada NPC

4. Skenario *Use case* Menggerakan Perahu

Tabel 3.15 Scenario *Use case* Menggerakan Perahu

Nama *Use case*	Menggerakan Perahu
Aktor	*User*
Type	*Primary, Essential*
Tujuan	Menampilkan cara menggerakan perahu dalam game
Kondisi Awal	Pemain berada diatas perahu
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Tekan tombol A jika ingin perahu bergerak ke kiri	
	2. Gerak Perahu ke kiri
3. Tekan tombol D jika ingin perahu bergerak ke kanan	
	4. Gerak Perahu ke Kanan

5. Skenario *Use case* Ambil Korban

Tabel 3.16 Scenario *Use case* Ambil Korban

Nama *Use case*	Ambil Korban
Aktor	*User*

Type	*Primary, Essential*
Tujuan	Mengambil korban dalam game
Kondisi Awal	Pemain berada dekat dengan korban

Aksi Aktor	Reaksi Sistem
Skenario *N*ormal	
1. Klik dan Drag korban ke perahu	
	2. Kuota korban bertambah
	3. Nilai bertambah

6. Skenario *Use case* Mengembalikan Korban ke Pos

Tabel 3.17 Scenario *Use case* Mengembalikan korban ke pos

Nama *Use case*	Mengembalikan korban ke pos
Aktor	*User*
Type	*Primary, Essential*
Tujuan	Mengurangi kuota korban yang dibawa
Kondisi Awal	Pemain memiliki lebih dari satu kuota korban

Aksi Aktor	Reaksi Sistem
Skenario *N*ormal	
1. Pemain menuju pos evakuasi	
	2. Kuota korban berkurang
	3. Nilai bertambah

7. Skenario *Use case* Ambil Makanan

Tabel 3.18 Scenario *Use case* Ambil Makanan

Nama *Use case*	Ambil Makanan

Aktor	*User*
Type	*Primary, Essential*
Tujuan	Menambah *health* pada pemain pada *game*
Kondisi Awal	Pemain dekat dengan makanan yang ada pada *game*

Aksi Aktor	Reaksi Sistem
Skenario *N*ormal	
1. Klik dan Drag makanan ke perahu	
	2. *Health* pemain bertambah

8. Skenario *Use case* Menghilangkan Batang Pohon

Tabel 3.19 Scenario *Use case* Menghilangkan Batang Pohon

Nama *Use case*	Menghilangkan Batang Pohon
Aktor	*User*
Type	*Primary, Essential*
Tujuan	Menghilangkan tantangan batang pohon yang muncul
Kondisi Awal	Tantangan batang pohon muncul pada *game*

Aksi Aktor	Reaksi Sistem
Skenario *N*ormal	
1. Menghilangkan batang pohon yang muncul	
	2. Batang pohon menghilang
	3. Nilai bertambah

9. Skenario *Use case* Menghilangkan Ular

Tabel 3.20 Scenario *Use case* Menghilangkan Ular

Nama *Use case*	Menghilangkan Ular
Aktor	*User*
Type	*Primary, Essential*
Tujuan	Menghilangkan tantangan ular yang muncul
Kondisi Awal	Tantangan ular muncul pada *game*

Aksi Aktor	Reaksi Sistem
Skenario *N*ormal	
1. Menghilangkan ular yang muncul	
	2. Ular menghilang
	3. Nilai bertambah

10. Skenario *Use case* Menghilangkan Kabel Listrik yang Terputus

Tabel 3.21 Scenario *Use case* Menghilangkan Kabel Listrik yang Terputus

Nama *Use case*	Menghilangkan Kabel Listrik yang Terputus
Aktor	*User*
Type	*Primary, Essential*
Tujuan	Menghilangkan tantangan Kabel Listrik yang Terputus yang muncul
Kondisi Awal	Tantangan Kabel Listrik yang Terputus muncul pada *game*

Aksi Aktor	Reaksi Sistem
Skenario *N*ormal	
1. Menghilangkan Kabel Listrik yang Terputus yang muncul	
	2. Kabel listrik yang terputus menghilang
	3. Nilai bertambah

11. Skenario *Use case* Lihat Petunjuk

Tabel 3.22 Scenario *Use case* Lihat Petunjuk

Nama Use case	Lihat Petunjuk
Aktor	User
Type	Primary, Essential
Tujuan	Menampilkan isi menu petunjuk permainan
Kondisi Awal	Pemain berada pada menu utama game

Aksi Aktor	**Reaksi Sistem**
Skenario *N*ormal	
	1. Menampilkan *Menu* utama
2. Pilih *Menu* lihat petunjuk	
	3. Menampilkan layar petunjuk permainan
4. Klik tanda panah jika ingin melihat kembali isi petunjuk yang lain	
	5. Menampilkan Isi Petunjuk Yang Lain
6. Klik kembali jika ingin kembali ke menu utama	
	7. Menampilkan Menu Utama

12. Skenario *Use case* Pengaturan Suara

Tabel 3.23 Scenario *Use case* Pengaturan Suara

Nama *Use case*	Pengaturan Suara
Aktor	*User*

Type	*Primary, Essential*
Tujuan	Menampilkan pengatur suara pada game
Kondisi Awal	Tampilan pengatur ON atau OFF pada layar

