BAB 3
ANALISIS DAN KEBUTUHAN ALGORITMA

3.1 Analisis Masalah

Berdasarkan uraian pada latar belakang masalah, maka yang menjadi pokok permasalahan pada penelitian ini adalah bagaimana merancang dan menerapkan kecerdasan buatan (*artificial intellegence*) pada objek NPC *Tower* menggunakan Logika *Fuzzy*.

3.2 Analisis *Game*

Berikut ini adalah analisis *game* atau simulasi yang akan dibangun.

3.2.1 Analisis *Gameplay*

Perancangan *gameplay* pada penelitian ini pada umumnya hampir sama seperti *game-game Tower Defense* yang sudah ada sebelumnya. Tugas *user* atau pemain adalah menempatkan *tower* dan mengeluarkan musuh. yang menjadi pembeda adalah user memasukan status kekuatan unit musuh secara manual untuk mengetahui pola perilaku NPC *Tower* dalam penentuan target tembak berdasarkan variabel input atau masukan menggunakan logika *fuzzy*.

3.2.2 Identifikasi NPC

Objek NPC pada penelitian ini adalah NPC *Tower* dan NPC Musuh. Akan tetapi objek NPC yang akan dikembangkan pada penelitian adalah NPC *Tower*. Untuk lebih jelasnya, berikut ini adalah tugas dan status yang dimiliki masing – masing NPC.

1. NPC *Tower*

   NPC *Tower* memiliki tugas untuk menghentikan pergerakan unit musuh yang berusaha bergerak menuju *goal point*. Dimana NPC *Tower* memiliki beberapa status seperti:

a. *Damage*

Damage atau dmg adalah angka hit point, ketika melakukan serangan yang akan mengurangi jumlah *health point* (HP) unit musuh.

b. *Range*

Range merupakan jangkauan tembak, dimana NPC *Tower* akan menyerang sesuai dengan jangkauan tembak yang telah ditetapkan.

2. NPC Musuh

Tugas NPC Musuh adalah bergerak menuju *goal point* dan mempunyai status seperti berikut ini:

a. *Health Point* (HP)

Health point merupakan ukuran seberapa banyak damage dari NPC *Tower* yang dapat ditahan unit musuh sebelum mati.

b. *Speed* (kecepatan)

Speed pada unit musuh adalah kecepatan bergerak setiap unit untuk mencapai *goal point*.

3.2.3 Identifikasi perilaku NPC

Untuk menentukan dan mendesain teknik AI yang sesuai pada objek NPC hal pertama yang dilakukan adalah mengidentifikasi perilaku pada objek NPC yang akan dikembangkan. Penjelasan perilaku NPC akan dijabarkan pada Tabel 3.1.

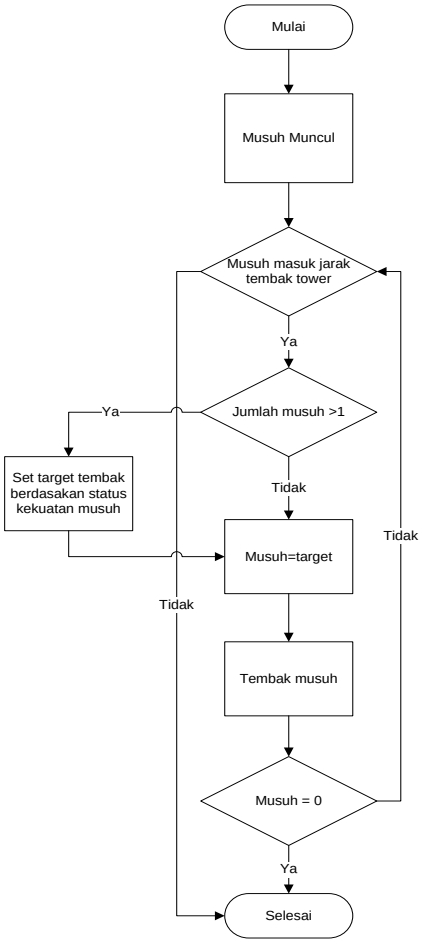
Tabel 3. 1 Identifikasi perilaku NPC *Tower*

NPC	Perilaku (Output)	Kebutuhan AI
Tower	Mengklasifikasikan tingkat bahaya musuh berdasarkan jarak dan status kekuatan musuh untuk dijadikan target tembak atau serang	Logika *Fuzzy*

3.2.4 Perancangan Perilaku NPC

Pada umumnya, unit musuh dalam *game Tower Defense* memiliki status kekuatan yang berbeda antara unit musuh satu dengan yang lainnya, perbedaan inilah yang menjadi acuan perancangan dan pengimplementasian AI pada penelitian ini.

Berdasarkan identifikasi perilaku NPC pada Tabel 3.1. Metode AI yang akan digunakan pada penelitian ini adalah logika *Fuzzy*. Penggunaan logika *Fuzzy* berfungsi untuk menentukan target tembak atau serang pada NPC *Tower* dengan acuan jarak dan status kekuatan yang ada pada musuh. Untuk lebih jelasnya. perancangan perilaku NPC *Tower* pada penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3. 1 *Flowchart* Perancangan Perilaku NPC *Tower*

3.3 Analisis Masukan

Dalam logika *Fuzzy*, variabel input yang digunakan adalah berupa interval, sehingga input yang berupa bilangan tegas harus diubah ke dalam bilangan *Fuzzy*. Sedangkan dalam himpunan tegas, input yang digunakan adalah suatu bilangan tegas.

Variabel input maupun output dibagi menjadi satu atau lebih himpunan *Fuzzy*. variabel input dibagi menjadi tiga, yaitu jarak, kecepatan dan *health point*. Serta target *priority* sebagai variabel ouput. Penentuan variabel yang digunakan dalam penelitian ini, bisa dilihat pada Tabel 3.2.

