BAB 3
ANALISIS ALGORITMA DAN PERANCANGAN

3.1 Analisis Masalah

Game simulasi bergenre *race* ini memiliki konsep mensimulasikan pengendara sepeda motor. Algoritma logika *fuzzy* merupakan solusi yang bisa diterapkan pada *game* simulasi bergenre *race* untuk mengatur tingkah laku serta menghasilkan aksi dan reaksi secara otomatis pada karakter NPC (*Non-Playable Character*) pada mobil rintangan berdasarkan kondisi yang dihadapi.

3.2 Analisis *Game*

Game simulasi bergenre *race*, misi utama dan *gameplay* dari *game* ini adalah mengendarai sepeda motor selamat sampai tujuan dan menghidari rintangan mobil di depannya.

Game simulasi bergenre *race* ini menerapkan algoritma logika *fuzzy* untuk mengatur tingkah laku serta menghasilkan aksi dan reaksi secara otomatis pada karakter NPC (*Non-Playable Character*) mobil rintangan yang ada. Tingkah laku pada NPC (*Non-Playable Character*) pada mobil rintangan yaitu sebagai berikut :

1. NPC bisa berpindah jalur dengan mendahului mobil rintangan di depannya.
2. NPC memilih tetap berada di jalurnya.

3.2.1 Analisis Leveling

Dalam *game* simulasi bergenre *race* ini terdapat 2 level yang masing – masing level dibedakan berdasarkan tingkat kesulitan rintangan yang harus dihadapi. Berikut ini penjelasan tingkat level kesulitan dapat di lihat pada Tabel 3.1.

Tabel 3.1 *Leveling Game*

Level	Deskripsi
Mudah	Tidak ada NPC yang menggunakan logika *fuzzy*
Sulit	Terdapat NPC yang menggunakan logika *fuzzy*

3.2.2 Analisis *Game* Sejenis

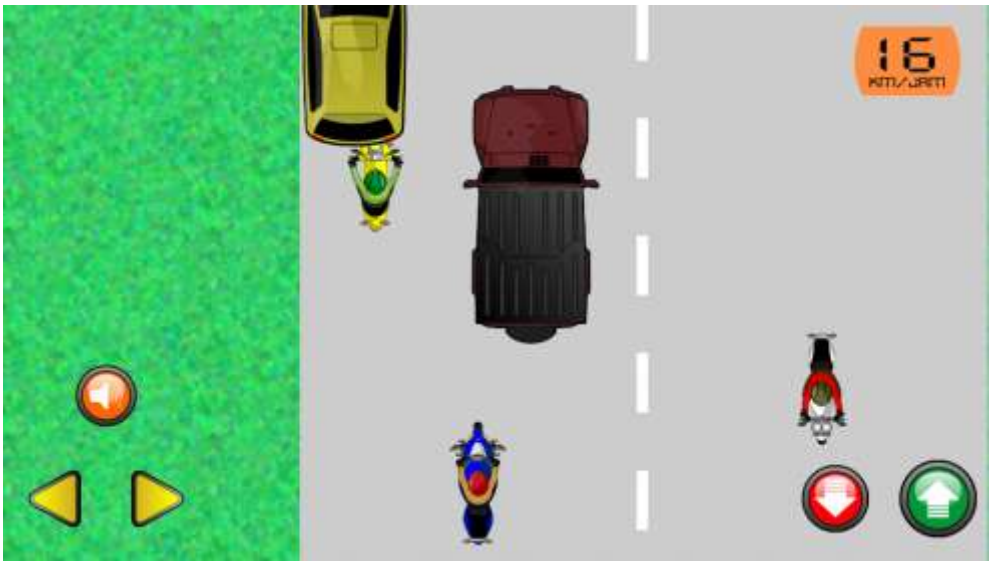
Sebelum merancang *game* yang akan dibuat, terlebih dahulu dilakukan analisis pada *game* sejenis yang sudah ada. Ini bertujuan agar *game* yang sudah dibuat nanti memiliki solusi dari permasalahan *game* yang sudah ada.

1. *Game* Simulasi Aman di Platform Android

   Game Simulasi Aman merupakan *game* simulasi sepeda motor bergenre *racing* berbasis *mobile*. *Game* Simulasi Aman ini merupakan *game* sederhana dengan mensimulasikan sepeda motor dan melewati rintangan yang ada. Tujuan dari *game* ini adalah mengendarai sepeda motor selamat sampai tujuan yaitu dari rumah ke kantor.

2. *Storyline*

   Game ini menceritakan tentang simulasi berkendara aman melakukan perjalanan dari rumah menuju kantor dengan melewati rintangan mobil dan motor.



Gambar 3.1 Tampilan *Game* Simulasi Aman

3.3 Analisis Masukan

Berikut ini adalah parameter atau variabel input dan output pada penelitian ini. Bisa dilihat pada tabel 3.2 dibawah ini.

Tabel 3.2 Variabel Input dan Output *Fuzzy*

Fungsi	Nama Variabel	Keterangan	Satuan	Semesta Pembicaraan
Input	Kecepatan	Variabel kecepatan di peroleh dari mobil rintangan yang menghalangi	pixel	[0-120]
	Jarak	Variabel jarak di peroleh dari ketika NPC mendekati mobil rintangan yang di depannya	pixel	[0-60]
	Jumlah Keadaan	Variabel keadaan di peroleh dari adanya mobil rintangan yang menghalangi atau tidak	-	[0-4]
Output	Keputusan	Mendahului mobil yang di depannya	-	[0-100]
		Tetap berada di jalur yang di laluinya	-	

Dari variabel yang telah dimunculkan, kemudian disusun domain himpunan *Fuzzy*. Berdasarkan domain tersebut, selanjutnya ditentukan fungsi keanggotaan dari masing – masing variabel seperti terlihat pada Tabel 3.3.

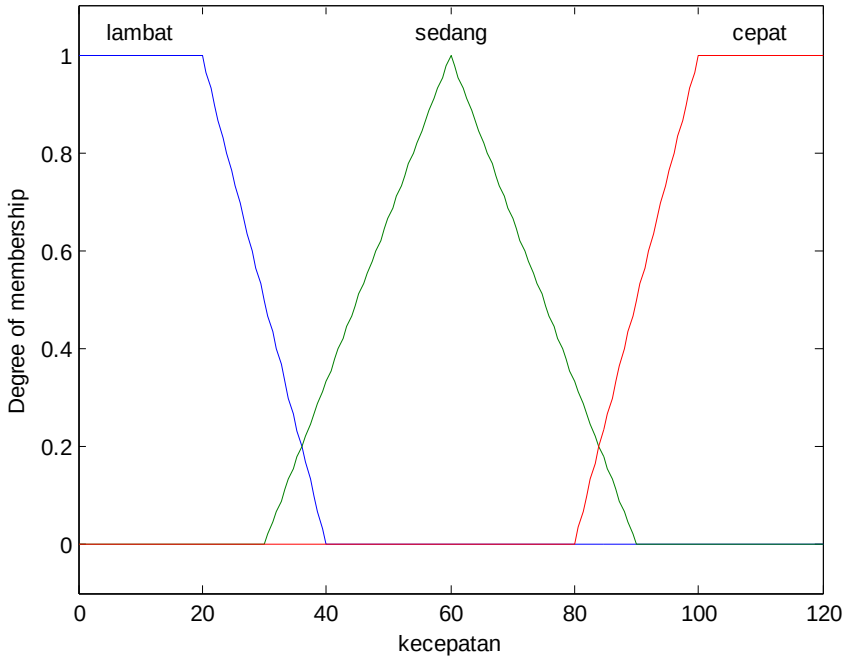
Tabel 3.3 Tabel Himpunan *Fuzzy*

Fungsi	Variabel	Himpunan	Domain	Fungsi Keanggotaan	Parameter
Input	Kecepatan	Lambat	[0-40]	Trapesium	(0;0;20;40)
		Sedang	[30-80]	Segitiga	(30;60;80)
		Cepat	[70-120]	Trapesium	(70;100;120;120)
	Jarak	Dekat	[0-20]	Trapesium	(0;0;5;20)
		Sedang	[10-40]	Segitiga	(10;25;40)
		Jauh	[30-60]	Trapesium	(30;55;60;60)
	Jumlah Keadaan	Sedikit	[0-1]	Segitiga	(0;0;1)
		Sedang	[1-3]	Segitiga	(1;2;3)
		Banyak	[2-4]	Segitiga	(3;4;4)
Output	Keputusan	Aman	[0-100]	Trapesium	(0;0;40;100)
		Berbahaya	[100-200]	Trapesium	(100;160;200;200)

Berdasarkan tabel himpunan *Fuzzy* diatas, maka pemetaan *input-output* pada penelitan ini adalah sebagai berikut :

1. Variabel Kecepatan

Variabel kecepatan merupakan variabel yang didapat dari laju kendaraan, Pada variabel kecepatan dibentuk tiga himpunan *fuzzy*, yaitu lambat, sedang dan cepat. Untuk merepresentasikan variabel kecepatan digunakam bentuk kurva trapesium untuk himpunan *fuzzy* lambat, kurva segitiga untuk himpunan *fuzzy* sedang dan kurva trapesium untuk himpunan *fuzzy* cepat. Untuk lebih jelasnya dapat dilihat pada gambar dibawah ini.



Gambar 3.2 Fungsi Keanggotaan Variabel Kecepatan

Dengan fungsi keanggotaan sebagai berikut :

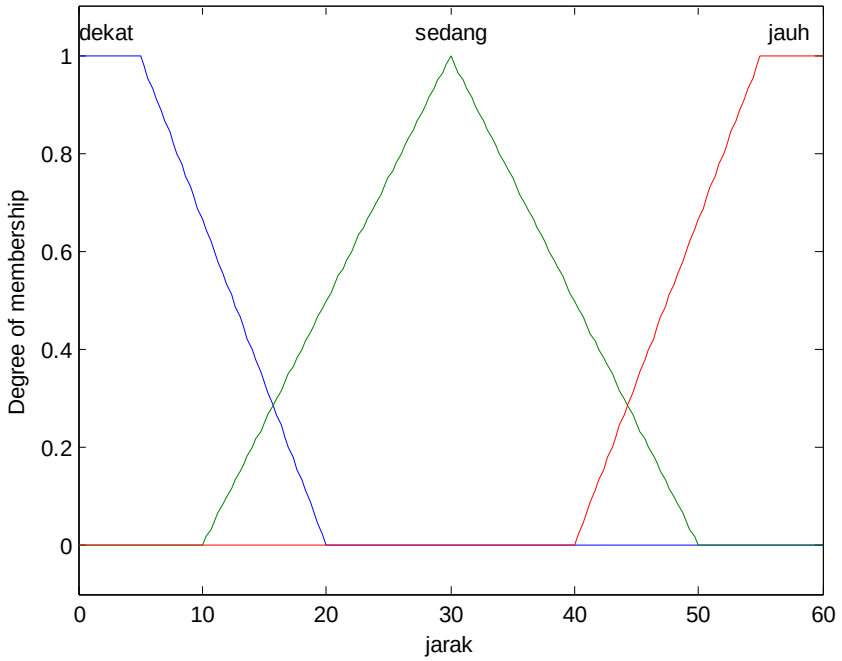
$$\mu Lambat(x) = \begin{cases} 1; & x \leq 20 \\ \frac{40-x}{40-20}; & 20 \leq x \leq 40 \\ 0; & x \geq 40 \end{cases}$$

$$\mu Sedang(x) = \begin{cases} 0; & x \leq 30 \; atau \; x \geq 90 \\ \frac{x-30}{60-30}; & 30 \leq x \leq 60 \\ \frac{60-x}{90-60}; & 60 \leq x \leq 90 \end{cases}$$

$$\mu Cepat(x) = \begin{cases} 0; & x \leq 80 \\ \dfrac{x-80}{100-80}; & 80 \leq x \leq 100 \\ 1; & x \geq 100 \end{cases}$$

2. Variabel Jarak

Variabel jarak diperoleh dari selisih jarak kendaraan *user* atau pemain dengan kendaraan NPC, pada variabel kecepatan dibentuk tiga himpunan *fuzzy*, yaitu dekat, sedang dan jauh. Untuk merepresentasikan variabel jarak digunakam bentuk kurva trapesium untuk himpunan *fuzzy* dekat, kurva segitiga untuk himpunan *fuzzy* sedang dan kurva trapesium untuk himpunan *fuzzy* jauh. Untuk lebih jelasnya dapat dilihat pada gambar dibawah ini.