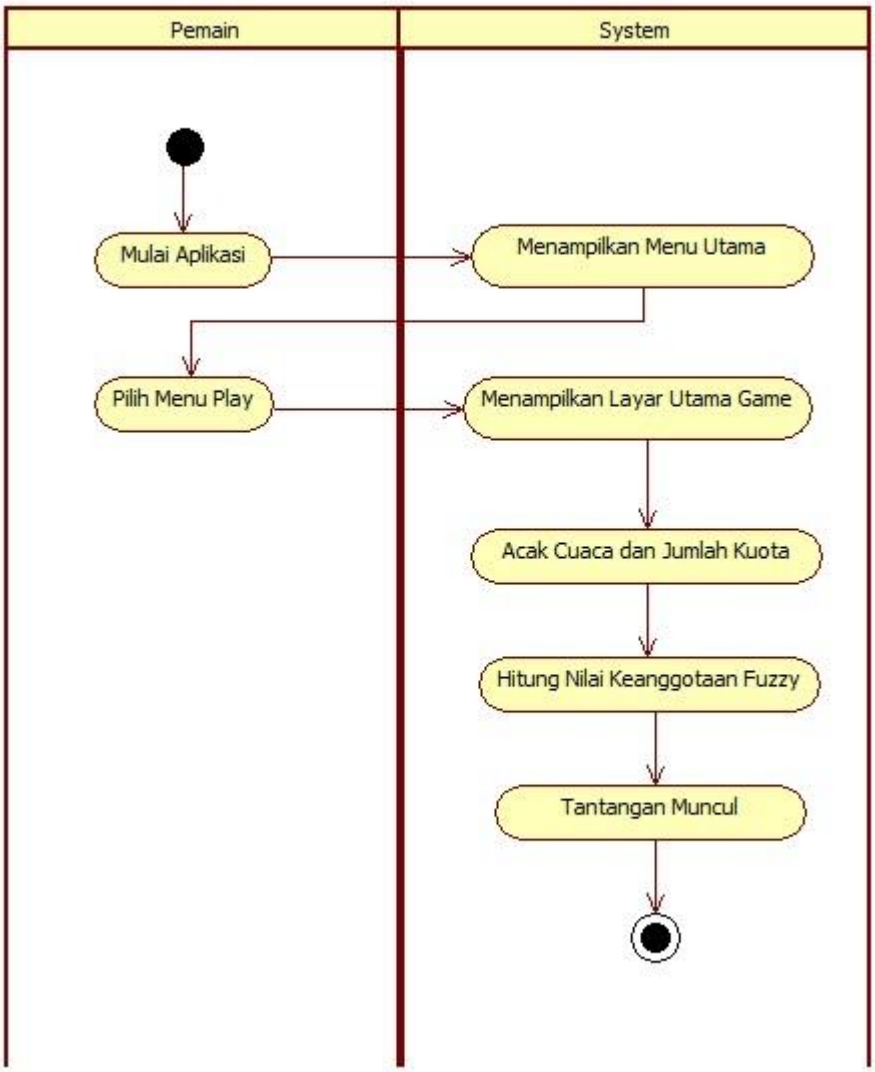
Aksi Aktor	**Reaksi Sistem**
Skenario *N*ormal	
	1. Menampilkan *Menu* utama
	2. Inisialisasi suara = *ON* atau *TRUE*
3. Klik tombol *OFF* pada layar menu utama	
	4. Suara = *OFF* atau *FALSE*

3.5.7 *Activity Diagram*

Diagram *activity* menggambarkan proses-proses yang terjadi mulai aktivitas dimulai sampai aktivitas berhenti pada sistem yang sedang dirancang. Bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir.

1. *Activity Diagram* Mulai *Game*

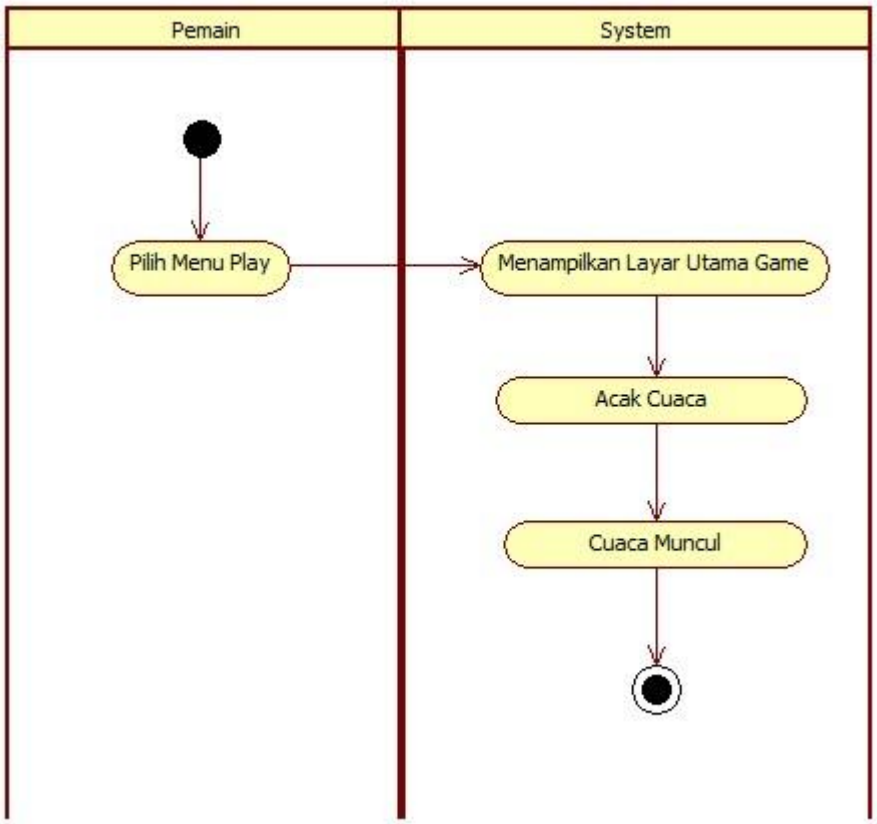
Proses ini terjadi saat pemain memilih *Menu* play pada *Menu* utama. Selanjutnya, sistem akan masuk ke *scene game*. Berikut adalah *activity diagram* mulai *game* yang terdapat pada gambar 3.5



Gambar 3.5 *Activity Diagram* Mulai *Game*

2. *Activity Diagram* Acak Cuaca

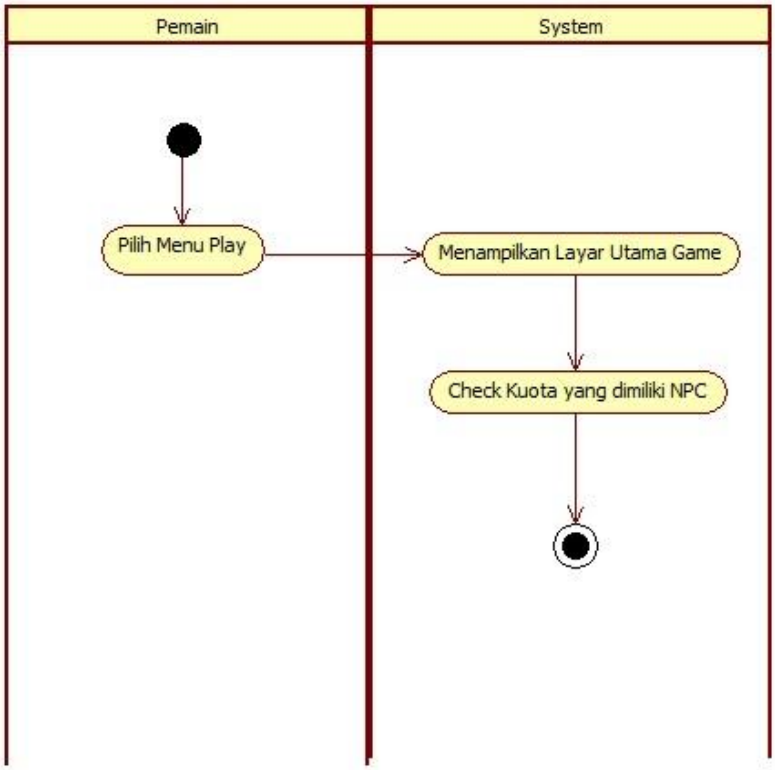
Proses ini terjadi saat pemain masuk dalam *game*. Selanjutnya, sistem akan melakukan pengacakan cuaca. Berikut adalah *activity diagram* Acak Cuaca yang terdapat pada gambar 3.6