Tabel 3. 2 Variabel Input dan Output *Fuzzy*

Fungsi	Nama Variabel	Satuan	Semesta Pembicaraan	Keterangan
Input	Jarak	Pixel	[0-400]	Jarak Musuh dengan *Tower*
	Kecepatan	–	[0-10]	Kecepatan bergerak pada unit musuh
	Health Point (HP)	–	[0-1000]	*Health point* pada unit musuh
Output	Target *Priority*	–	[0-120]	Nilai target *priority* pada musuh sebagai acuan target tembak pada NPC *Tower*

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*

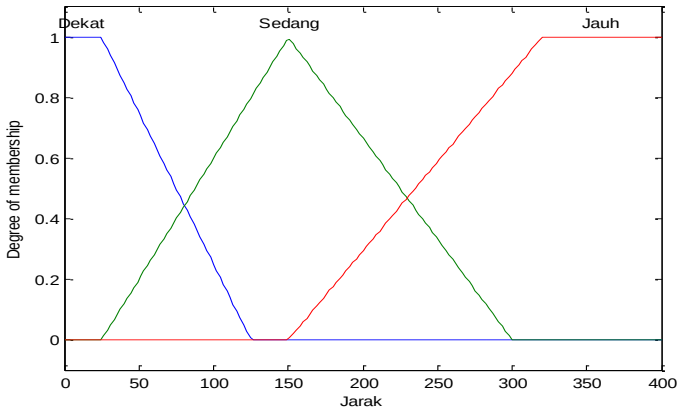
Fungsi	Variabel	Himpunan	Domain	Fungsi Keanggotaan	Parameter
Input	**Jarak**	Dekat	[0-125]	Trapesium	(0;0;25;125)
		Sedang	[25-300]	Segitiga	(25;150;300)
		Jauh	[150-400]	Trapesium	(150;320;400;400)
	Kecepatan	Lambat	[0-3]	Trapesium	(0;0;0;3)
		Normal	[1-8]	Segitiga	(1;4;8)
		Cepat	[4-10]	Trapesium	(4;8;10;10)

Fungsi	Variabel	Himpunan	Domain	Fungsi Keanggotaan	Parameter
	***Health* Point (HP)**	Sedikit	[0-500]	Trapesium	(0;0;500)
		Sedang	[200-800]	Segitiga	(200;500;800)
		Banyak	[500-1000]	Trapesium	(500;1000;1000)
Output	**Target Priority**	Sangat Rendah	[0-40]	Trapesium	(0;0;20;40)
		Rendah	[20-60]	Segitiga	(20;40;60)
		Normal	[40-80]	Segitiga	(40;60;80)
		Tinggi	[60-100]	Segitiga	(60;80;100)
		Sangat Tinggi	[80-120]	Trapesium	(80;100;120;120)

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

1. Variabel Jarak

   Variabel Jarak merupakan variabel jarak musuh dengan *tower*, variabel ini terbagi menjadi tiga himpunan *Fuzzy*, yaitu: Dekat, Sedang dan Jauh. Berikut ini fungsi keanggotaan untuk variabel jarak.



Gambar 3. 2 Fungsi Keanggotaan Variabel Jarak

Range nilai untuk variabel jarak adalah 0 - 400. Untuk lebih jelasnya, berikut ini adalah tetapan nilai untuk variabel jarak.

a. Himpunan *Fuzzy* Dekat

Himpunan *Fuzzy* dekat memiliki domain nilai antara 0 – 125, dimana fungsi keanggotaannya dihitung berdasarkan fungsi trapesium.

$$\mu Dekat(x) = \begin{cases} 1; & x \leq 25 \\ \frac{125-x}{125-25}; & 25 \leq x \leq 125 \\ 0; & x \geq 125 \end{cases} \quad (3.1)$$

b. Himpunan *Fuzzy* Sedang

Himpunan *Fuzzy* Sedang memiliki domain nilai antara 25-300, dimana fungsi keanggotaannya dihitung berdasarkan fungsi segitiga.

$$\mu Sedang(x) = \begin{cases} 0; & x \leq 25 \; atau \; x \geq 300 \\ \frac{x-25}{150-25}; & 25 \leq x \leq 150 \\ \frac{150-x}{300-150}; & 150 \leq x \leq 300 \end{cases} \quad (3.2)$$

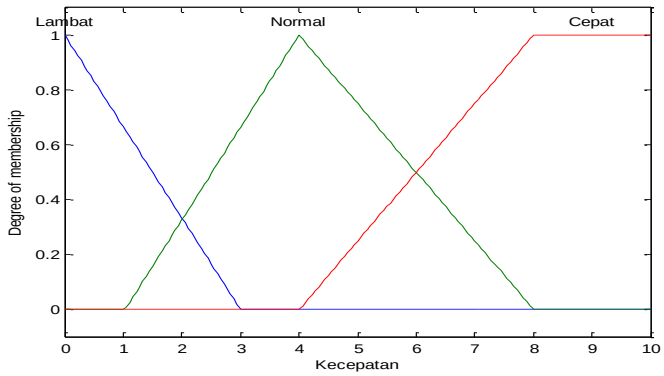
c. Himpunan *Fuzzy* Jauh

Himpunan *Fuzzy* Jauh memiliki domain nilai antara 150-400, dimana fungsi keanggotaannya dihitung berdasarkan fungsi trapesium.

$$\mu Jauh(x) = \begin{cases} 0; & x \leq 150 \\ \frac{x-150}{300-150}; & 150 \leq x \leq 300 \\ 1; & x \geq 300 \end{cases} \quad (3.3)$$

2. Variabel Kecepatan

Variabel Kecepatan, merupakan kecepatan gerak yang ada pada unit musuh, variabel ini terbagi menjadi tiga himpunan *Fuzzy*, yaitu : Lambat, Normal dan Cepat. Untuk lebih jelasnya dapat dilihat pada Gambar 3.3.