Gambar 3.3 Fungsi Keanggotaan Variabel Jarak

Dengan fungsi keanggotaan sebagai berikut :

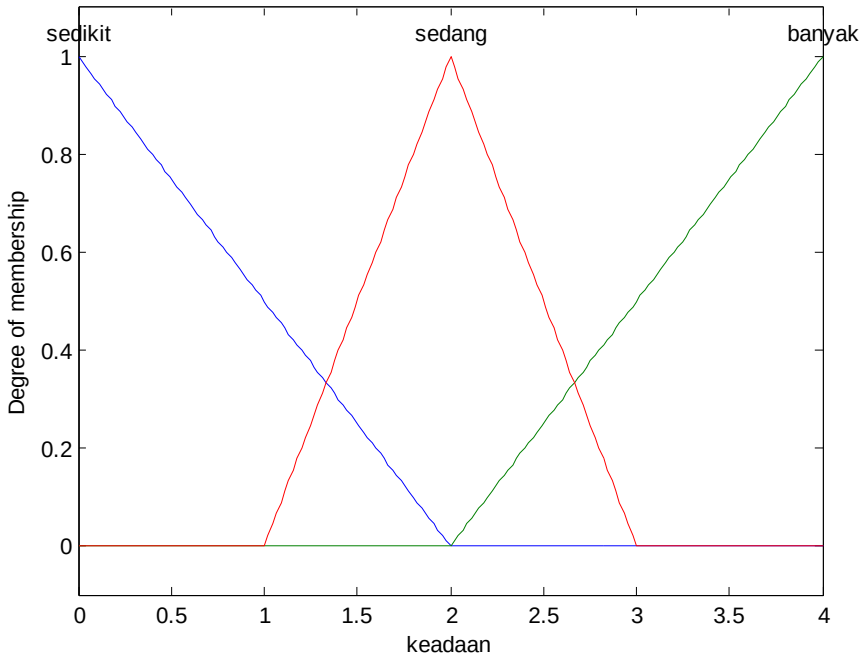
$$\mu Dekat(x) = \begin{cases} 1; & x \leq 5 \\ \dfrac{20-x}{20-5}; & 5 \leq x \leq 20 \\ 0; & x \geq 20 \end{cases}$$

$$\mu Sedang(x) = \begin{cases} 0; & x \leq 10 \; atau \; x \geq 50 \\ \dfrac{x-10}{30-10}; & 10 \leq x \leq 30 \\ \dfrac{30-x}{50-30}; & 30 \leq x \leq 50 \end{cases}$$

$$\mu Jauh(x) = \begin{cases} 0; & x \leq 40 \\ \dfrac{x-40}{55-40}; & 40 \leq x \leq 55 \\ 1; & x \geq 55 \end{cases}$$

3. Variabel Jumlah Keadaan

Variabel jumlah keadaan diperoleh dari banyaknya kendaraan yang berada di jalan, pada variabel keadaan dibentuk tiga himpunan *fuzzy*, yaitu sedikit, sedang dan banyak. Untuk merepresentasikan variabel keadaan digunakam bentuk kurva segitiga untuk himpunan *fuzzy* sedikit, kurva segitiga untuk himpunan *fuzzy* sedang dan kurva segitiga untuk himpunan *fuzzy* banyak. Untuk lebih jelasnya dapat dilihat pada gambar dibawah ini.



Gambar 3.4 Fungsi Keanggotaan Variabel Jumlah Keadaan

Dengan fungsi keanggotaan sebagai berikut :

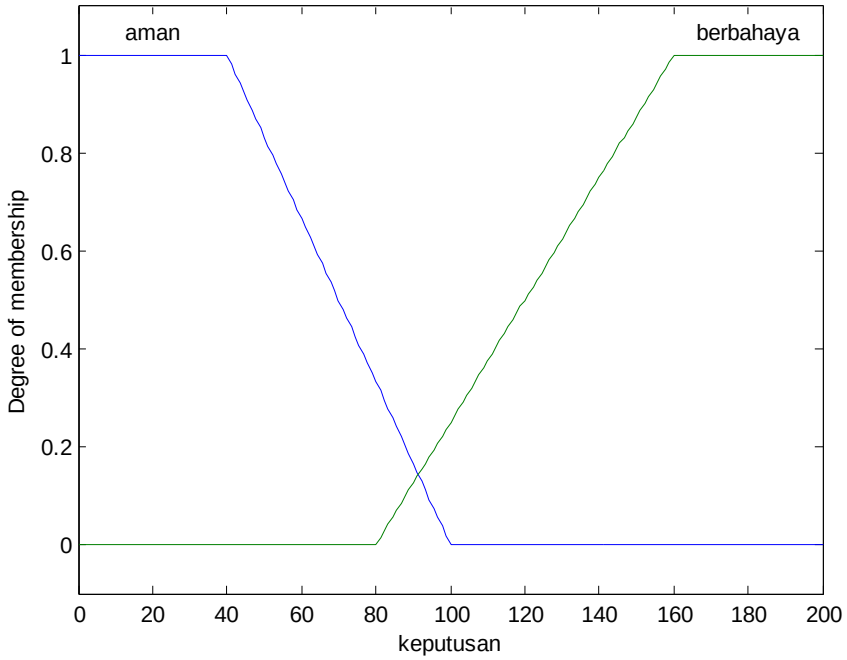
$$\mu Sedikit(x) = \begin{cases} \frac{2-x}{2-0}; & 0 \leq x \leq 2 \\ 0; & x \geq 2 \end{cases}$$

$$\mu Sedang(x) = \begin{cases} 0; & x \leq 1 \; atau \; x \geq 3 \\ \frac{x-1}{2-1}; & 1 \leq x \leq 2 \\ \frac{2-x}{3-2}; & 2 \leq x \leq 3 \end{cases}$$

$$\mu Banyak(x) = \begin{cases} 0; & x \leq 2 \\ \frac{x-2}{4-2}; & 2 \leq x \leq 4 \\ 1; & x \geq 4 \end{cases}$$

4. Variabel Keputusan

Variabel Keputusan, merupakan variabel *output*, pada variabel keputusan dibentuk dua himpunan *fuzzy*, yaitu aman dan berbahaya. Untuk merepresentasikan variabel keputusan digunakan bentuk kurva trapesium untuk himpunan *fuzzy* aman dan kurva trapesium untuk himpunan *fuzzy* berbahaya. Untuk lebih jelasnya dapat dilihat pada gambar dibawah ini.



Gambar 3.5 Fungsi Keanggotaan Variabel Keputusan

Dengan fungsi keanggotaan sebagai berikut :

$$\mu Aman(z) = \begin{cases} 1; & z \leq 40 \\ \frac{100-z}{100-40}; & 40 \leq z \leq 100 \\ 0; & z \geq 100 \end{cases}$$

$$\mu Berbahaya(z) = \begin{cases} 0; & z \leq 80 \\ \frac{z-80}{160-80}; & 80 \leq z \leq 160 \\ 1; & z \geq 160 \end{cases}$$

3.4 Analisis Algoritma *Fuzzy*

Pada penelitian ini logika *Fuzzy* digunakan untuk menentukan keputusan NPC untuk mendahului mobil di depannya dengan berpindah jalur atau tetap berada di jalur yang di laluinya. Berdasarkan analisis masukan variabel yang digunakan pada penelitian ini berupa kecepatan, jarak dan keadaan sebagai input, dan keputusan sebagai ouput. Berikut ini adalah contoh penerapan logika *fuzzy* dalam menentukan keputusan pada NPC mobil rintangan.

3.4.1 Contoh Penerapan

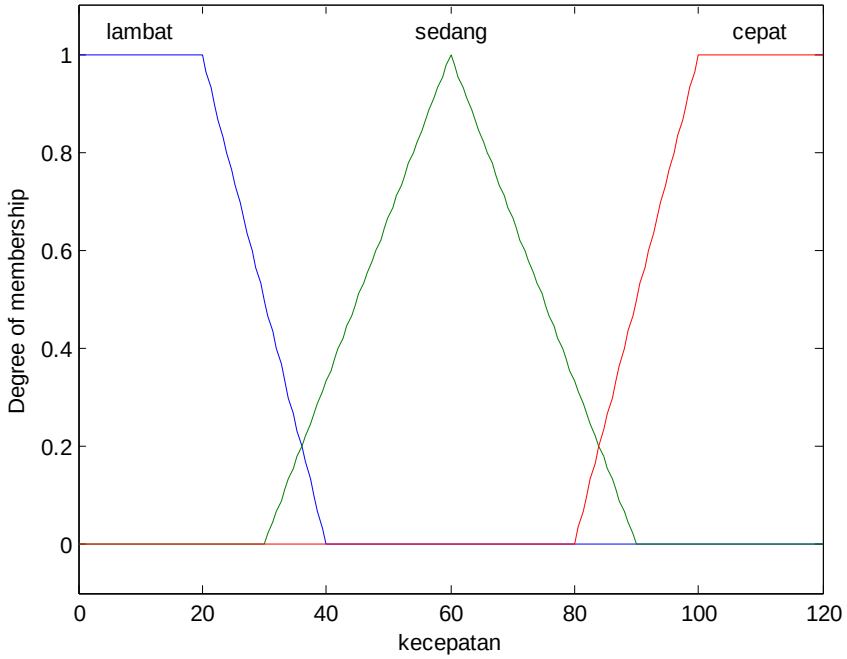
Jika kecepatan mobil rintangan yang di depannya adalah 35, jarak 15 dan keadaannya 1 maka perilaku yang akan di tunjukan oleh NPC mobil rintangan akan di jelaskan sebagai berikut :

3.4.1.1 Menentukan Himpunan *Fuzzy*

Himpunan *fuzzy* beserta fungsi keanggotaan dari variabel kecepatan, jarak dan keadaan direpresentasikan sebagai berikut :

1. Variabel *Fuzzy* Kecepatan

   Perhitungan variable kecepatan dari setiap himpunan *fuzzy* (lambat, sedang, cepat) dimana nilai kecepatan 35 adalah sebagai berikut :



Gambar 3.6 Fungsi Keanggotaan untuk Kecepatan 35

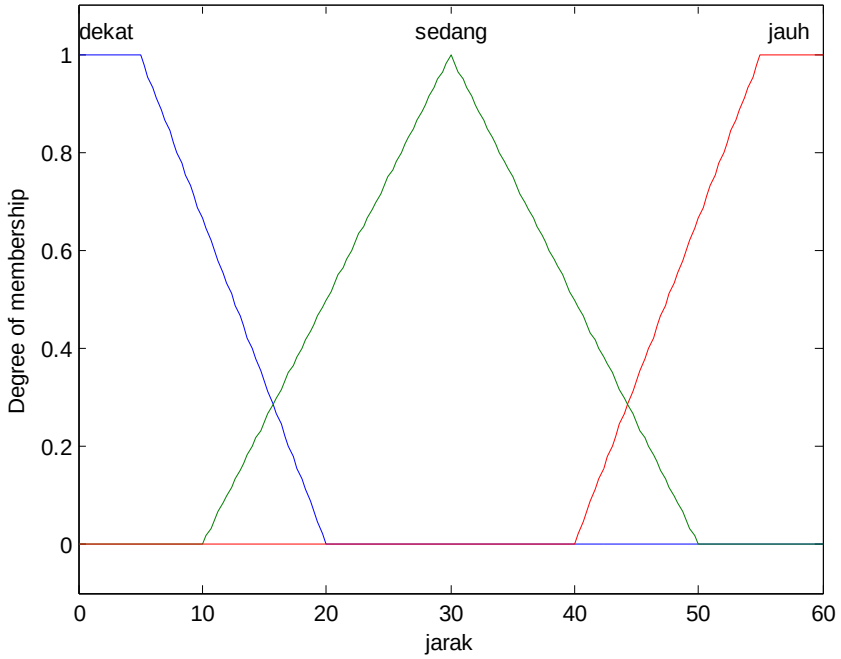
Kecepatan sebesar 35 termasuk ke dalam himpunan *Fuzzy* LAMBAT dan SEDANG, dengan tingkat keanggotaan sesuai fungsi berikut :

$$\mu Lambat[35] = \frac{40 - 35}{40 - 20} = 0.25$$

$$\mu Sedang[35] = \frac{35 - 30}{60 - 30} = 0.16$$

2. Variabel *Fuzzy* Jarak

   Perhitungan variable jarak dari setiap himpunan *fuzzy* (dekat, sedang, jauh) dimana nilai jarak 15 adalah sebagai berikut :