Gambar 3.6 *Activity Diagram* Acak Cuaca

3. *Activity Diagram* Jumlah Kuota

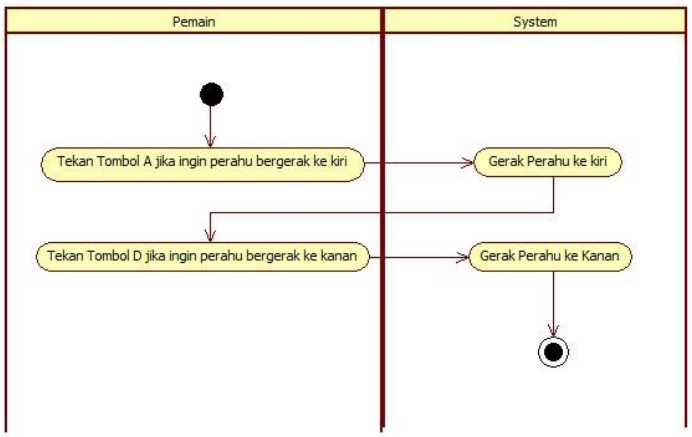
Proses ini terjadi saat pemain masuk dalam *game*. Selanjutnya, sistem akan melakukan pengecekan jumlah kuota. Berikut adalah *activity diagram* Jumlah Kuota yang terdapat pada gambar 3.7



Gambar 3.7 *Activity Diagram* Jumlah Kuota

4. *Activity Diagram* Menggerakan Perahu

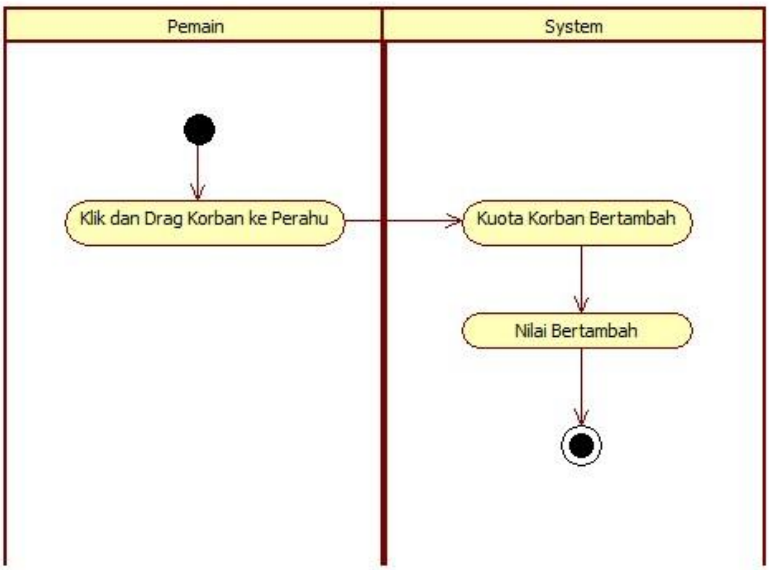
Proses ini terjadi saat pemain masuk dalam *game*. Selanjutnya, pemain akan melakukan aksi untuk menggerakan perahu. Berikut adalah *activity diagram* Menggerakan Perahu yang terdapat pada gambar 3.8



Gambar 3.8 *Activity Diagram* Menggerakan Perahu

5. *Activity Diagram* Ambil Korban

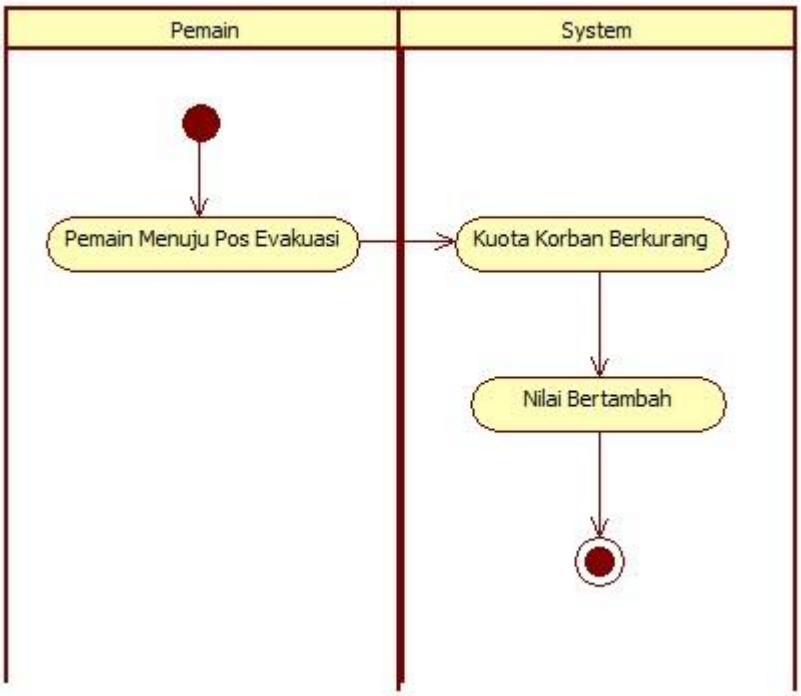
Proses ini terjadi saat pemain masuk dalam *game*. Selanjutnya, pemain akan melakukan aksi mengambil korban evakuasi. Berikut adalah *activity diagram* Ambil Korban yang terdapat pada gambar 3.9



Gambar 3.9 *Activity Diagram* Ambil Korban

6. *Activity Diagram* Mengembalikan Korban ke Pos

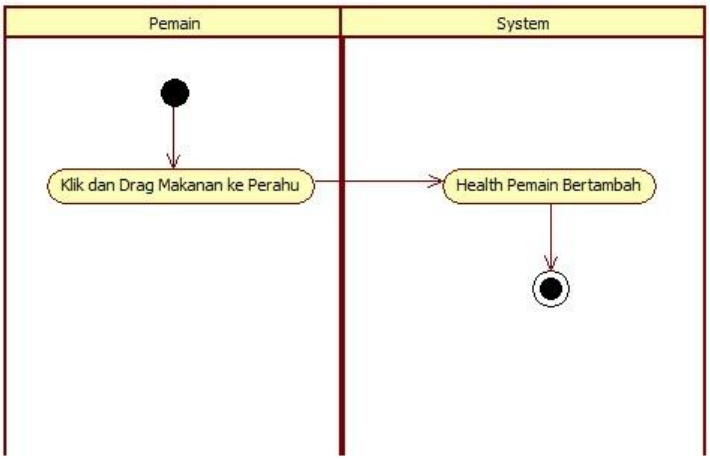
Proses ini terjadi saat pemain masuk dalam *game*. Selanjutnya, pemain akan melakukan aksi mengurangi kuota korban. Berikut adalah *activity diagram* Mengembalikan Korban ke Pos yang terdapat pada gambar 3.10