Gambar 3. 3 Fungsi Keanggotaan Variabel Kecepatan

Range nilai atau semesta pembicaraan untuk variabel kecepatan adalah 1-10. Untuk lebih jelasnya, berikut ini adalah tetapan nilai untuk variabel kecepatan.

a. Himpunan *Fuzzy* Lambat

Himpunan *Fuzzy* Lambat memiliki domain nilai antara 0–3, dimana fungsi keanggotaannya dihitung berdasarkan fungsi segitiga.

$$\mu Lambat(x) = \begin{cases} \frac{3-x}{3-0}; & 0 \leq x \leq 3 \\ 0; & x \geq 3 \end{cases} \quad (3.4)$$

b. Himpunan *Fuzzy* Normal

Himpunan *Fuzzy* Normal memiliki domain nilai antara 1 - 8, dimana fungsi keanggotaannya dihitung berdasarkan fungsi segitiga.

$$\mu Normal(x) = \begin{cases} 0; & x \leq 1 \; atau \; x \geq 8 \\ \frac{x-1}{4-1}; & 1 \leq x \leq 4 \\ \frac{4-x}{8-4}; & 4 \leq x \leq 8 \end{cases} \quad (3.5)$$

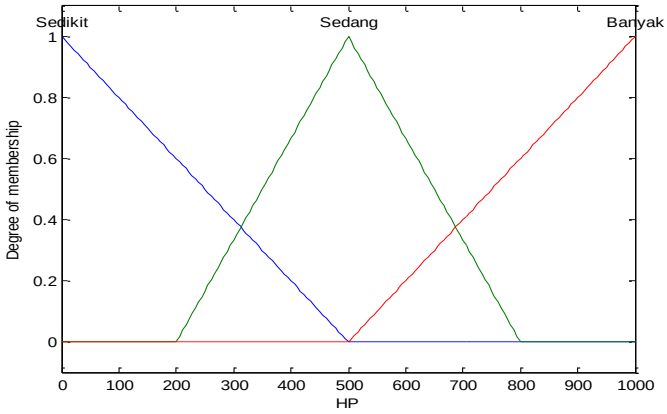
c. Himpunan *Fuzzy* Cepat

Himpunan *Fuzzy* Cepat memiliki domain nilai antara 4 - 10, dimana fungsi keanggotaannya dihitung berdasarkan fungsi trapesium.

$$\mu Cepat(x) = \begin{cases} 0; & x \leq 4 \\ \frac{x-4}{8-4}; & 4 \leq x \leq 8 \\ 1; & x \geq 8 \end{cases} \quad (3.6)$$

3. Variabel HP

Variabel HP (*Health Point*), terbagi menjadi tiga himpunan *Fuzzy*, yaitu : Sedikit, Sedang dan Banyak. Berikut ini fungsi keanggotaan untuk Variabel HP.



Gambar 3. 4 Fungsi Keanggotaan Variabel HP

Range nilai atau semesta pembicaraan untuk variabel HP adalah 0-1000. Untuk lebih jelasnya, berikut ini adalah tetapan nilai untuk variabel HP.

a. Himpunan *Fuzzy* Sedikit

Himpunan *Fuzzy* Sedikit memiliki domain nilai antara 0 - 500, dimana fungsi keanggotaannya dihitung berdasarkan fungsi trapesium.

$$\mu Sedikit(x) = \begin{cases} \frac{500-x}{500-0}; & 0 \leq x \leq 500 \\ 0; & x \geq 500 \end{cases} \quad (3.7)$$

b. Himpunan *Fuzzy* Sedang

Himpunan *Fuzzy* Sedang memiliki domain nilai antara 200 - 800, dimana fungsi keanggotaannya dihitung berdasarkan fungsi segitiga.

$$\mu Sedang(x) = \begin{cases} 0; & x \leq 200 \; atau \; x \geq 800 \\ \frac{x-200}{500-200}; & 200 \leq x \leq 500 \\ \frac{500-x}{800-500}; & 500 \leq x \leq 800 \end{cases} \quad (3.8)$$

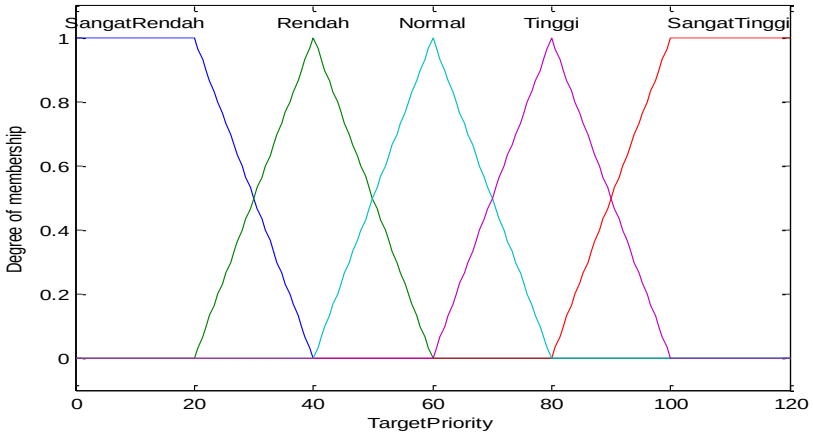
c. Himpunan *Fuzzy* Banyak

Himpunan *Fuzzy* Banyak memiliki domain nilai antara 500 - 1000, dimana fungsi keanggotaannya dihitung berdasarkan fungsi segitiga.

$$\mu Banyak(x) = \begin{cases} 0; & x \leq 500 \\ \frac{x-500}{1000-500}; & 500 \leq x \leq 1000 \\ 1; & x \geq 1000 \end{cases} \quad (3.9)$$