Gambar 3.7 Fungsi Keanggotaan untuk Jarak 15

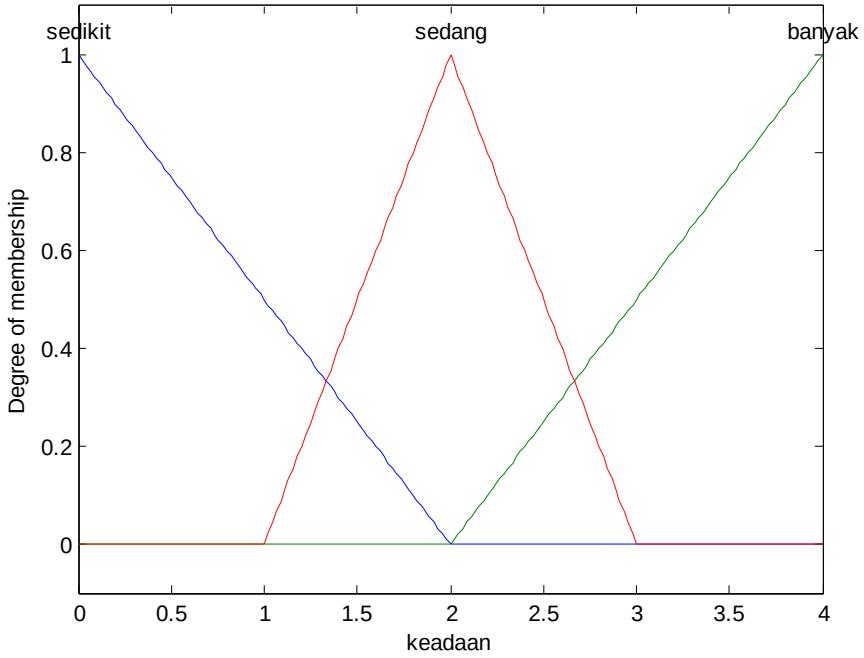
Jarak sebesar 15 termasuk ke dalam himpunan *Fuzzy* DEKAT dan SEDANG, dengan tingkat keanggotaan sesuai fungsi berikut :

$$\mu Dekat[15] = \frac{20 - 15}{20 - 5} = 0.33$$

$$\mu Sedang[15] = \frac{15 - 10}{30 - 10} = 0.25$$

3. Variabel *Fuzzy* Jumlah Keadaan

   Perhitungan variable keadaan dari setiap himpunan *fuzzy* (sedikit, sedang, banyak) dimana nilai keadaan 3 adalah sebagai berikut :



Gambar 3.8 Fungsi Keanggotaan untuk Jumlah Keadaan 3

Keadaan sebesar 1 termasuk ke dalam himpunan *Fuzzy* Sedang dan Banyak, dengan tingkat keanggotaan sesuai fungsi berikut :

$$\mu Sedikit[1] = \frac{2-1}{2-0} = 0.5$$

$$\mu Sedang[1] = \frac{1-1}{2-1} = 0$$

3.4.1.2 Fungsi Implikasi

Berikut ini adalah pembentukan aturan pada NPC Mobil :

```
[1] IF Kecepatan cepat AND Jarak jauh AND Jumlah Keadaan
    sedikit THEN Aman
[2] IF Kecepatan cepat AND Jarak jauh AND Jumlah Keadaan
    sedang THEN Aman
[3] IF Kecepatan cepat AND Jarak jauh AND Jumlah Keadaan
    banyak THEN Aman
```

[4] IF Kecepatan cepat AND Jarak sedang AND Jumlah Keadaan sedikit THEN Aman

[5] IF Kecepatan cepat AND Jarak sedang AND Jumlah Keadaan sedang THEN Aman

[6] IF Kecepatan cepat AND Jarak sedang AND Jumlah Keadaan banyak THEN Aman

[7] IF Kecepatan cepat AND Jarak dekat AND Jumlah Keadaan sedikit THEN Aman

[8] IF Kecepatan cepat AND Jarak dekat AND Jumlah Keadaan sedang THEN Aman

[9] IF Kecepatan cepat AND Jarak dekat AND Jumlah Keadaan banyak THEN Berbahaya

[10] IF Kecepatan sedang AND Jarak jauh AND Jumlah Keadaan sedikit THEN Aman

[11] IF Kecepatan sedang AND Jarak jauh AND Jumlah Keadaan sedang THEN Aman

[12] IF Kecepatan sedang AND Jarak jauh AND Jumlah Keadaan banyak THEN Aman

[13] IF Kecepatan sedang AND Jarak sedang AND Jumlah Keadaan sedikit THEN Aman

[14] IF Kecepatan sedang AND Jarak sedang AND Jumlah Keadaan sedang THEN Aman

[15] IF Kecepatan sedang AND Jarak sedang AND Jumlah Keadaan banyak THEN Berbahaya

[16] IF Kecepatan sedang AND Jarak dekat AND Jumlah Keadaan sedikit THEN Aman

[17] IF Kecepatan sedang AND Jarak dekat AND Jumlah Keadaan sedang THEN Berbahaya

[18] IF Kecepatan sedang AND Jarak dekat AND Jumlah Keadaan banyak THEN Berbahaya

[19] IF Kecepatan lambat AND Jarak jauh AND Jumlah Keadaan Sedikit THEN Aman

[20] IF Kecepatan lambat AND Jarak jauh AND Jumlah Keadaan sedang THEN Aman

[21]IF Kecepatan lambat AND Jarak jauh AND Jumlah Keadaan banyak THEN Berbahaya

[22]IF Kecepatan lambat AND Jarak sedang AND Jumlah Keadaan sedikit THEN Aman

[23]IF Kecepatan lambat AND Jarak sedang AND Jumlah Keadaan sedang THEN Berbahaya

[24]IF Kecepatan lambat AND Jarak sedang AND Jumlah Keadaan banyak THEN Berbahaya

[25]IF Kecepatan lambat AND Jarak dekat AND Jumlah Keadaan sedikit THEN Berbahaya

[26]IF Kecepatan lambat AND Jarak dekat AND Jumlah Keadaan sedang THEN Berbahaya

[27]IF Kecepatan lambat AND Jarak dekat AND Jumlah Keadaan banyak THEN Berbahaya

Setelah aturan dibentuk, maka dilakukan aplikasi fungsi implikasi Fungsi implikasi yang digunakan dalam proses ini adalah fungsi MIN, yaitu dengan mengambil tingkat keanggotaan yang minimum dari variabel input sebagai outputnya. Berdasarkan aturan-aturan yang sesuai dengan kondisi tersebut.

[13]IF Kecepatan sedang AND Jarak sedang AND Jumlah Keadaan sedikit THEN Aman

$$\alpha - Predikat_1 \begin{array}{l} = \mu_{kcptnSedang \cap jarakSedang \cap KeadaanSedikit} \\ = \min\left(\mu_{kcptnSedang(35)};\ \mu_{jarakSedang(15)};\ \mu_{keadaanSedikit(1)}\right) \\ = \min\left(0.16; 0.25; 0.5\right) \\ = 0.16 \end{array}$$

[14]IF Kecepatan sedang AND Jarak sedang AND Jumlah Keadaan sedang THEN Aman

$$\alpha - Predikat_2 \begin{array}{l} = \mu_{kcptnSedang \cap jarakSedang \cap KeadaanSedang} \\ = \min\left(\mu_{kcptnSedang(35)};\ \mu_{jarakSedang(15)};\ \mu_{keadaanSedang(1)}\right) \\ = \min\left(0.16; 0.25; 0\right) \\ = 0 \end{array}$$

[16]IF Kecepatan sedang AND Jarak dekat AND Jumlah Keadaan sedikit THEN Aman

$$\alpha - Predikat_3 = \mu_{kcptnSedang \cap jarakDekat \cap KeadaanSedikit} = \min(\mu_{kcptnSedang}(35);\ \mu_{jarakDekat}(15);\ \mu_{keadaanSedikit}(1)) = \min(0.16; 0.33; 0.5) = 0.16$$

[17]IF Kecepatan sedang AND Jarak dekat AND Jumlah Keadaan sedang THEN Berbahaya

$$\alpha - Predikat_4 = \mu_{kcptnSedang \cap jarakDekat \cap KeadaanSedang} = \min(\mu_{kcptnSedang}(35);\ \mu_{jarakDekat}(15);\ \mu_{keadaanSedang}(1)) = \min(0.16; 0.33; 0) = 0$$

[22]IF Kecepatan lambat AND Jarak sedang AND Jumlah Keadaan sedikit THEN Aman

$$\alpha - Predikat_5 = \mu_{kcptnLambat \cap jarakSedang \cap KeadaanSedikit} = \min(\mu_{kcptnLambat}(35);\ \mu_{jarakSedang}(15);\ \mu_{keadaanSedikit}(1)) = \min(0.25; 0.25; 0.5) = 0.25$$

[23]IF Kecepatan lambat AND Jarak sedang AND Jumlah Keadaan sedang THEN Berbahaya

$$\alpha - Predikat_6 = \mu_{kcptnLambat \cap jarakSedang \cap KeadaanSedang} = \min(\mu_{kcptnLambat}(35);\ \mu_{jarakSedang}(15);\ \mu_{keadaanSedang}(1)) = \min(0.25; 0.25; 0) = 0$$

[25]IF Kecepatan lambat AND Jarak dekat AND Jumlah Keadaan sedikit THEN Berbahaya

$$\alpha - Predikat_7 = \mu_{kcptnLambat \cap jarakDekat \cap KeadaanSedikit} = \min(\mu_{kcptnLambat}(35);\ \mu_{jarakDekat}(15);\ \mu_{keadaanSedikit}(1)) = \min(0.25; 0.33; 0.5) = 0.25$$

[26]IF Kecepatan lambat AND Jarak dekat AND Jumlah Keadaan sedang THEN Berbahaya

$$\alpha - Predikat_8 = \mu_{kcptnLambat \cap jarakDekat \cap KeadaanSedang} = \min(\mu_{kcptnLambat}(35);\ \mu_{jarakDekat}(15);\ \mu_{keadaanSedang}(1)) = \min(0.25; 0.33; 0) = 0$$