Gambar 3.10 *Activity Diagram* Mengembalikan Korban ke Pos

7. *Activity Diagram* Ambil Makanan

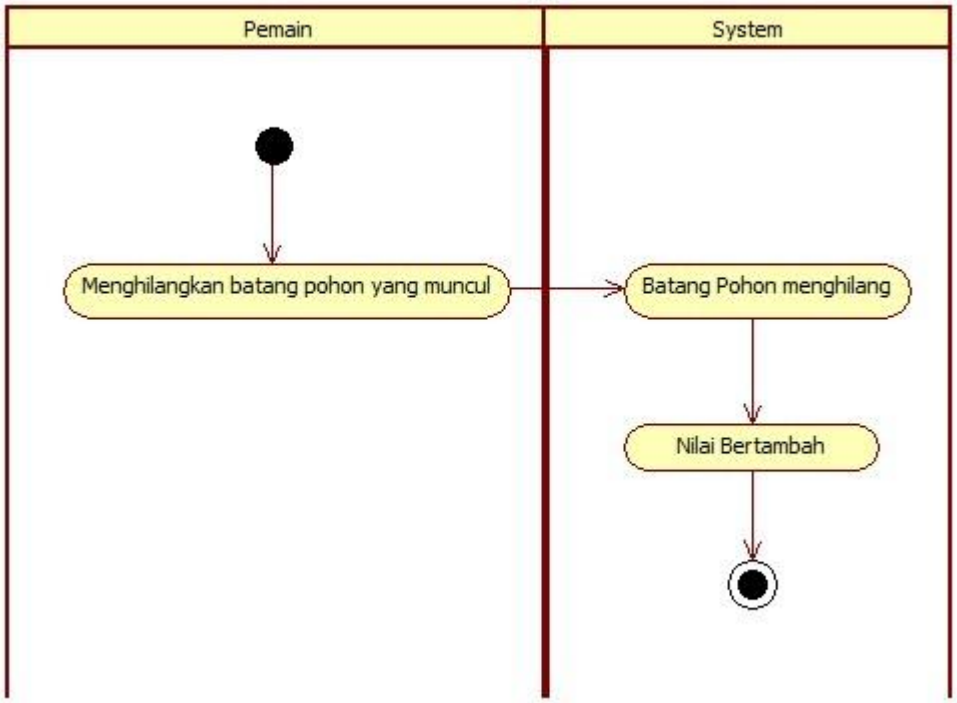
Proses ini terjadi saat pemain masuk dalam *game*. Selanjutnya, pemain akan melakukan aksi pengambilan makanan untuk menambah *health*. Berikut adalah *activity diagram* Ambil Makanan yang terdapat pada gambar 3.11



Gambar 3.11 *Activity Diagram* Ambil Makanan

8. *Activity Diagram* Menghilangkan Batang Pohon

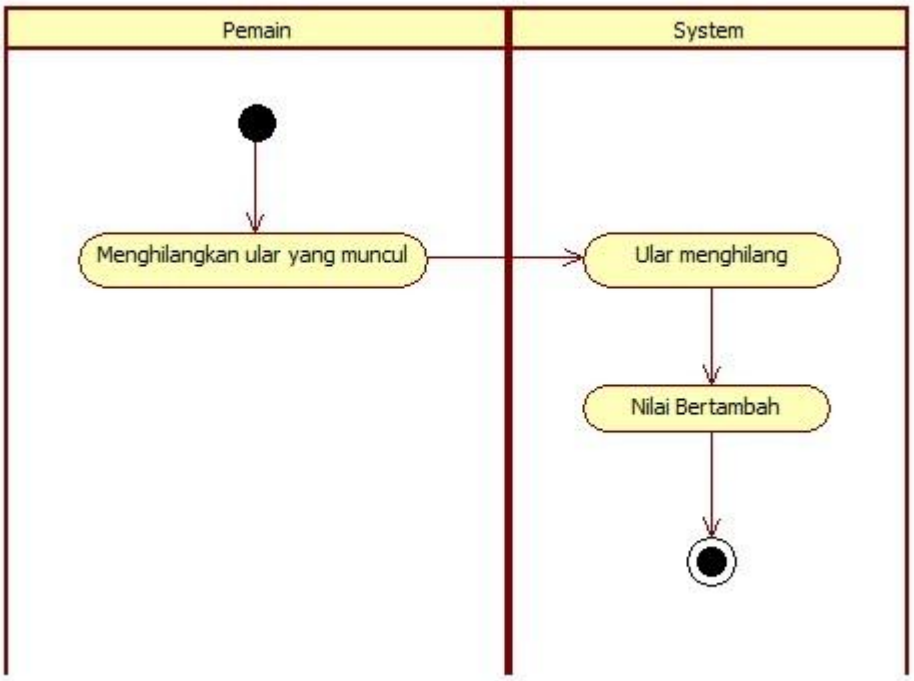
Proses ini terjadi saat pemain masuk dalam *game*. Selanjutnya, pemain akan melakukan aksi menghilangkan tantangan berupa batang pohon yang ada pada *game*. Berikut adalah *activity diagram* Menghilangkan Batang Pohon yang terdapat pada gambar 3.12



Gambar 3.12 *Activity Diagram* Menghilangkan Batang Pohon

9. *Activity Diagram* Meghilangkan Ular

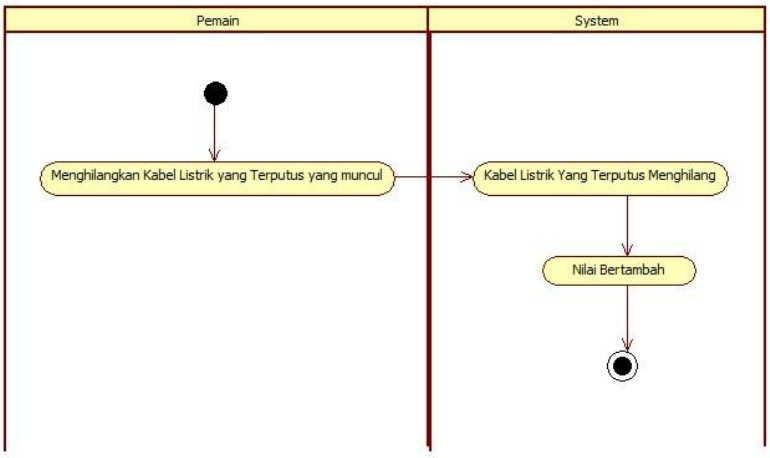
Proses ini terjadi saat pemain masuk dalam *game*. Selanjutnya, pemain akan melakukan aksi menghilangkan tantangan berupa seekor ular yang ada pada *game*. Berikut adalah *activity diagram* Menghilangkan Ular yang terdapat pada gambar 3.13