4. Variabel Target *Priority*

Variabel Target *Priority*, merupakan variabel *output*, dimana nilai dari variabel ini, akan menjadi acuan NPC *Tower* dalam menentukan target tembak. Variabel target *priority* memiliki empat himpunan *Fuzzy*, yaitu : rendah, normal, tinggi dan sangat tinggi. Untuk lebih jelasnya, berikut ini adalah fungsi keanggotaan untuk variabel target *priority*:



Gambar 3. 5 Fungsi Keanggotaan Variabel Target

Range nilai atau semesta pembicaraan untuk variabel target *priority* adalah 0 - 120. Berikut ini adalah tetapan nilai untuk variabel target *priority*.

a. Himpunan *Fuzzy* Sangat Rendah

Himpunan *Fuzzy* Sangat Rendah memiliki domain nilai antara 0 - 40, dimana fungsi keanggotaannya dihitung berdasarkan fungsi trapesium.

$$\mu SangatRendah(z) = \begin{cases} 1; & z \leq 20 \\ \frac{40-z}{40-20}; & 20 \leq z \leq 40 \\ 0; & z \geq 40 \end{cases} \quad (3.10)$$

b. Himpunan *Fuzzy* Rendah

Himpunan *Fuzzy* Rendah memiliki domain nilai antara 20 - 60, dimana fungsi keanggotaannya dihitung berdasarkan fungsi segitiga.

$$\mu Rendah(z) = \begin{cases} 0; & z \leq 20 \; atau \; z \geq 60 \\ \frac{z-20}{40-20}; & 20 \leq z \leq 40 \\ \frac{40-z}{60-40}; & 40 \leq z \leq 60 \end{cases} \quad (3.11)$$

c. Himpunan *Fuzzy* Normal

Himpunan *Fuzzy* Normal memiliki domain nilai antara 40 - 80, dimana fungsi keanggotaannya dihitung berdasarkan fungsi segitiga.

$$\mu Normal(z) = \begin{cases} 0; & z \leq 40 \; atau \; z \geq 80 \\ \frac{z-40}{60-40}; & 40 \leq z \leq 60 \\ \frac{60-z}{80-60}; & 60 \leq z \leq 80 \end{cases} \quad (3.12)$$

d. Himpunan *Fuzzy* Tinggi

Himpunan *Fuzzy* Tinggi memiliki domain nilai antara 60 - 100, dimana fungsi keanggotaannya dihitung berdasarkan fungsi segitiga.

$$\mu Tinggi(z) = \begin{cases} 0; & z \leq 60 \; atau \; z \geq 100 \\ \frac{z-60}{80-60}; & 60 \leq z \leq 80 \\ \frac{80-z}{100-80}; & 80 \leq z \leq 100 \end{cases} \quad (3.14)$$

e. Himpunan *Fuzzy* SangatTinggi

Himpunan *Fuzzy* SangatTinggi memiliki domain nilai antara 80 - 120, dimana fungsi keanggotaannya dihitung berdasarkan fungsi trapesium.

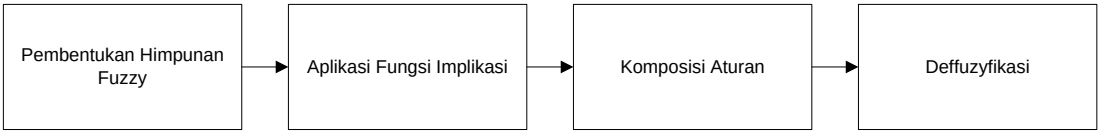
$$\mu SangatTinggi(z) = \begin{cases} 0; & z \leq 80 \\ \frac{z-80}{100-80}; & 80 \leq z \leq 100 \\ 1; & z \geq 100 \end{cases} \quad (3.15)$$

3.4 Analisis Logika *Fuzzy*

Pada penelitian ini logika *Fuzzy* digunakan untuk menentukan target tembak pada NPC *Tower*, dengan mengolah input *Fuzzy* menjadi output *Fuzzy*. dimana nilai output akan menjadi acuan NPC *Tower* dalam menentukan target tembak.

3.4.1 *Fuzzy* Mamdani

Dalam penentuan target tembak pada NPC *Tower* digunakan metode inferensi *Fuzzy* Mamdani, dengan tahapan sebagai berikut :



Gambar 3. 6 Tahapan Inferensi *Fuzzy* Mamdani

Berikut ini adalah contoh kasus penentuan target tembak pada NPC *Tower* menggunakan logika *Fuzzy*, dengan sistem inferensi Mamdani.

3.4.2 Contoh Penerapan

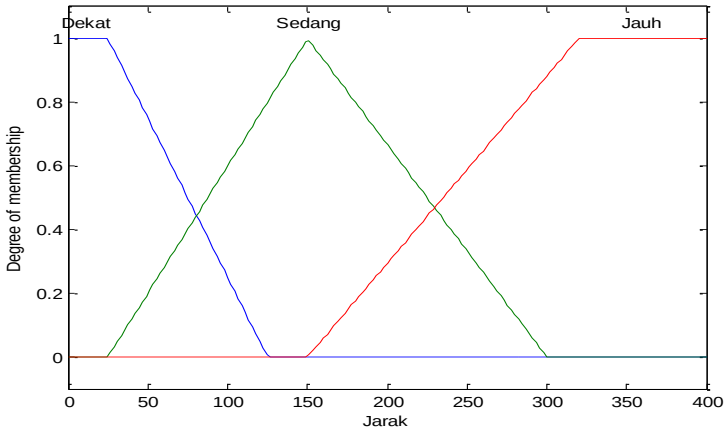
Jika jarak unit musuh pada *tower* adalah 280, kecepatan 5 dan HP 540 maka nilai target priority dari musuh tersebut adalah sebagai berikut:

3.4.2.1 Menentukan Himpunan *Fuzzy*

Himpunan *Fuzzy* beserta fungsi keanggotaan dari variabel jarak, kecepatan dan HP direpresentasikan sebagai berikut :