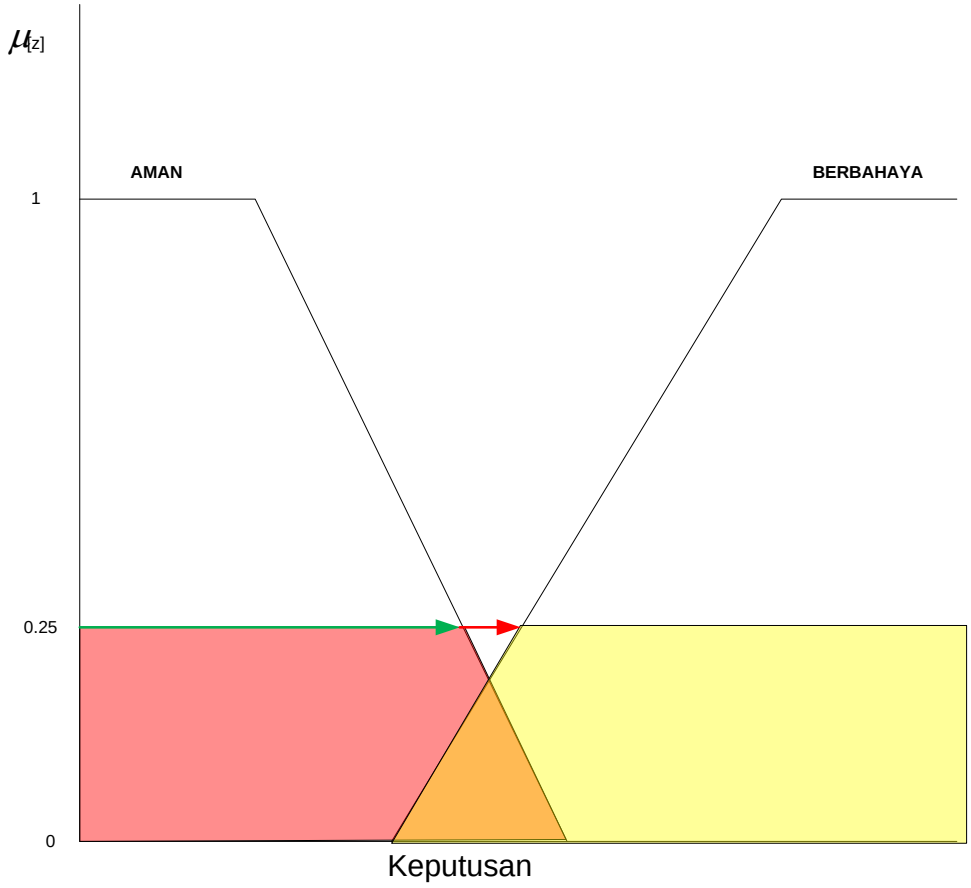
3.4.1.3 Komposisi Aturan

Dari hasil aplikasi fungsi implikasi dari tiap aturan, digunakan metode MAX untuk melakukan komposisi antar semua aturan.

$$\mu_{sf}(z) = \mu_{aman(z)} \cup \mu_{berbahaya(z)}$$

$$= maks\{0.25, 0.25\}$$

Untuk lebih jelasnya bisa dilihat pada gambar dibawah ini.



Gambar 3.9 Daerah Hasil Komposisi Aturan Metode Max

Dimana fungsi keanggotaan variabel keputusan adalah sebagai berikut :

$$\mu Aman(z) = \begin{cases} 1; & z \leq 40 \\ \frac{100-z}{100-40}; & 40 \leq z \leq 100 \\ 0; & z \geq 100 \end{cases}$$

$$\mu Berbahaya(z) = \begin{cases} 0; & z \leq 80 \\ \frac{z-80}{160-80}; & 80 \leq z \leq 160 \\ 1; & z \geq 160 \end{cases}$$

Sehingga ketika nilai himpunan aman = 0.25 didapat fungsi keanggotaan seperti berikut ini :

$\mu Aman(z) = \frac{100-z}{100-40} = \mathbf{0.25}$ maka didapat hasil z adalah 85

Ketika nilai himpunan Berbahaya = 0.25 didapat fungsi keanggotaan seperti berikut ini :

$\mu Berbahaya(z) = \frac{z-80}{160-80} = 0.25$ maka didapat nilai z adalah 100

Berikut ini fungsi keanggotaan dari hasil komposisi aturan.

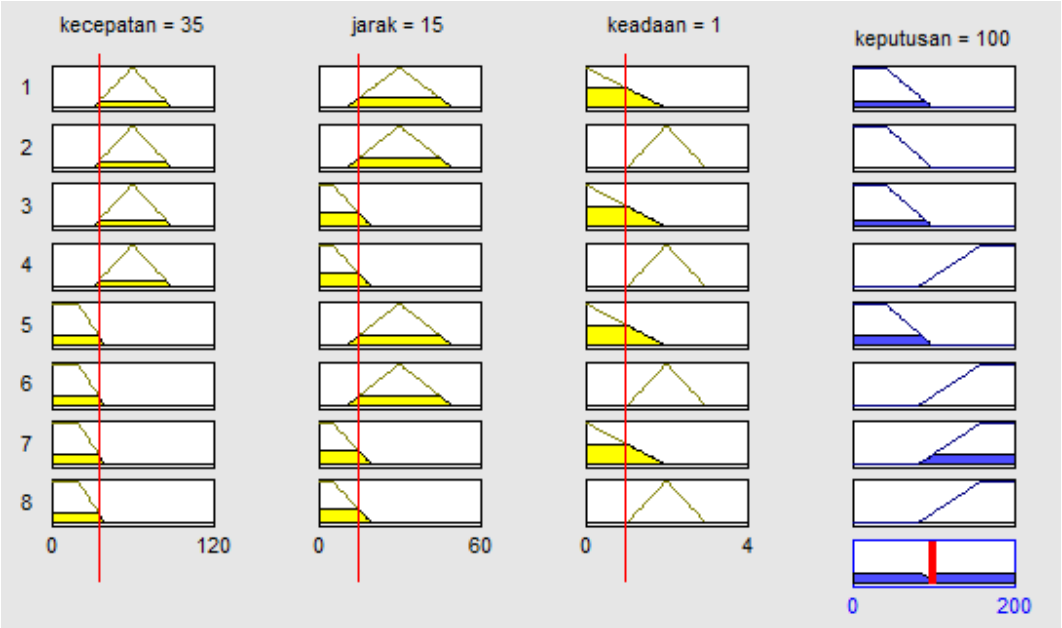
$$\mu Keputusan(z) = \begin{cases} 0.25; & 0 \leq z \leq 85 \\ \frac{z-80}{160-80}; & 85 \leq z \leq 100 \\ 0.25 & 100 \leq z \leq 200 \end{cases}$$

3.4.1.4 Penegasan (*Defuzzy*)

Berikut ini adalah perhitungan deffuzifikasi menggunakan metode *centroid.*

$$Z = \frac{\int_0^{85} 0.25x\,dx + \int_{85}^{100} \frac{x-80}{160-80}x\,dx + \int_{100}^{200} 0.25x\,dx}{\int_0^{85} 0.25\,dx + \int_{85}^{100} \frac{x-80}{160-80}\,dx + \int_{100}^{200} 0.25\,dx} = \frac{4873.4375}{48.59375} = 100.28$$

Nilai keputusan 100.28 termasuk kedalam kategori berbahaya. Berikut ini hasil perhitungan berdasarkan *fuzzy logic* toolbox pada matlab.



Gambar 3.10 Perhitungan Menggunakan *Fuzzy Logic Toolbox Matlab*

3.5 Analisis Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan perangkat lunak dilakukan berdasarkan kebutuhan game. Spesifikasi kebutuhan perangkat lunak akan dibagi ke dalam dua bagian yaitu SKPL-F (Spesifikasi Kebutuhan Perangkat Lunak Fungsional) dan SKPL-NF (Spesifikasi Kebutuhan Perangkat Lunak Non-Fungsional). Spesifikasi kebutuhan perangkat lunak dalam penelitian ini dapat dilihat pada tabel berikut.

Tabel 3.4 SKPL *User Requirement*

Kode	Kebutuhan
SKPL–F1	Perangkat Lunak dapat memasukan parameter yang diinginkan oleh user.
SKPL-F2	Perangkat Lunak dapat memproses hasil masukan.
SKPL-F3	Perangkat Lunak dapat melakukan proses fuzzyfikasi untuk mencari nilai derajat keanggotaan.
SKPL-F4	Perangkat Lunak dapat menampilkan hasil dari proses Fuzzyfikasi.

Tabel 3.5 SKPL System *Requirement*

Kode	Kebutuhan
SKPL–F1	1. NPC berpindah jalur dan mendahului mobil rintangan yang di depannya. 2. NPC tetap berada di jalur aman
SKPL-F2	1. Proses perhitungan masukan yang diterima system yang kemudian diolah menjadi keluaran sesuai keinginan *user*.

Tabel 3.6 SKPL Non-Fungsional

Kode	Kategori	Kebutuhan
SKPL-NF1	*Efficiency (Product Requirement)*	Sistem dapat melakukan *fuzzifikasi*
SKPL-NF2	*Operational (Organizatonal Requirement)*	Setiap kesalahan (*error*) yang terjadi disertai dengan pesan *error* kepada pengguna
SKPL-NF3	*Development (Organizatonal Requirement)*	Untuk menjalankan sistem yang di bangun maka dibutuhkan : a. *Microsoft Windows* sebagai sistem operasi b. Greenfoot *tools* pemograman

3.5.1 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non fungsional adalah langkah dimana seorang pembangun perangkat lunak menganalisis sumber daya yang akan digunakan dan menggunakan perangkat lunak yang dibangun. Perangkat keras dan perangkat lunak yang dimiliki harus sesuai dengan kebutuhan, sehingga dapat ditentukan kompabilitas aplikasi yang dibangun terhadap sumber daya yang ada.

3.5.1.1 Analisis Kebutuhan Perangkat Keras

Perangkat keras yang dibutuhkan untuk penerapanlogika *fuzzy* pada *game*yang akan dibuat. seperti yang terlihat pada Tabel 3.7.

Tabel 3.7 Spesifikasi Perangkat Keras Yang Dibutuhkan

No	Perangkat Keras	Spesifikasi
1	Prosesor	Kecepatan 2.4 GHz
2	Monitor	Monitor 14'', resolusi 1024x768
3	VGA On-board	Kecepatan 512 MB
4	Optical Drive DVD-ROM	Standard
5	Keyboard	Standard
6	Mouse	Standard
7	Memori	2GB

3.5.1.2 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan untuk penerapanlogika *fuzzy* pada *game*yang akan dibuat. Adapun perangkat lunak yang dibutuhkan dapat dilihat pada Tabel 3.8.

Tabel 3.8 Analisis Kebutuhan Perangkat Lunak

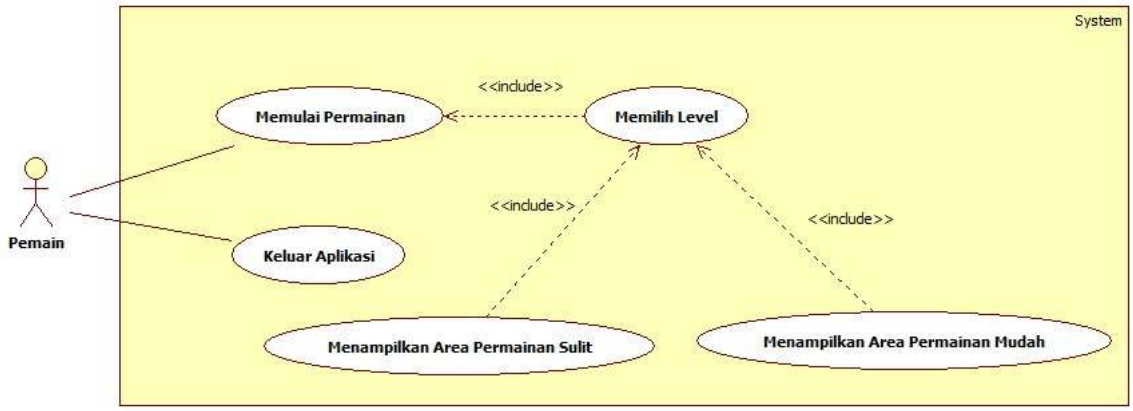
No	Perangkat Lunak	Keterangan
1	Sistem Operasi	Windows 7
2	*Tool*s Program	Greenfoot

3.5.2 Analisis Kebutuhan Fungsional

Spesifikasi kebutuhan fungsional adalah spesifikasi tentang hal-hal yang akan dilakukan sistem ketika diimplementasikan. Analisis kebutuhan ini diperlukan untuk menentukan keluaran yang akan dihasilkan sistem, masukan yang diperlukan sistem, lingkup proses yang digunakan untuk mengolah masukan menjadi keluaran.

3.5.2.1 Use Case Diagram

Use case diagram merupakan gambaran umum dari sebuah sistem sehingga pengguna mengerti mengenai sistem yang akan dibangun. *Use case* dapat dilihat pada gambar 3.11.