Gambar 3.13 *Activity Diagram* Menghilangkan Ular

10. *Activity Diagram* Klik Kabel Listrik yang Terputus

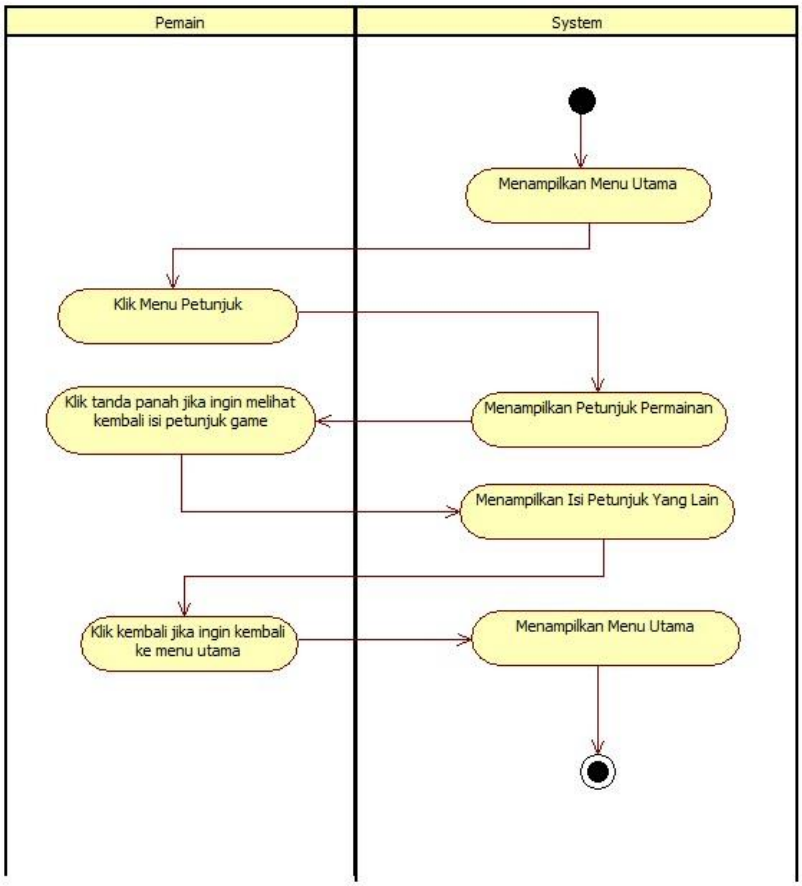
Proses ini terjadi saat pemain masuk dalam *game*. Selanjutnya, pemain akan melakukan aksi menghilangkan tantangan berupa kabel liistrik yang terputus yang ada pada *game.* Berikut adalah *activity diagram* Klik Kabel Listrik yang Terputus yang terdapat pada gambar 3.14



Gambar 3.14 *Activity Diagram* Menghilangkan Kabel Listrik yang Terputus

11. *Activity Diagram* Lihat Petunjuk

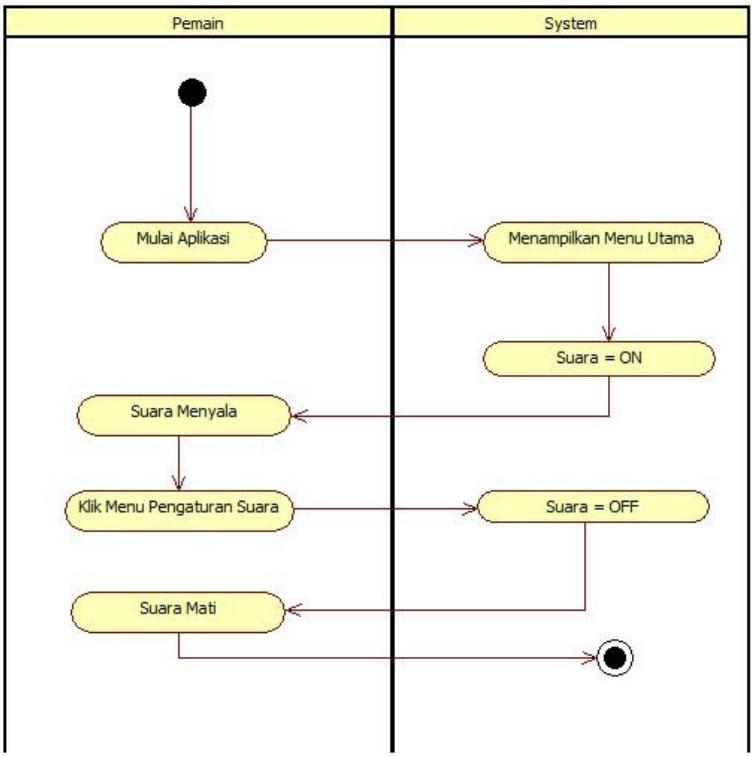
Proses ini terjadi saat pemain memilih *Menu* petunjuk pada *Menu* utama. Selanjutnya, sistem akan menampilkan informasi petunjuk permainan. Berikut adalah *activity diagram* lihat petunjuk yang terdapat pada gambar 3.15



Gambar I.15 *Activity Diagram* Lihat Petunjuk

12. *Activity Diagram* Pengaturan Suara

Proses ini terjadi saat pemain memilih *Menu* Pengaturan Suara pada *Menu* utama. Selanjutnya, sistem akan menampilkan menu Pengauran suara pada *game*. Berikut adalah *activity diagram* pengaturan suara yang terdapat pada Gambar 3.16