1. Variabel *Fuzzy* Jarak

   Perhitungan variabel jarak dari setiap himpunan *Fuzzy* (Dekat, Sedang dan Jauh) dimana nilai jarak [280] adalah sebagai berikut:



Gambar 3. 7 Fungsi Keanggotaan untuk Jarak 280

Jarak sebesar 280 termasuk ke dalam himpunan *Fuzzy* Sedang dan Jauh, dengan tingkat keanggotaan sesuai fungsi berikut:

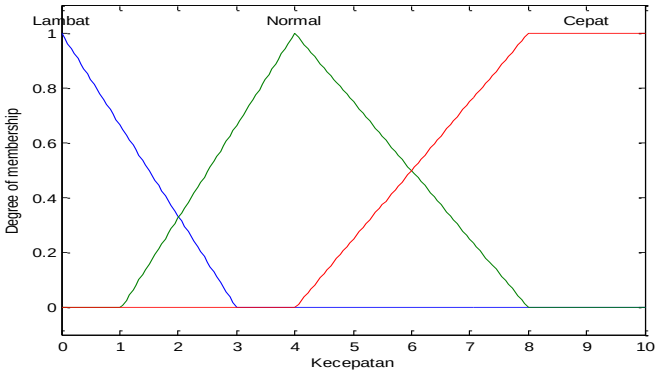
$$\mu Dekat[280] = 280 \geq 125 = 0$$

$$\mu Sedang[280] = \frac{300 - 280}{300 - 150} = 0.13$$

$$\mu Jauh[280] = \frac{280 - 150}{300 - 150} = 0.86$$

2. Variabel *Fuzzy* Kecepatan

Perhitungan variabel Fuzzy kecepatan dari setiap himpunan *Fuzzy* Lambat, Normal dan Cepat dimana nilai Kecepatan [5] adalah sebagai berikut :



Gambar 3. 8 Fungsi Keanggotaan untuk Kecepatan 5

Kecepatan sebesar 5 termasuk ke dalam himpunan *Fuzzy* Normal dan Cepat, dengan tingkat keanggotaan sesuai fungsi berikut:

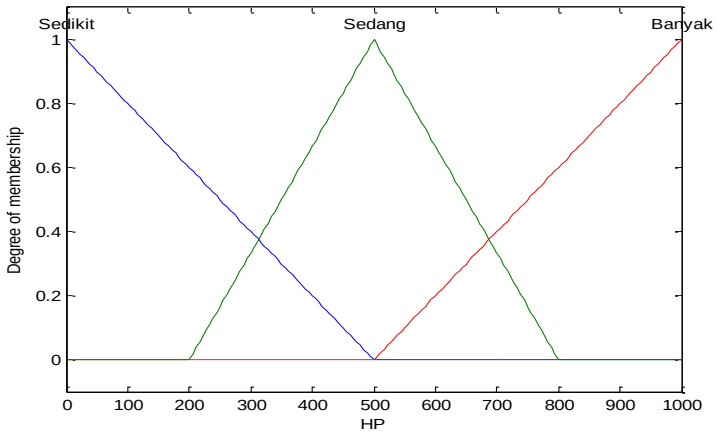
$$\mu Lambat[5] = 5 \geq 3 = 0$$

$$\mu Normal[5] = \frac{8-5}{8-4} = 0.75$$

$$\mu Cepat[5] = \frac{5-4}{8-4} = 0.25$$

3. Variabel *Fuzzy* HP

Perhitungan variabel *Fuzzy* HP dari himpunan *Fuzzy* Sedikit, Sedang dan Banyak dimana nilai HP [540] adalah sebagai berikut :



Gambar 3. 9 Fungsi Keanggotaan untuk HP 540

HP sebesar 540 termasuk ke dalam himpunan *Fuzzy* Sedang dan Banyak, dengan tingkat keanggotaan sesuai fungsi berikut:

$$\mu Sedikit[540] = 540 \geq 500 = 0$$

$$\mu Sedang[540] = \frac{800-540}{800-500} = 0.86$$

$$\mu Banyak[540] = \frac{540-500}{1000-500} = 0.08$$

3.4.2.2 Fungsi Implikasi

Setelah pembentukan himpunan *Fuzzy*, maka dilakukan pembentukan aturan *Fuzzy*. Aturan - aturan dibentuk untuk menyatakan relasi antara input dan output. Tiap aturan merupakan suatu implikasi. Operator yang digunakan untuk menghubungkan antara dua input adalah operator *AND*, dan yang memetakan antara input-output adalah *IF-THEN*. Proposisi yang mengikuti *IF* disebut anteseden, sedangkan proposisi yang mengikuti *THEN* disebut konsekuen. Berikut ini adalah tetapan aturan pemilihan target tembak pada NPC *Tower*.

Tabel 3. 4 Tetapan Aturan Pemilihan Target Tembak Pada NPC *Tower*

ANTESEDEN				**KONSEKUEN**	
Operator	**Jarak**	**Operator**	**Kecepatan**	**Operator**	**Target *Priority***
IF	Dekat	**AND**	Cepat Normal Lambat	**THEN**	Sangat Tinggi Tinggi Normal
	Sedang		Cepat Normal Lambat		Tinggi Normal Rendah
	Jauh		Cepat Normal Lambat		Normal Rendah SangatRendah
Operator	**Jarak**	**Operator**	**HP**	**Operator**	**Target *Priority***
IF	Dekat	**AND**	Sedikit Sedang Banyak	**THEN**	Sangat Tinggi Tinggi Normal
	Sedang		Sedikit Sedang Banyak		Tinggi Normal Rendah
	Jauh		Sedikit Sedang Banyak		Normal Rendah SangatRendah

Berdasarkan tetapan aturan pada Tabel 3.4, maka pembentukan aturan dalam pemilihan target tembak pada NPC *Tower* adalah sebagai berikut :