Gambar 3.11 *Use Case* Diagram

Definisi *use case* berfungsi untuk menjelaskan fungsi dari *use case* yang terdapat pada diagram *use case*. Definisi *use case* dijelaskan pada tabel 3.9.

Tabel 3.9 Definisi *Use Case*

No	*Use Case*	Keterangan
1	Mulai Permainan	Proses untuk memulai permainan
2	Melilih *Level*	Proses untuk memilih *level* mudah dan sulit

3	Menampilkan Area Permainan Sulit	Proses untuk menampilkan area permainan sulit
4	Menampilkan Area Permainan Mudah	Proses untuk menampilkan area permainan mudah
5	Keluar	Proses untuk keluar dari permainan

3.5.2.2 Skenario *Use Case*

Skenario *use case* merupakan penjelasan mengenai alur kejadian yang di dalamnya menjelaskan urutan interaksi antara aktor dengan *use case* tersebut. Adapun skenario *use case* digambarkan sebagai berikut.

Tabel 3.10 Skenario *Use Case* Memulai Permainan

Identifikasi	
Nomor	1
Nama	Memulai Permainan
Tujuan	Memulai permainan
Deskripsi	Proses untuk memulai permainan
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pemain berada di menu utama permainan
Aksi Aktor	Reaksi Sistem
1. Memilihmulai	
	2. Menampilkan menu pilihan level

Tabel 3.11 Skenario *Use Case* Memilih Level

Identifikasi	
Nomor	2
Nama	Permainan
Tujuan	Menampilkan pilihan level
Deskripsi	Proses untuk memilihlevel
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pemain berada di tampilan pilih level
Aksi Aktor	Reaksi Sistem
	1. Menampilkan pilihan level
2. Jika memilih level mudah	
	3. Menampilkan area permainan
4. Jika memilih level sulit	
	5. Menampilkan area permainan
6. Jika memilih kembali	
	7. Menampilkan menu utama

Tabel 3.12 Skenario *Use Case* Menampilkan Arena Permainan Sulit

Identifikasi	
Nomor	3
Nama	Menampilkan Area Permainan Sulit
Tujuan	Menampilkan area permainan sulit
Deskripsi	Proses untuk menampilkan area permainan sulit

Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pemain berada di area permainan sulit
Aksi Aktor	Reaksi Sistem
	1. Menampilkan area permainan sulit
	2. Menampilkan karakter
	3. Menampilkan NPC
	4. Menampilkan fuzzy pada NPC
	5. Cek kecepatan
	6. Cek jarak
	7. Cek keadaan
	8. Defuzzy
	9. Menampilkan pergerakan NPC
10. Menekan tombol kanan/kiri	
	11. Menampilkan pergerakan karakter
	12. Cek tabrakan
	13. Cek jarak tempuh

Tabel 3.13 Skenario *Use Case* Menampilkan Arena Permainan Mudah

Identifikasi	
Nomor	4
Nama	Menampilkan Area Permainan Mudah
Tujuan	Menampilkan area permainan mudah
Deskripsi	Proses untuk menampilkan area permainan mudah
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pemain berada di area permainan mudah
Aksi Aktor	Reaksi Sistem
	1. Menampilkan area permainan mudah
	2. Menampilkan karakter
	3. Menampilkan NPC
	4. Menampilkan pergerakan NPC
5. Menekan tombol kanan/kiri	
	6. Menampilkan pergerakan karakter
	7. Cek tabrakan
	8. Cek jarak tempuh

Tabel 3.14 Skenario *Use Case* Keluar Permainan

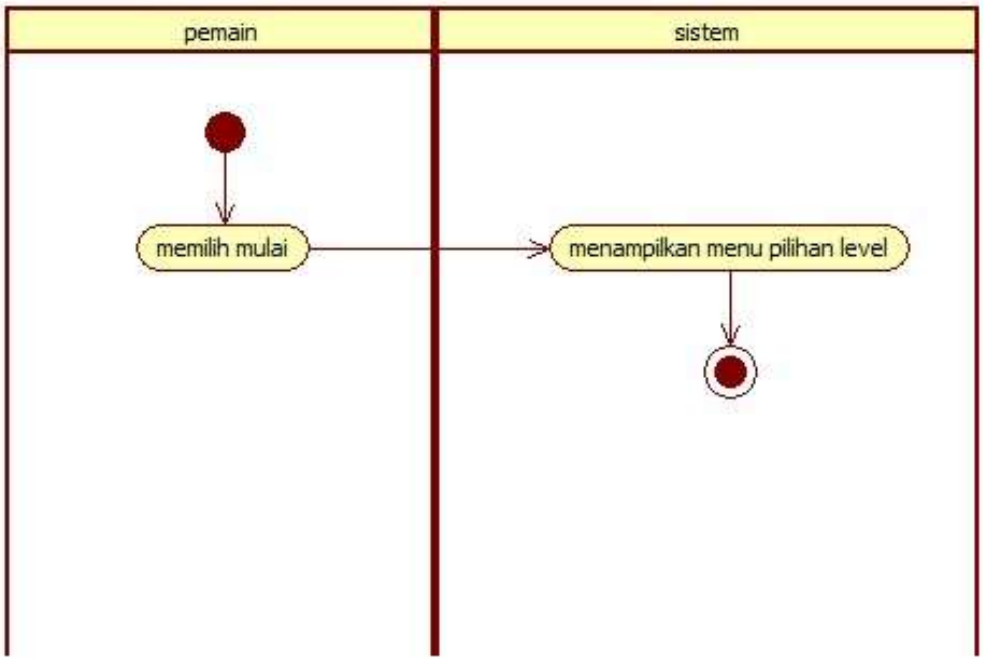
Identifikasi	
Nomor	5
Nama	Keluar Permainan
Tujuan	Keluar dariaplikasi permainan
Deskripsi	Proses untuk keluar dari permainan
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pemain berada di menu utama permainan
Aksi Aktor	Reaksi Sistem
1. Memilih keluar	
	2. Menutup aplikasi permainan

3.5.2.3 *Activity* Diagram

Activity diagram merupakan gambaran alur aktivitas pada permainan yang sedang dirancang, bagaimana masing – masing alur aktivitas berawal, *decision* yang mungkin terjadi, dan bagaimana berakhir. *Activity* diagram dijelaskan melalui gambar berikut

1. *Activity Diagram* Memulai Permainan

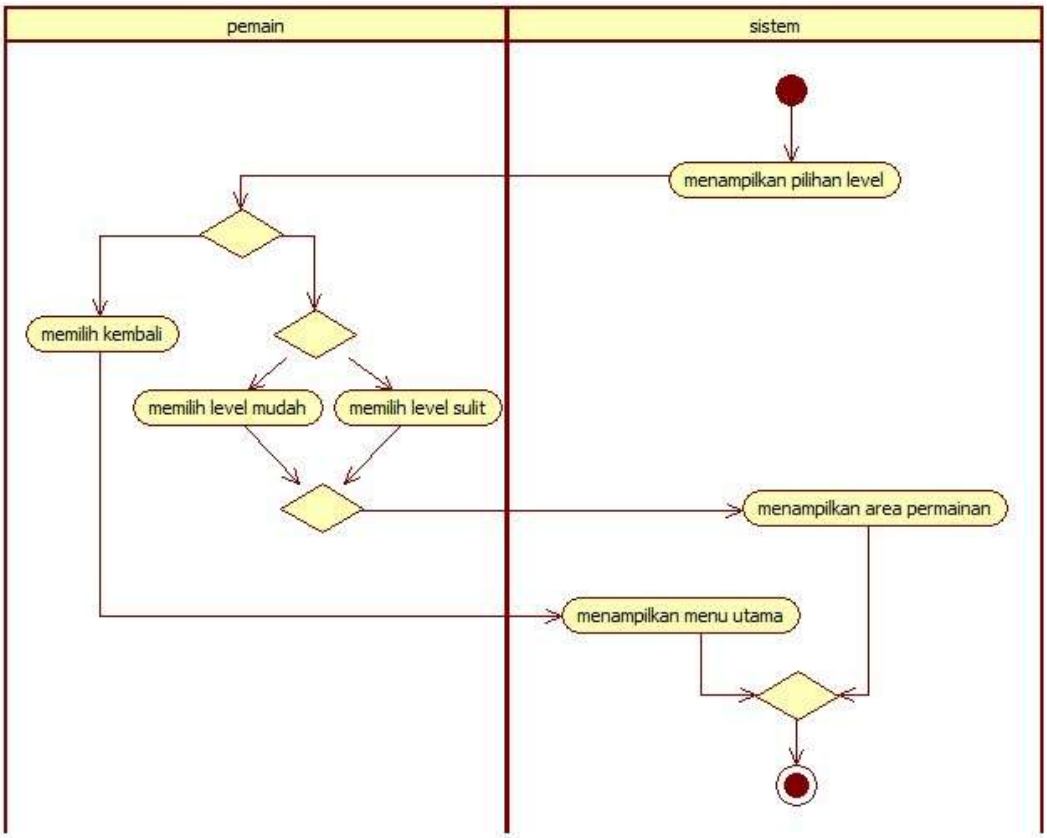
   Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas memulai *game*. *Activity diagram* memulai *game* dapat dilihat pada gambar 3.12.



Gambar 3.12 *Activity Diagram* Memulai Permainan

2. *Activity Diagram* Memilih *Level*

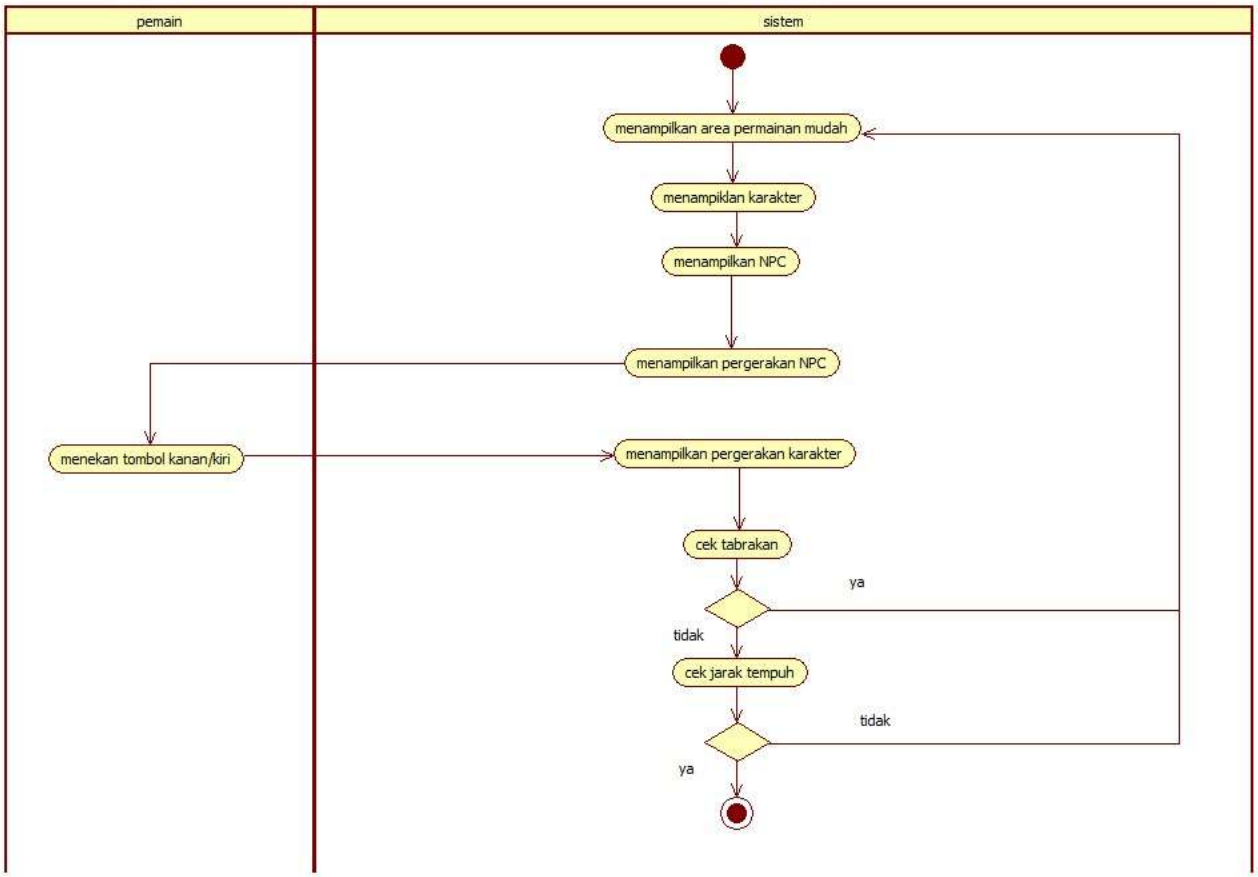
   Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas permainan.*Activity diagram* memilih level dapat dilihat pada gambar 3.13.