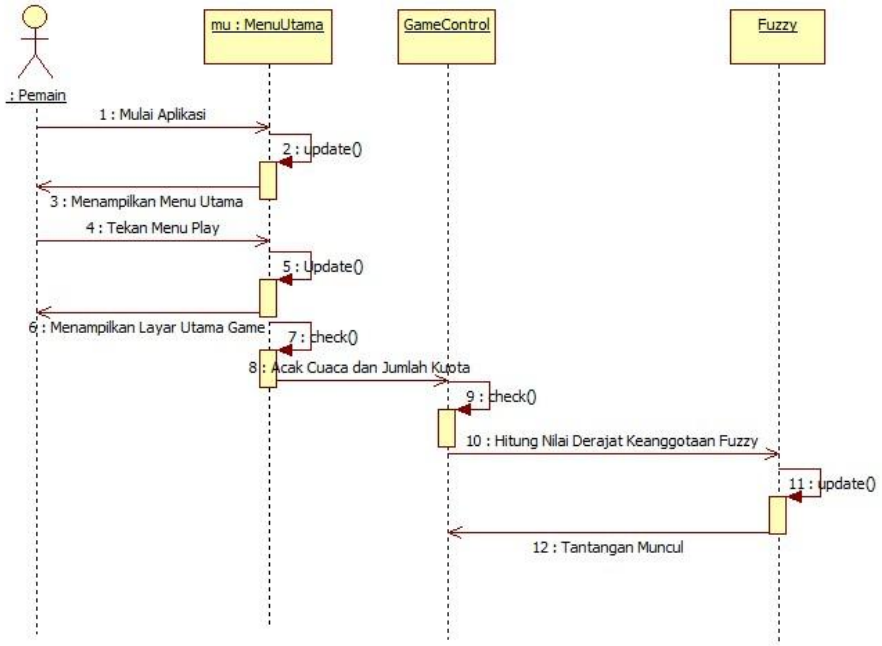


Gambar I.16 *Activity Diagram* Pengaturan Suara

3.5.8 *Sequence Diagram*

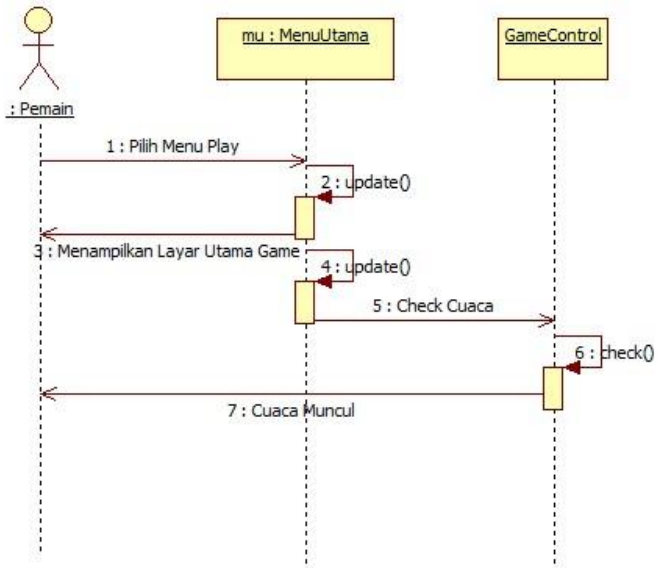
Sequence Diagram merupakan penggambaran dari interaksi antar objek di dalam dan sekitar sistem (termasuk pengguna, *display*, dan sebagainya). *Sequence Diagram* terdiri atas dimensi vertical (waktu) dan dimensi horizontal (objek-objek yang terkait). Berikut adalah beberapa *diagram sequence* dari *game* yang akan dibangun :

1. *Sequence Diagram* Mulai *Game*



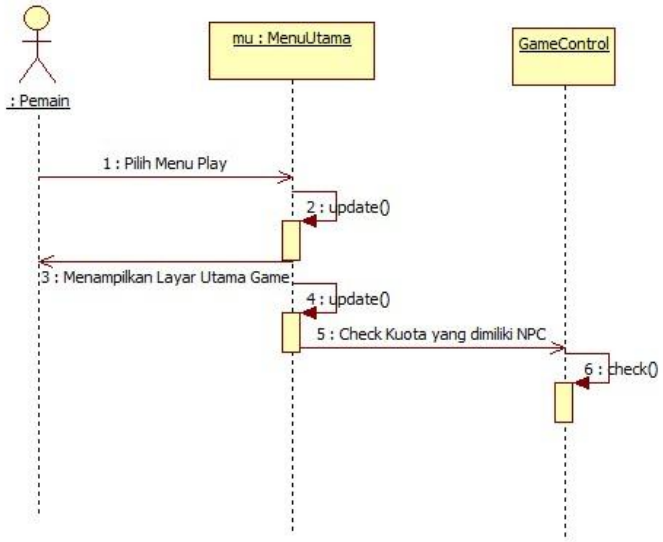
Gambar I.17 *Sequence Diagram* Mulai *Game*

2. *Sequence Diagram* Acak Cuaca



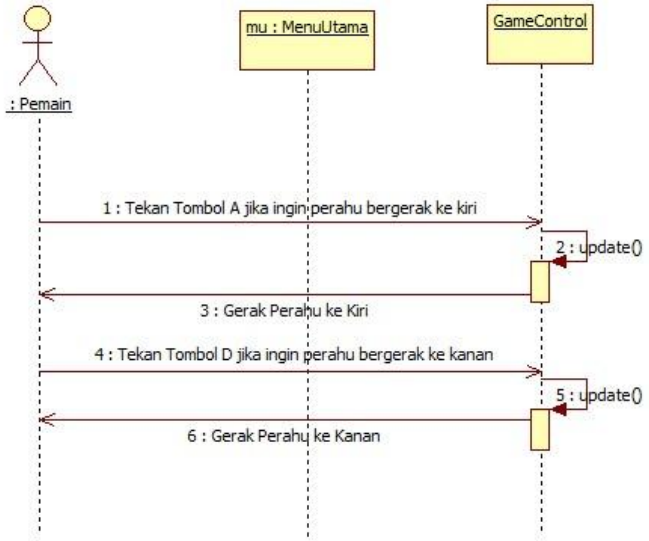
Gambar I.18 *Sequence Diagram* Acak Cuaca

3. *Sequence Diagram* Jumlah Kuota



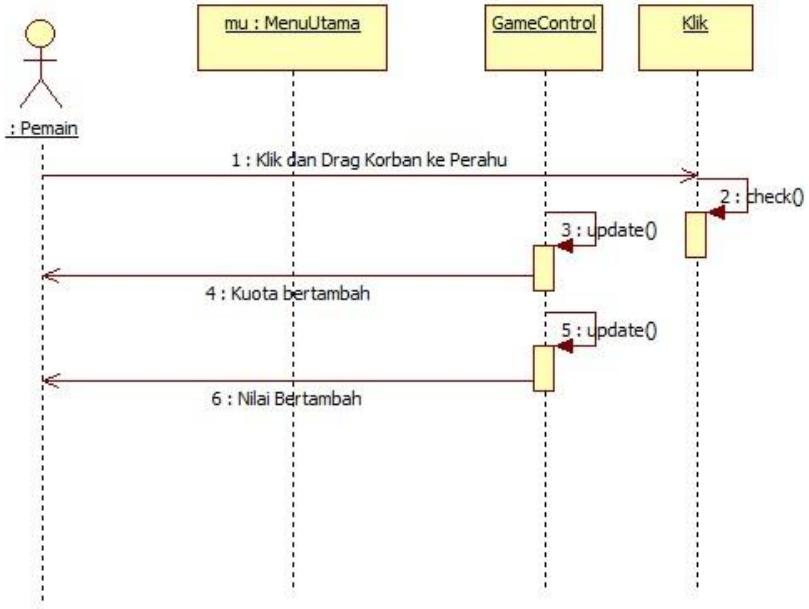
Gambar I.19 *Sequence Diagram* Jumlah Kuota