[R1] IF Jarak DEKAT AND Kecepatan CEPAT THEN TargetPriority SANGATTINGGI;

[R2] IF Jarak DEKAT AND Kecepatan NORMAL THEN TargetPriority TINGGI;

[R3] IF Jarak DEKAT AND Kecepatan LAMBAT THEN TargetPriority NORMAL;

[R4] IF Jarak SEDANG AND Kecepatan CEPAT THEN TargetPriority TINGGI;

[R5] IF Jarak SEDANG AND Kecepatan NORMAL THEN TargetPriority NORMAL ;

[R6] IF Jarak SEDANG AND Kecepatan LAMBAT THEN TargetPriority RENDAH;

[R7] IF Jarak JAUH AND Kecepatan CEPAT THEN TargetPriority NORMAL;

[R8] IF Jarak JAUH AND Kecepatan NORMAL THEN TargetPriority RENDAH;

[R9] IF Jarak JAUH AND Kecepatan LAMBAT THEN TargetPriority SANGATRENDAH ;

[R10] IF Jarak DEKAT AND HP BANYAK THEN TargetPriority SANGATTINGGI;

[R11] IF Jarak DEKAT AND HP SEDANG THEN TargetPriority TINGGI;

[R12] IF Jarak DEKAT AND HP SEDIKIT THEN TargetPriority NORMAL;

[R13] IF Jarak SEDANG AND HP BANYAK THEN TargetPriority TINGGI;

[R14] IF Jarak SEDANG AND HP SEDANG THEN TargetPriority NORMAL;

[R15] IF Jarak SEDANG AND HP SEDIKIT THEN TargetPriority RENDAH;

[R16] IF Jarak JAUH AND HP BANYAK THEN TargetPriority NORMAL;

[R17] IF Jarak JAUH AND HP SEDANG THEN TargetPriority RENDAH;

[R18] IF Jarak JAUH AND HP SEDIKIT THEN TargetPriority SANGATRENDAH;

Setelah aturan dibentuk, maka dilakukan aplikasi fungsi implikasi. Pada Metode Mamdani, fungsi implikasi yang digunakan adalah MIN, yang berarti tingkat keanggotaan yang didapat sebagai konsekuen dari proses ini adalah nilai minimum dari variabel jarak, kecepatan dan HP. Untuk lebih jelasnya dapat dilihat pada Tabel 3.5 dibawah ini.

Tabel 3. 5 Tabel Fungsi Implikasi

No	Fungsi Implikasi	
[R1]	**IF** Jarak **IS** Dekat **AND** Kecepatan **IS** Cepat **THEN** TargetPriority **IS** Sangat Tinggi	
	$\alpha - Predikat_1$	$= \mu_{JrkDekat \cap KcpCepat}$ $= \min(\mu_{JrkDekat}(280);\ \mu_{KcpCepat}(5))$ $= \min(0; 0.25)$ $= 0$
[R2]	**IF** Jarak **IS** Dekat **AND** Kecepatan **IS** Normal **THEN** TargetPriority **IS** Tinggi	
	$\alpha - Predikat_2$	$= \mu_{JrkDekat \cap KcpNormal}$ $= \min(\mu_{JrkDekat}(280);\ \mu_{KcpNormal}(5))$ $= \min(0; 0.75)$ $= 0$
[R3]	**IF** Jarak **IS** Dekat **AND** Kecepatan **IS** Lambat **THEN** TargetPriority **IS** Normal	
	$\alpha - Predikat_3$	$= \mu_{JrkDekat \cap KcpLambat}$ $= \min(\mu_{JrkDekat}(280);\ \mu_{KcpLambat}(5))$ $= \min(0; 0)$ $= 0$
[R4]	**IF** Jarak **IS** Sedang **AND** Kecepatan **IS** Cepat **THEN** TargetPriority **IS** Tinggi	
	$\alpha - Predikat_4$	$= \mu_{JrkSedang \cap KcpCepat}$ $= \min(\mu_{JrkSedang}(280);\ \mu_{KcpCepat}(5))$ $= \min(0.13; 0.25)$ $= 0.13$
[R5]	**IF** Jarak **IS** Sedang **AND** Kecepatan **IS** Normal **THEN** TargetPriority is Normal	
	$\alpha - Predikat_5$	$= \mu_{JrkSedang \cap KcpNormal}$ $= \min(\mu_{JrkSedang}(280);\ \mu_{KcpNormal}(5))$ $= \min(0.13; 0.75)$ $= 0.13$
[R6]	**IF** Jarak **IS** Sedang **AND** Kecepatan **IS** Lambat **THEN** TargetPriority **IS** Rendah	
	$\alpha - Predikat_6$	$= \mu_{JrkSedang \cap KcpLambat}$ $= \min(\mu_{JrkSedang}(280);\ \mu_{KcpLambat}(5))$ $= \min(0.13; 0)$ $= 0$
[R7]	**IF** Jarak **IS** Jauh **AND** Kecepatan **IS** Cepat **THEN** TargetPriority **IS** Normal	
	$\alpha - Predikat_7$	$= \mu_{JrkJauh \cap KcpCepat}$