Gambar 3.13 ***Activity Diagram*** **Memilih** ***Level***

3. *Activity Diagram* Menampilkan Area Permainan Mudah

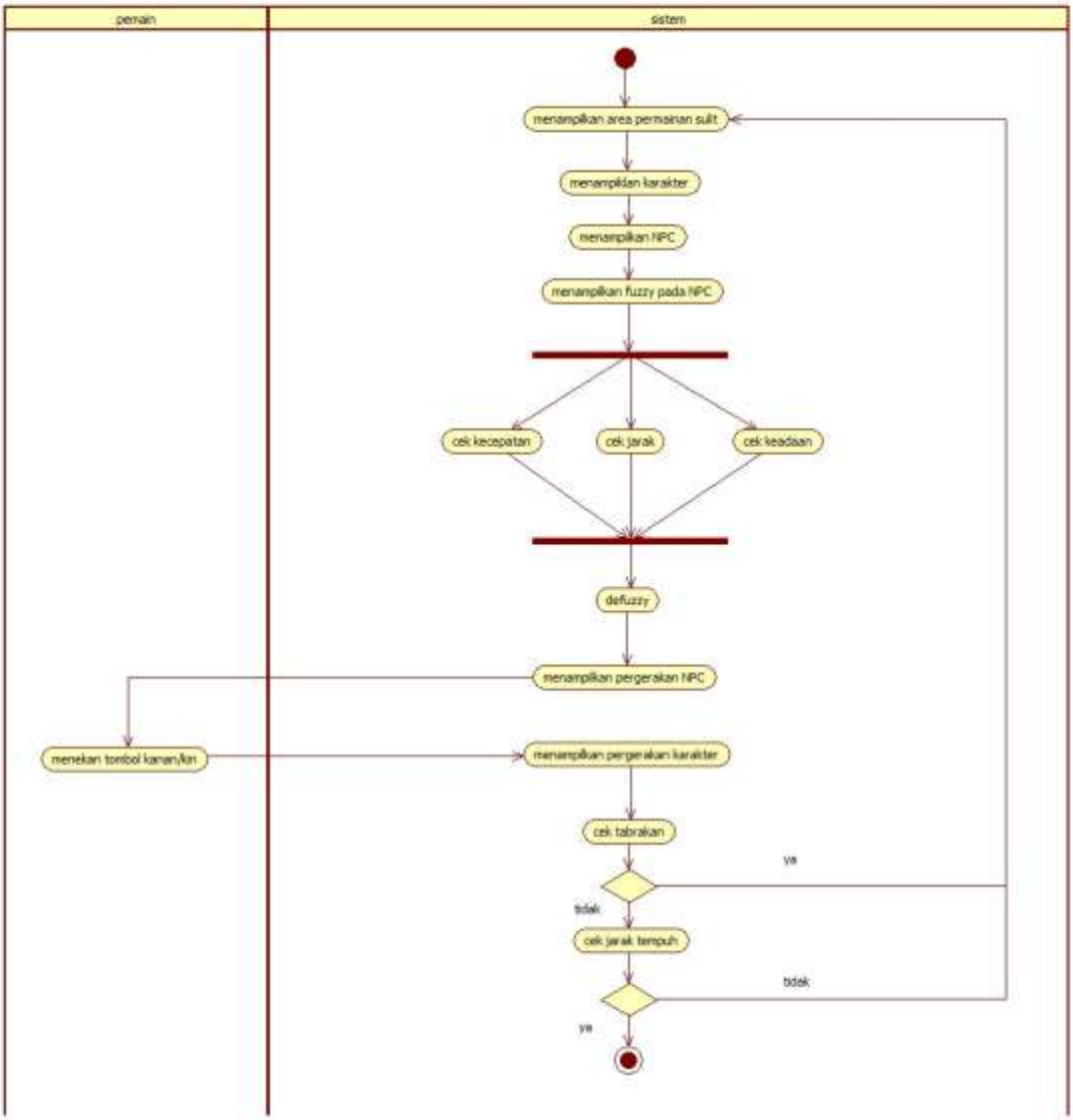
   Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas permainan. *Activity diagram* menampilkan area permainan dapat dilihat pada gambar 3.14.



Gambar 3.14 *Activity Diagram* Menampilkan Area Permainan Mudah

4. *Activity Diagram* Menampilkan Area Permainan Sulit

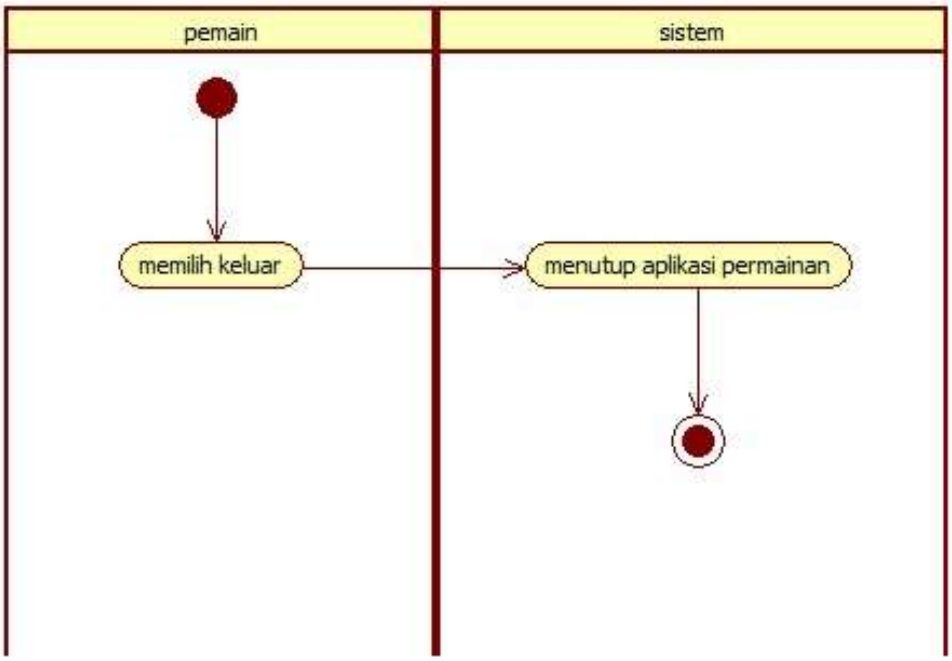
   Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas permainan. *Activity diagram* menampilkan area permainan dapat dilihat pada gambar 3.15.



Gambar 3.15 *Activity Diagram* Menampilkan Area Permainan Sulit

5. *Activity Diagram* Keluar Permainan

   Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas keluar dari permainan. *Activity diagram* keluar permainan dapat dilihat pada gambar 3.16.

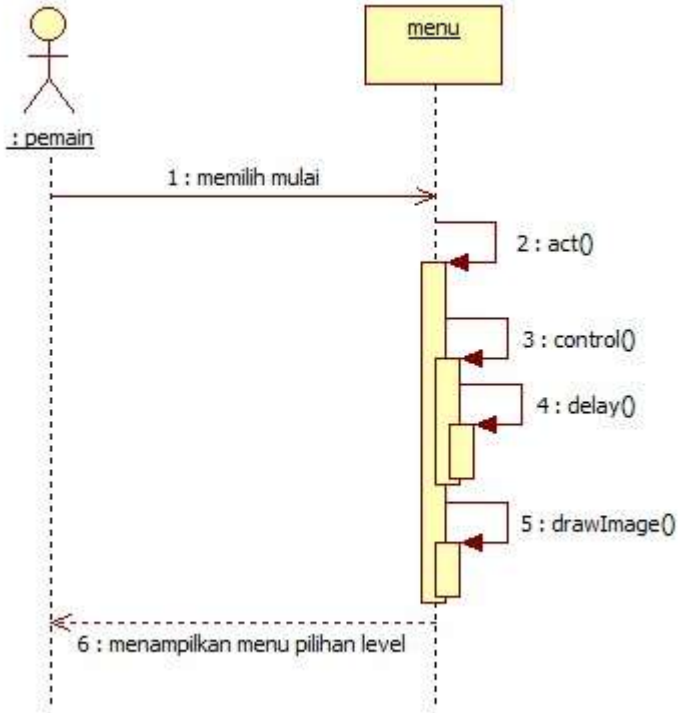


Gambar 3.16 *Activity Diagram* Keluar Permainan

3.5.2.4 *Sequence Diagram*

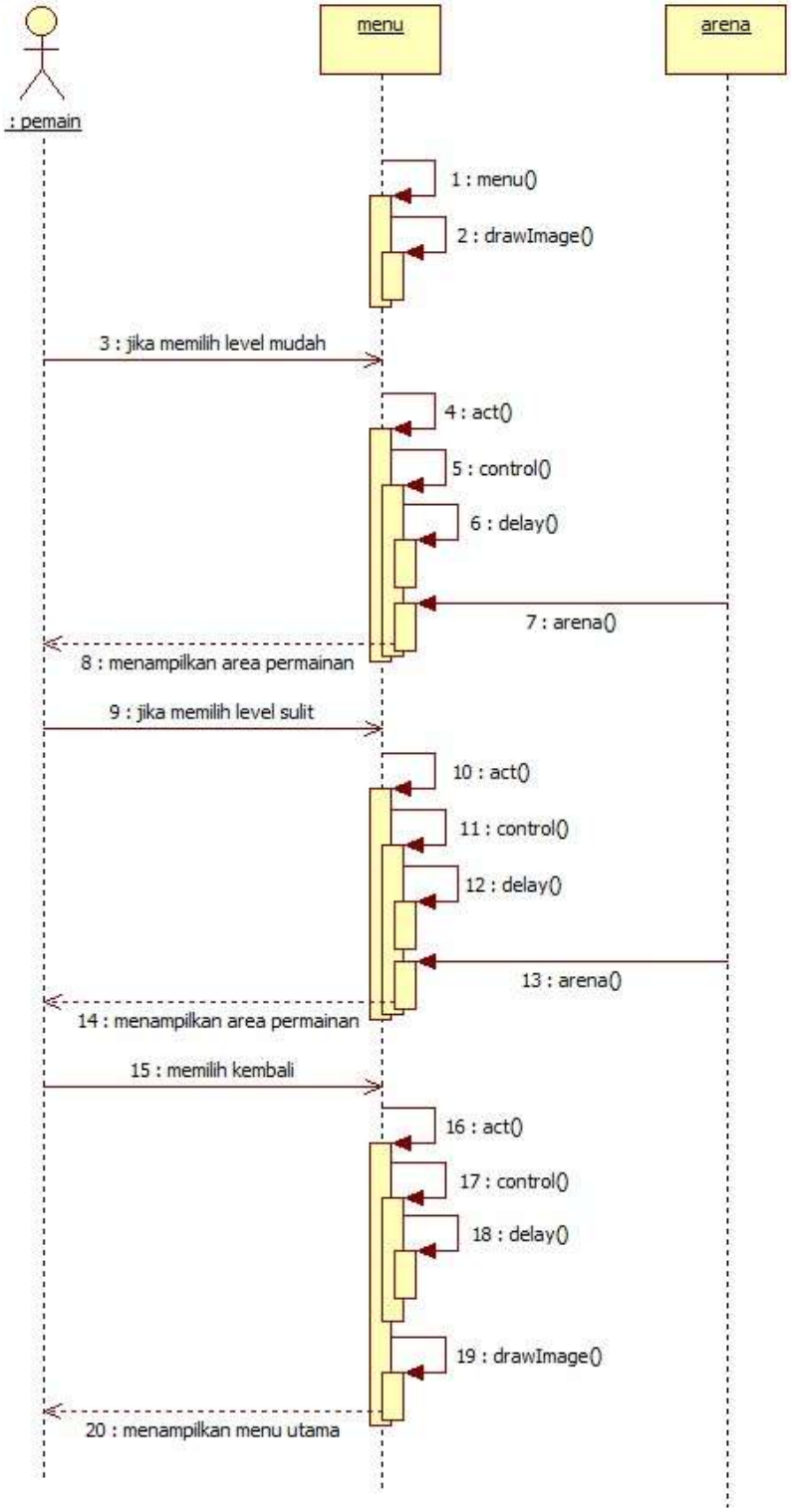
Sequence diagram menggambarkan tindakan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Berikut adalah gambar *sequence diagram.*