4. *Sequence Diagram* Menggerakkan Perahu



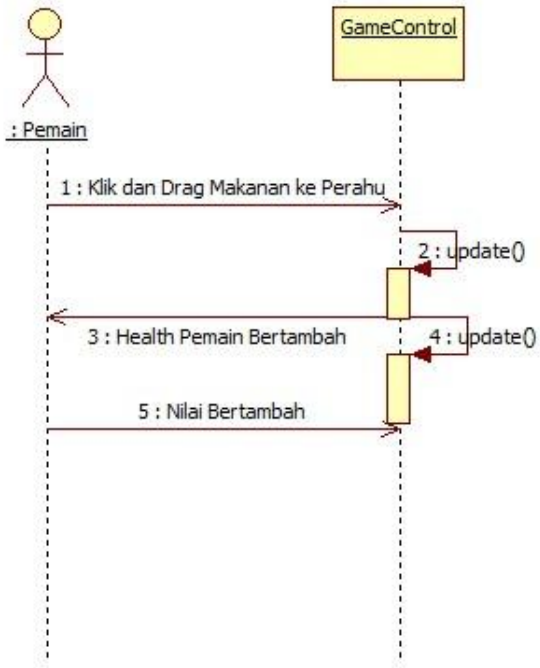
Gambar 3.20 *Sequence Diagram* Menggerakkan Perahu

5. *Sequence Diagram* Ambil Korban



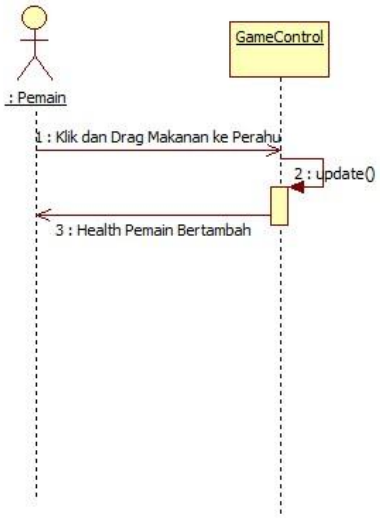
Gambar 3.21 *Sequence Diagram* Ambil Korban

6. *Sequence Diagram* Mengembalikan Korban ke Pos



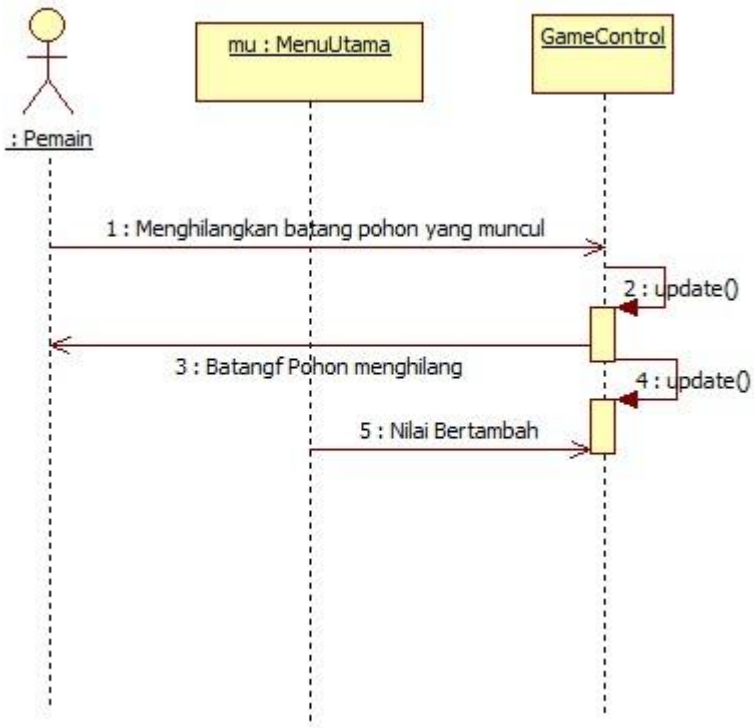
Gambar 3.22 *Sequence Diagram* Mengembalikan Korban ke Pos

7. *Sequence Diagram* Ambil Makanan



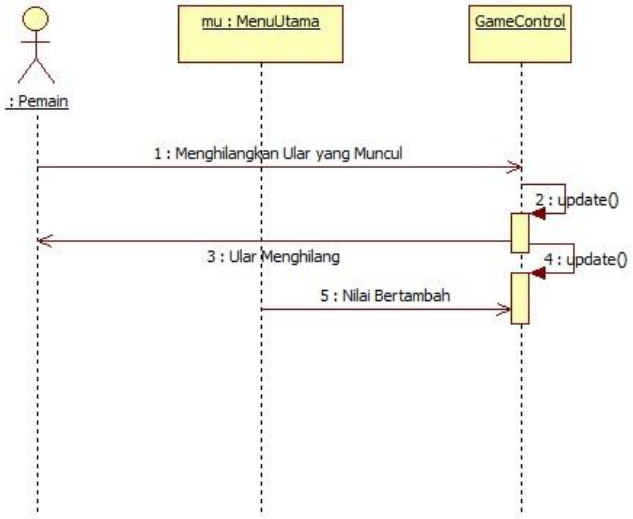
Gambar 3.23 *Sequence Diagram* Ambil Makanan

8. *Sequence Diagram* Menghilangkan Batang Pohon



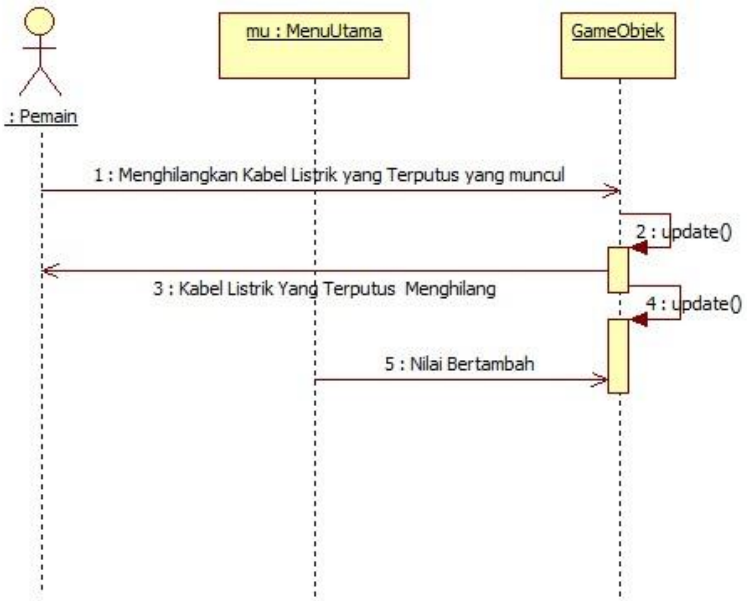
Gambar 3.24 *Sequence Diagram* Menghilangkan Batang Pohon