		$= \min(\mu_{JrkJauh(280)};\ \mu_{KcpCepat(5)})$ $= \min(0.86; 0.25)$ $= 0.25$
[R8]	**IF** Jarak **IS** Jauh **AND** Kecepatan **IS** Normal **THEN** TargetPriority **IS** Rendah	
	$\alpha - Predikat_8$	$= \mu_{JrkJauh \cap KcpNormal}$ $= \min(\mu_{JrkJauh(280)};\ \mu_{KcpNormal(5)})$ $= \min(0.86; 0.75)$ $= 0.75$
[R9]	**IF** Jarak **IS** Jauh **AND** Kecepatan **IS** Lambat **THEN** TargetPriority **IS** SangatRendah	
	$\alpha - Predikat_9$	$= \mu_{JrkJauh \cap KcpLambat}$ $= \min(\mu_{JrkJauh(280)};\ \mu_{KcpLambat(5)})$ $= \min(0.86; 0)$ $= 0$
[R10]	**IF** Jarak **IS** Dekat **AND** HP **IS** Banyak **THEN** TargetPriority **IS** SangatTinggi	
	$\alpha - Predikat_{10}$	$= \mu_{JrkDekat \cap HPBanyak}$ $= \min(\mu_{JrkDekat(280)};\ \mu_{HPBanyak(540)})$ $= \min(0; 0.08)$ $= 0$
[R11]	**IF** Jarak **IS** Dekat **AND** HP is Sedang **THEN** TargetPriority **IS** Tinggi	
	$\alpha - Predikat_{11}$	$= \mu_{JrkDekat \cap HPSedang}$ $= \min(\mu_{JrkDekat(280)};\ \mu_{HPSedang(540)})$ $= \min(0; 0.86)$ $= 0$
[R12]	**IF** Jarak **IS** Dekat **AND** HP **IS** Sedikit **THEN** TargetPriority **IS** Normal	
	$\alpha - Predikat_{12}$	$= \mu_{JrkDekat \cap HPSedikit}$ $= \min(\mu_{JrkDekat(280)};\ \mu_{HPSedikit(540)})$ $= \min(0; 0)$ $= 0$
[R13]	**IF** Jarak **IS** Sedang **AND** HP **IS** Banyak **THEN** TargetPriority **IS** Tinggi	
	$\alpha - Predikat_{13}$	$= \mu_{JrkSedang \cap HPBanyak}$ $= \min(\mu_{JrkSedang(280)};\ \mu_{HPBanyak(540)})$ $= \min(0.13; 0.08)$ $= 0.08$
[R14]	**IF** Jarak **IS** Sedang **AND** HP **IS** Sedang **THEN** TargetPriority **IS** Normal	
	$\alpha - Predikat_{14}$	$= \mu_{JrkSedang \cap HPSedang}$ $= \min(\mu_{JrkSedang(280)};\ \mu_{HPSedang(540)})$ $= \min(0.13; 0.86)$ $= 0.13$
[R15]	**IF** Jarak **IS** Sedang **AND** HP **IS** Sedikit **THEN** TargetPriority **IS** Rendah	
	$\alpha - Predikat_{15}$	$= \mu_{JrkSedang \cap HPSedikit}$ $= \min(\mu_{JrkSedang(280)};\ \mu_{HPSedikit(540)})$

		$= \min(0.13; 0)$ $= 0$
[R16]	**IF** Jarak **IS** Jauh **AND** HP **IS** Banyak **THEN** TargetPriority **IS** Normal	
	$\alpha - Predikat_{16}$	$= \mu_{JrkJauh \cap HPBanyak}$ $= \min(\mu_{JrkJauh(280)}; \mu_{HPBanyak(540)})$ $= \min(0.86; 0.08)$ $= 0.08$
[R17]	**IF** Jarak **IS** Jauh **AND** HP **IS** Sedang **THEN** TargetPriority **IS** Rendah	
	$\alpha - Predikat_{17}$	$= \mu_{JrkJauh \cap HPSedang}$ $= \min(\mu_{JrkJauh(280)}; \mu_{HPSedang(540)})$ $= \min(0.86; 0.86)$ $= 0.86$
[R18]	**IF** Jarak **IS** Jauh **AND** HP **IS** Sedikit **THEN** TargetPriority **IS** SangatRendah	
	$\alpha - Predikat_{18}$	$= \mu_{JrkJauh \cap HPSedikit}$ $= \min(\mu_{JrkJauh(280)}; \mu_{HPSedikit(540)})$ $= \min(0.86; 0)$ $= 0$

Berikut ini adalah hasil dari proses fungsi implikasi min pada Tabel 3.6.

Tabel 3. 6 Hasil Fungsi Implikasi Untuk Jarak 240, Kecepatan 5 dan HP 540

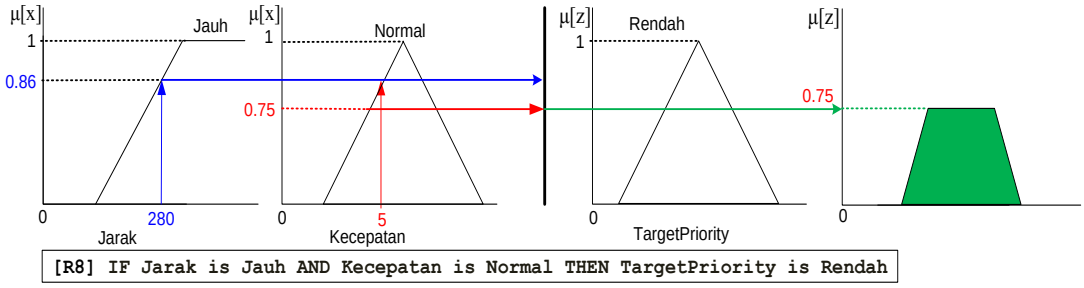
		Jarak		
		Dekat	Sedang	Jauh
Kecepatan	Lambat	[R3] **Normal** (0.00)	[R6] **Rendah** (0.00)	[R9] **Sangat Rendah** (0.00)
	Normal	[R2] **Tinggi** (0.00)	[R5] **Normal** (0.13)	[R8] **Rendah** (0.75)
	Cepat	[R1] **Sangat Tinggi** (0.00)	[R4] **Tinggi** (0.13)	[R7] **Normal** (0.25)
HP	Sedikit	[R12] **Normal** (0.00)	[R15] **Rendah** (0.00)	[R18] **Sangat Rendah** (0.00)
	Sedang	[R11] **Tinggi** (0.00)	[R14] **Normal** (0.13)	[R17] **Rendah** (0.86)
	Banyak	[R10] **Sangat Tinggi** (0.00)	[R13] **Tinggi** (0.08)	[R16] **Normal** (0.08)