1. *Sequence Diagram* Mulai



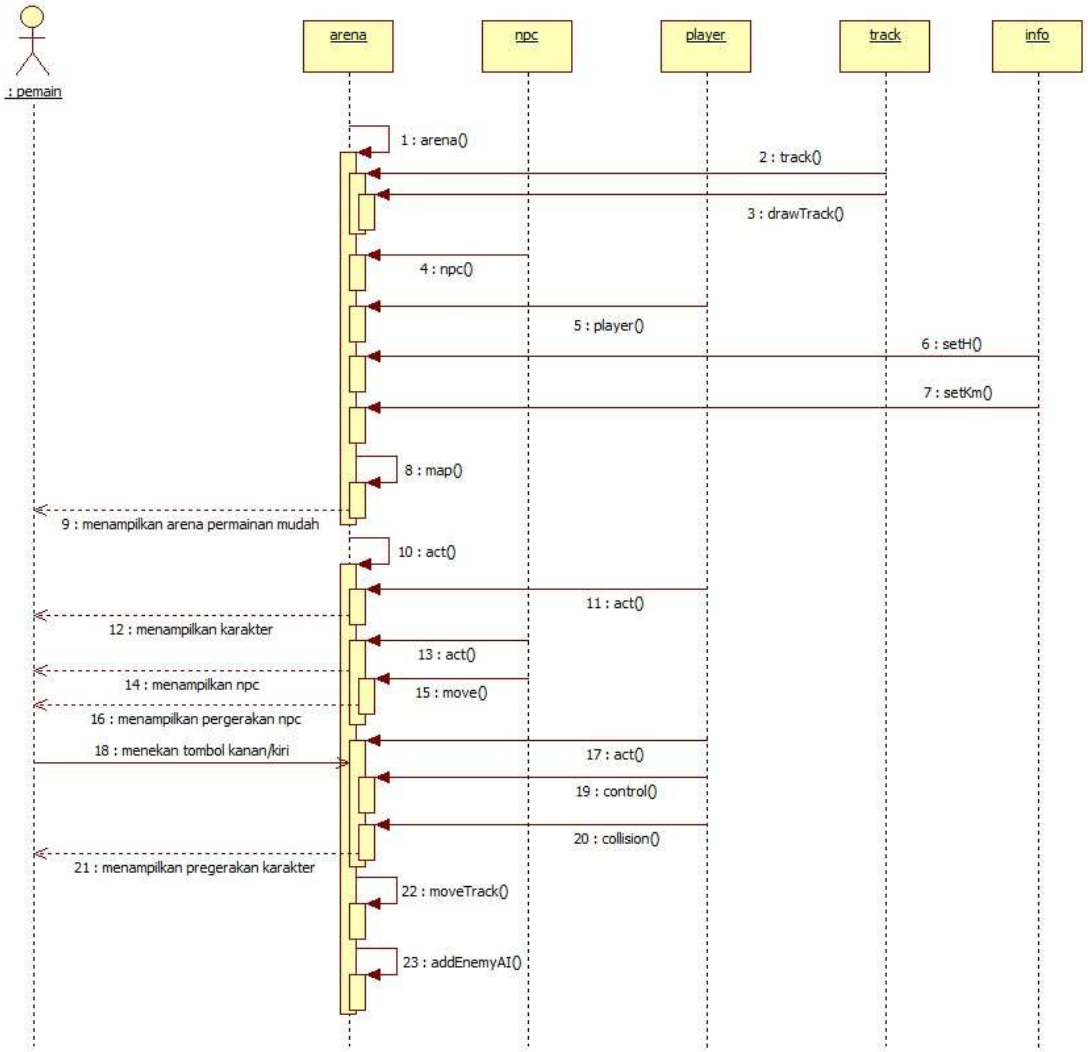
Gambar 3.17 *Sequence Diagram* Memulai Permainan

2. *Sequence Diagram* Memilih *Level*



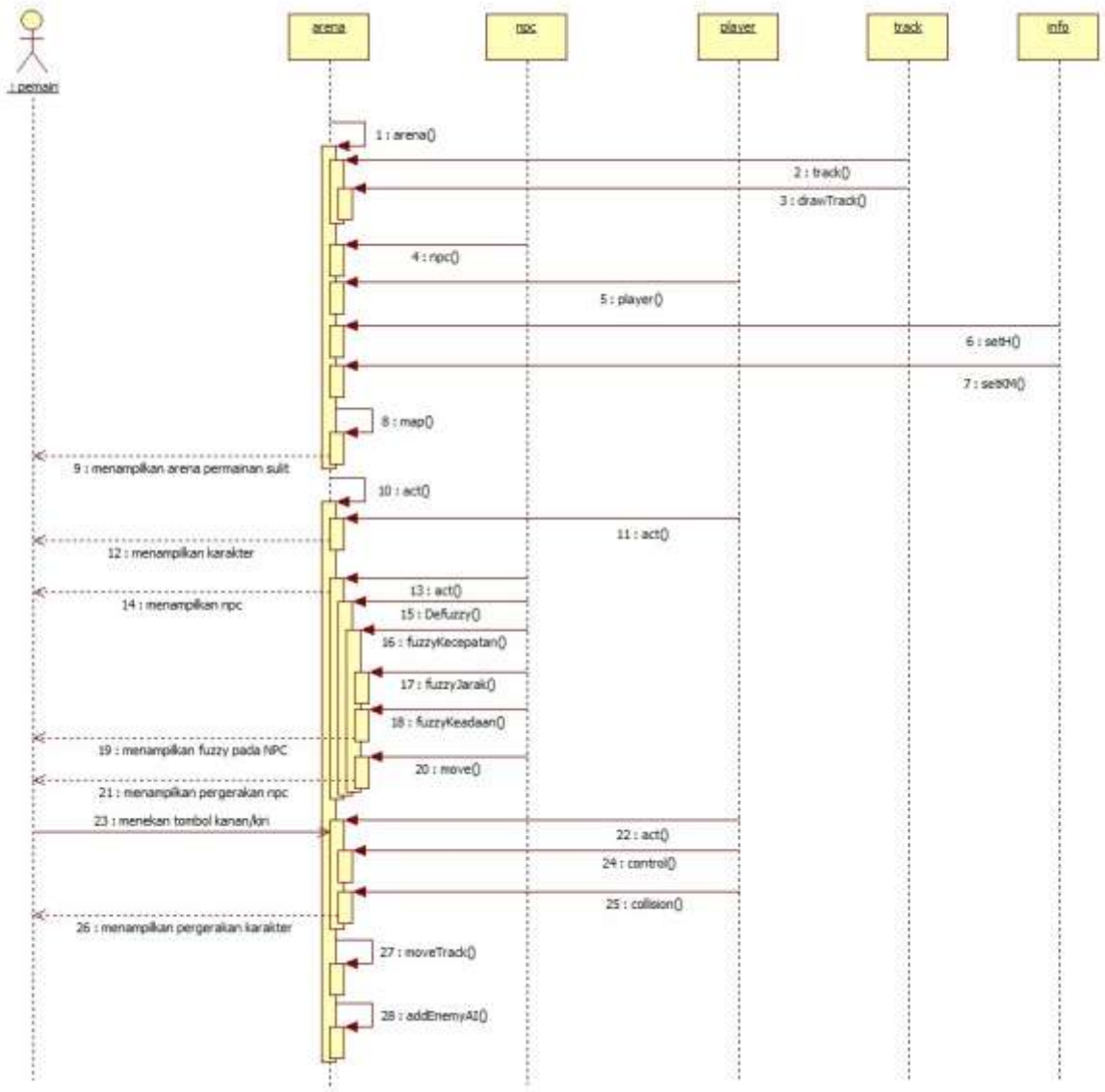
Gambar 3.18 *Sequence Diagram* Memilih *Level*

3. *Sequence Diagram* Menampilkan Area Permainan Mudah



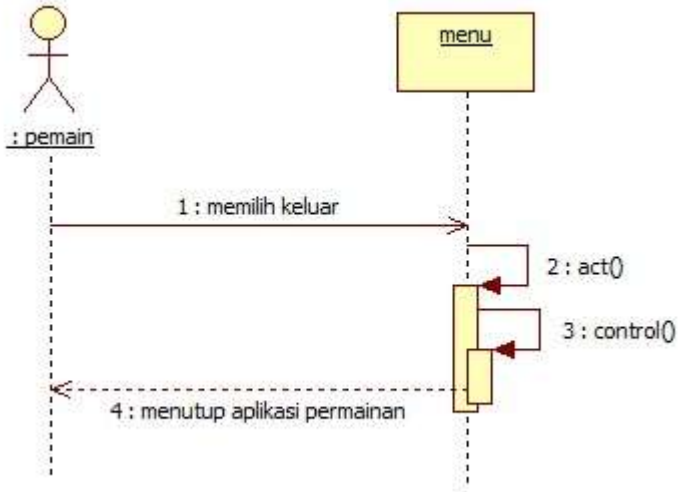
Gambar 3.19 *Sequence Diagram* Menampilkan Area Permainan Mudah

4. *Sequence Diagram* menampilkan Area Permainan Sulit



Gambar 3.20 *Sequence Diagram* Menampilkan Area Permainan Sulit

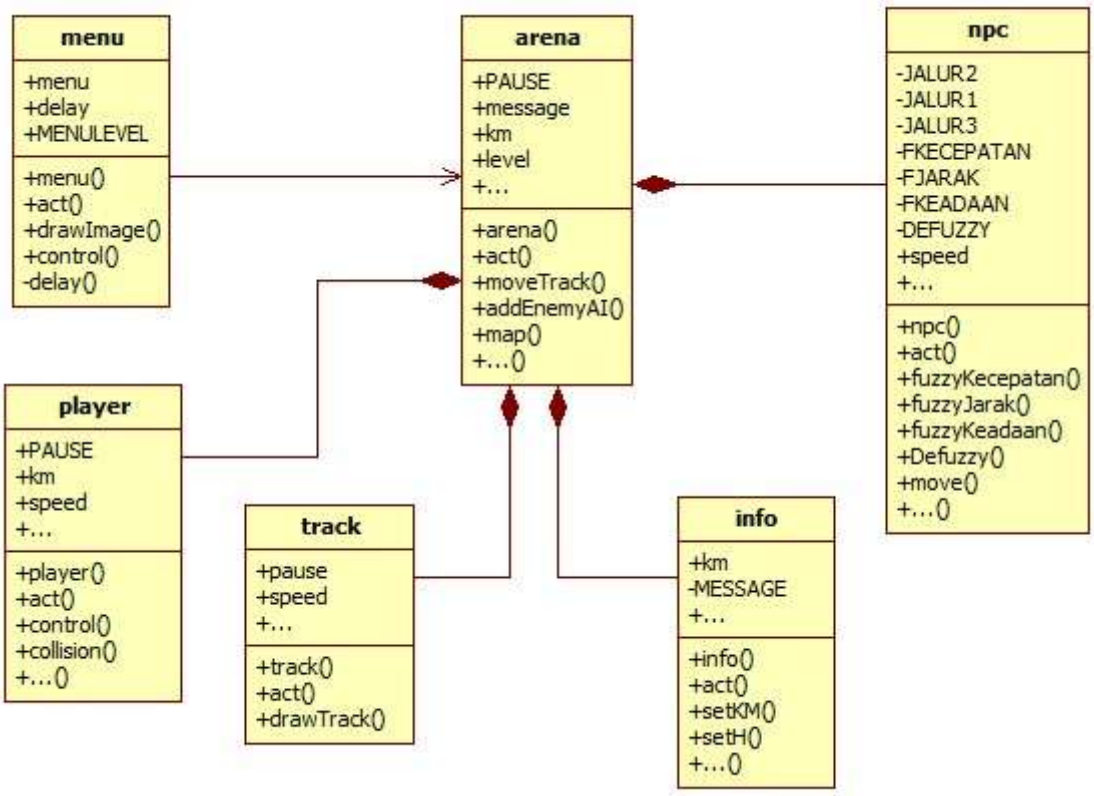
5. *Sequence Diagram* Keluar Permainan



Gambar 3.21 *Sequence Diagram* Keluar Permainan

3.5.2.5 *Class Diagram*

Class diagram merupakan gambaran sistem berupa relasi antar kelas – kelas ataupun paket – paketnya. *Class* diagram dijelaskan melalui gambar berikut.