9. *Sequence Diagram* Menghilangkan Ular



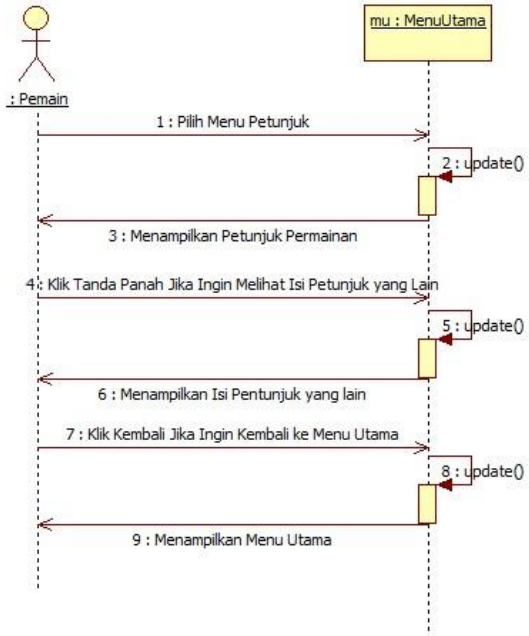
Gambar 3.25 *Sequence Diagram* Menghilangkan Ular

10. *Sequence Diagram* Menghilangkan Kabel Listrik yang Terputus



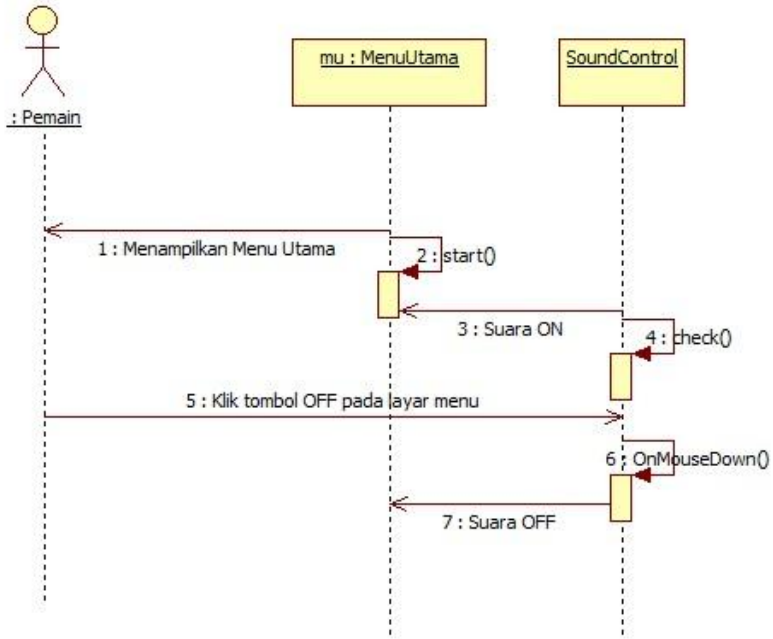
Gambar 3.26 *Sequence Diagram* Menghilangkan Kabel Listrik yang Terputus

11. *Sequence Diagram* Lihat Petunjuk



Gambar I.27 *Sequence Diagram* Lihat Petunjuk

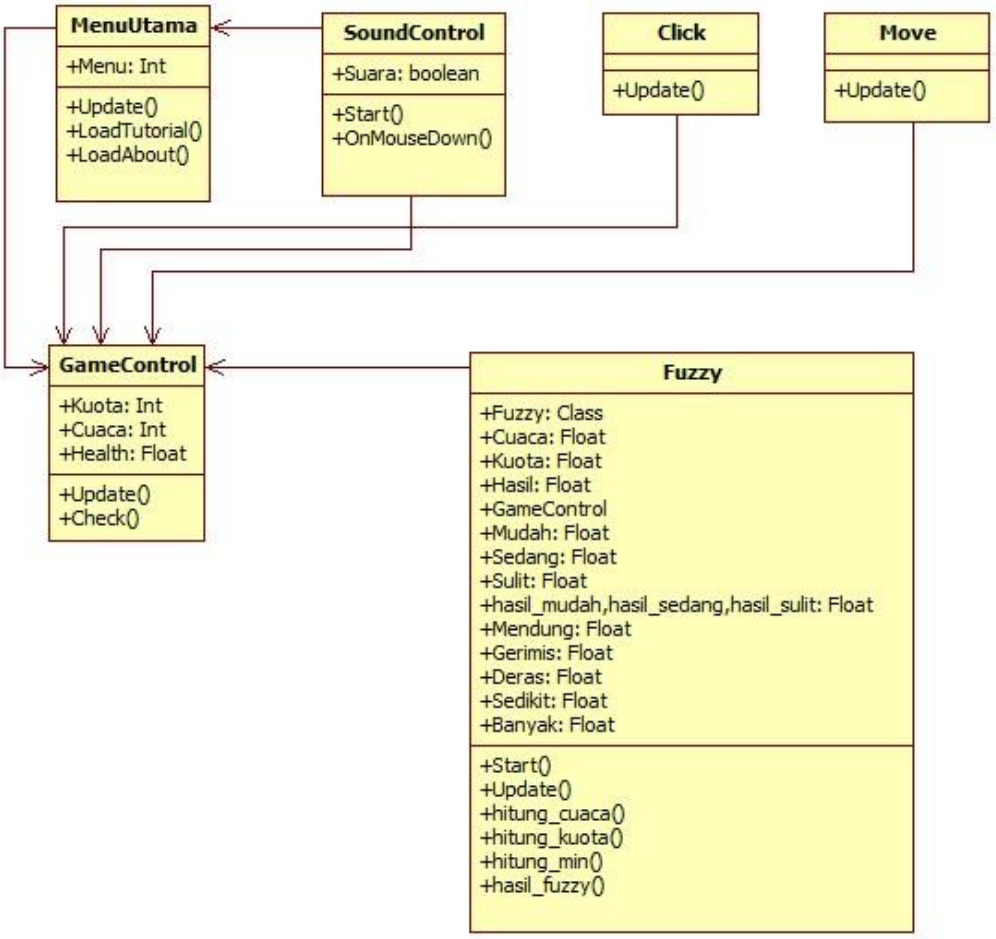
12. *Sequence Diagram* Pengaturan Suara



Gambar I.28 *Sequence Diagram* Pengaturan Suara

3.5.9 *Class Diagram*

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak dapat membuat kelas-kelas di dalam program perangkat lunak sesuai dengan perancangan diagram kelas. Berikut adalah *Class diagram* yang terdapat pada Gambar 3.29



Gambar 3.29 *Class Diagram*