Gambar 3.22 *Class Diagram*

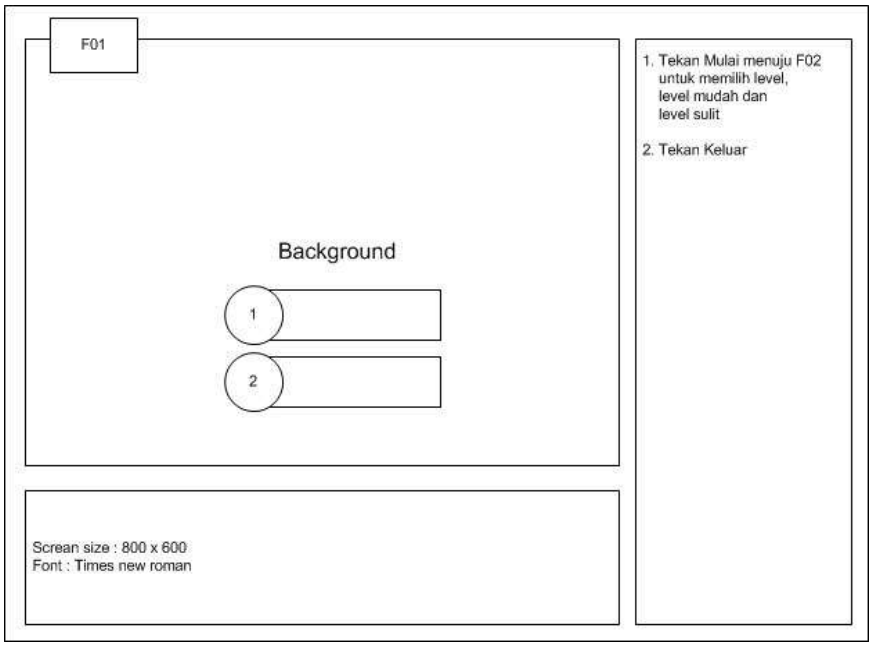
3.5.2.6 Perancangan Antarmuka

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.

1. Perancangan Antarmuka Menu Utama

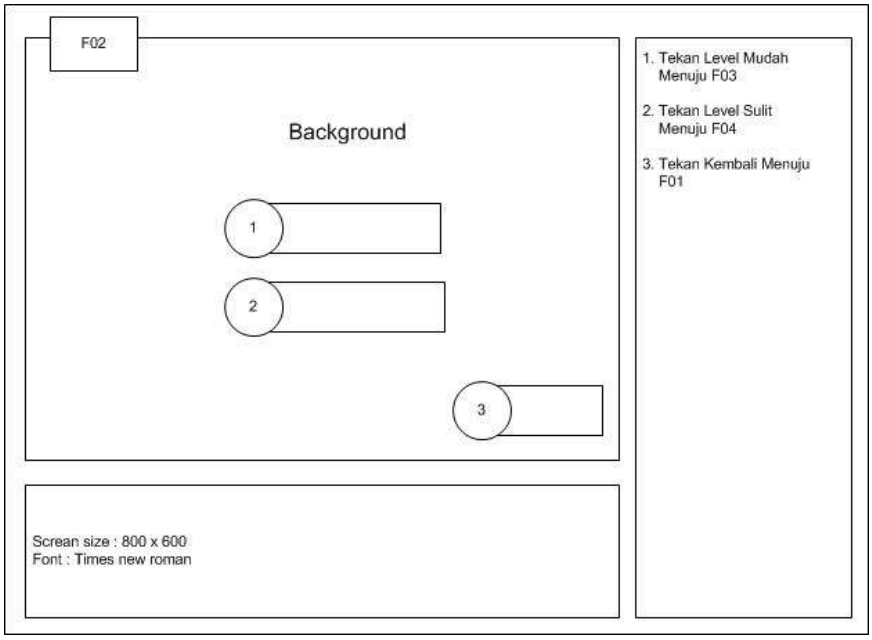
   Menu utama merupakan menu yang akan muncul pertama kali pada saat pengguna membuka *game*. Gambar 3.23 menunjukan perancangan antarmuka menu utama.



Gambar 3.23 Perancangan Antarmuka Menu Utama

2. Perancangan Antarmuka Menu Pilih *Level*

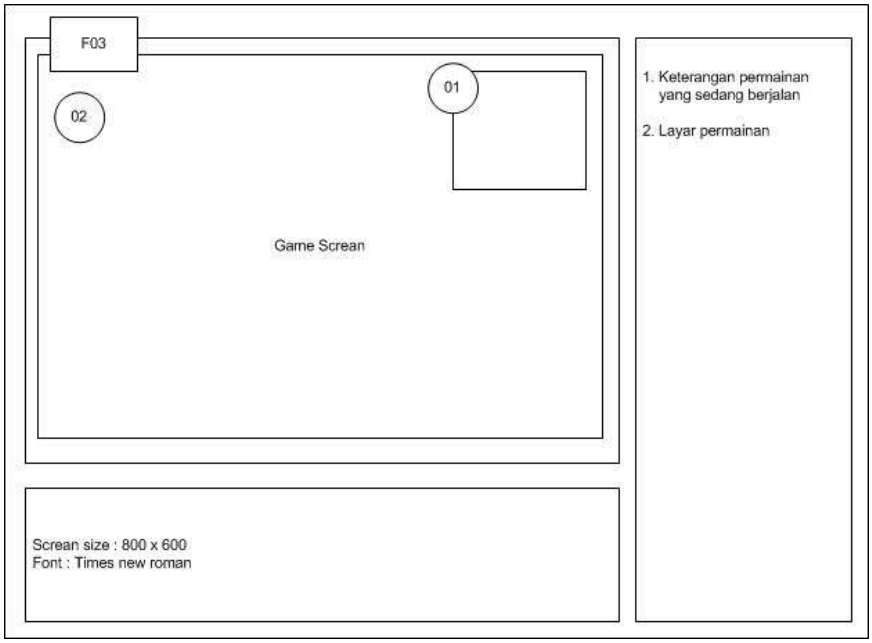
   Antarmuka menu pilih level merupakan perancangan antarmuka yang menampilkan pilih level *game* Simulasi bergenre *race*. Gambar 3.24 menunjukan perancangan antarmuka menu untuk memilih level.



Gambar 3.24 Perancangan Antarmuka Menu Plih *Level*

3. Perancangan Antarmuka *Level* Mudah

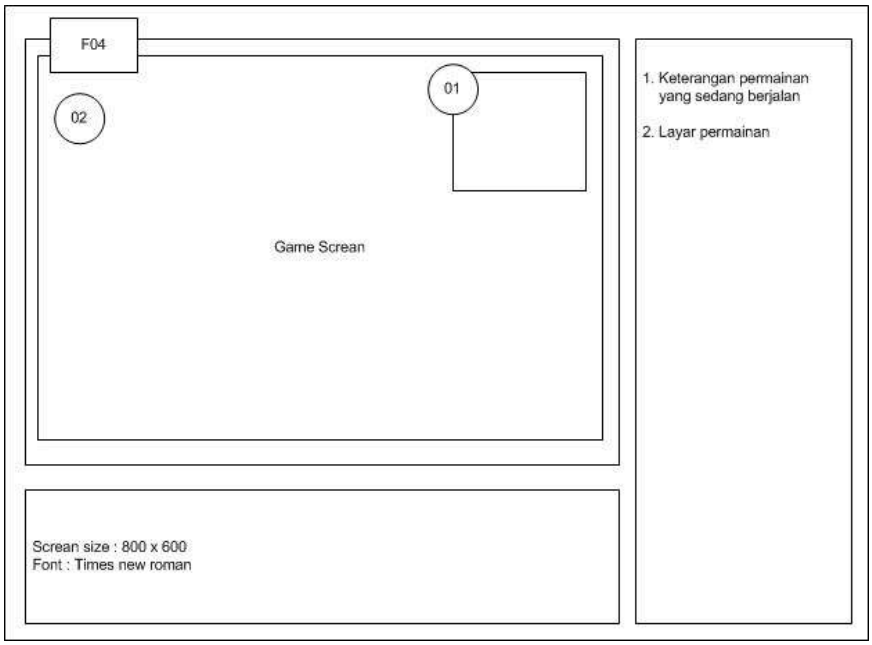
   Antarmuka *level* mudah merupakan perancangan antarmuka yang menampilkan *level* mudah permainan *game* Simulasi bergenre *race*. Gambar 3.25 menunjukan perancangan antarmuka level mudah.



Gambar 3.25 Perancangan Antarmuka *Level* Mudah

4. Perancangan Antarmuka *Level* Sulit

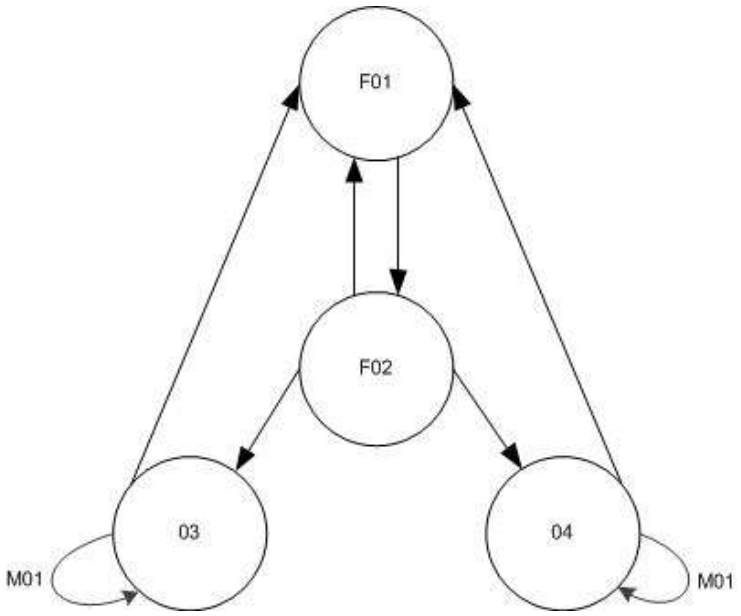
   Antarmuka *level* sulit merupakan perancangan antarmuka yang menampilkan *level* sulit permainan *game* Simulasi bergenre *race*. Gambar 3.26 menunjukan perancangan antarmuka level sulit.



Gambar 3.26 Perancangan Antarmuka *Level* Sulit

3.5.2.7 Jaringan Semantik

Untuk mengetahui keterhubungan dari setiap tampilan yang dibuat dapat menggunakan jaringan semantik. Gambar 3.27 adalah jaringan semantik pada *game* simulasi bergenre *race*.



Gambar 3.27 Jaringan Semantik