Pada Tabel 3.6 nilai dari hasil proses implikasi min untuk himpunan *Fuzzy* rendah adalah (0.75;0.86), himpunan *Fuzzy* normal (0.13;0.25;0.13;0.08) dan himpunan *Fuzzy* tinggi (0.13;0.08). Sedangkan nilai untuk himpunan *Fuzzy* sangat tinggi dan sangat rendah adalah (0.00). Berdasarkan hasil nilai yang diperoleh, didapat aturan- aturan berikut ini :

[R8] **IF** Jarak **IS** Jauh **AND** Kecepatan **IS** Normal **THEN** TargetPriority **IS** Rendah

$$\alpha - Predikat_8 = \mu_{JrkJauh \cap KcpNormal} = \min(\mu_{JrkJauh(280);}\ \mu_{KcpNormal(5)}) = \min(0.86; 0.75) = 0.75$$

Untuk lebih jelasnya bisa dilihat pada Gambar 3.10.

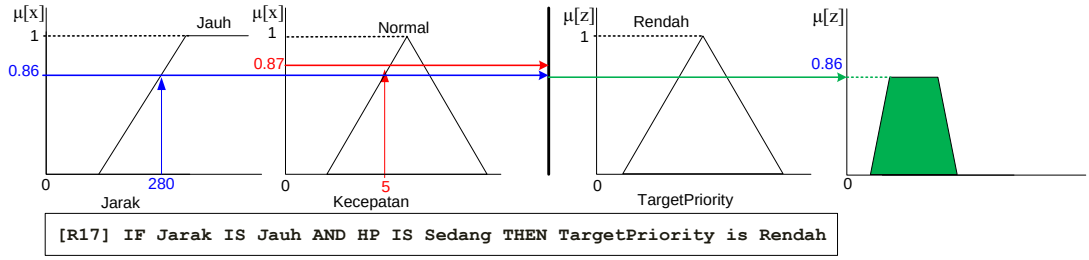


Gambar 3. 10 Aplikasi Fungsi Implikasi R8

[R17] **IF** Jarak **IS** Jauh **AND** HP **IS** Sedang **THEN** TargetPriority **IS** Rendah

$$\alpha - Predikat_{17} = \mu_{JrkJauh \cap HPSedang} = \min(\mu_{JrkJauh(280);}\ \mu_{HPSedang(540)}) = \min(0.86; 0.87) = 0.86$$

Untuk lebih jelasnya bisa dilihat pada Gambar 3.11.

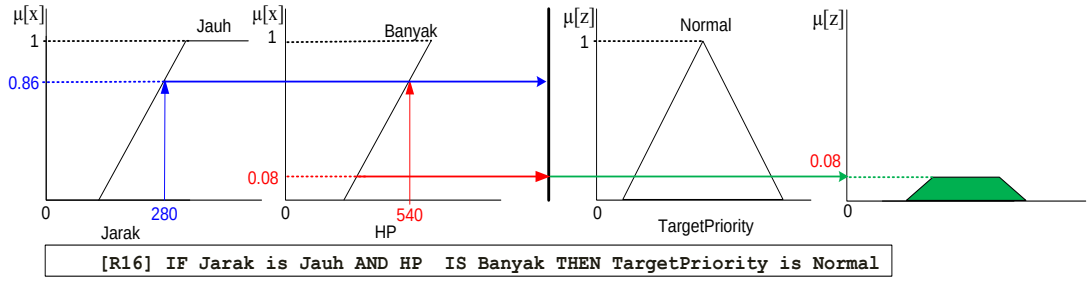


Gambar 3. 11 Aplikasi Fungsi Implikasi R17

[R16] **IF** Jarak **IS** Jauh **AND** HP **IS** Banyak **THEN** TargetPriority **IS** Normal

$$\alpha - Predikat_{16} = \mu_{JrkJauh \cap HPBanyak}$$
$$= \min(\mu_{JrkJauh(280)};\ \mu_{HPBanyak(540)})$$
$$= \min(0.86; 0.08)$$
$$= 0.08$$

Untuk lebih jelasnya bisa dilihat pada Gambar 3.12.

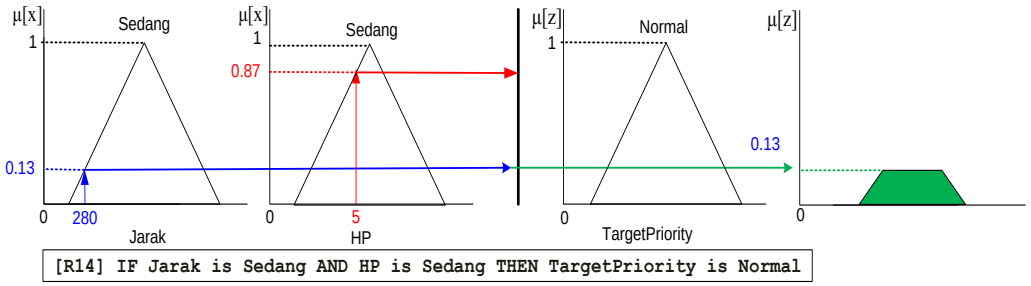


Gambar 3. 12 Aplikasi Fungsi Implikasi R16

[R14] **IF** Jarak **IS** Sedang **AND** HP **IS** Sedang **THEN** TargetPriority **IS** Normal

$$\alpha - Predikat_{14} = \mu_{JrkSedang \cap HPSedang}$$
$$= \min(\mu_{JrkSedang(280)};\ \mu_{HPSedang(540)})$$
$$= \min(0.13; 0.87)$$
$$= 0.13$$

Untuk lebih jelasnya bisa dilihat pada Gambar 3.13.

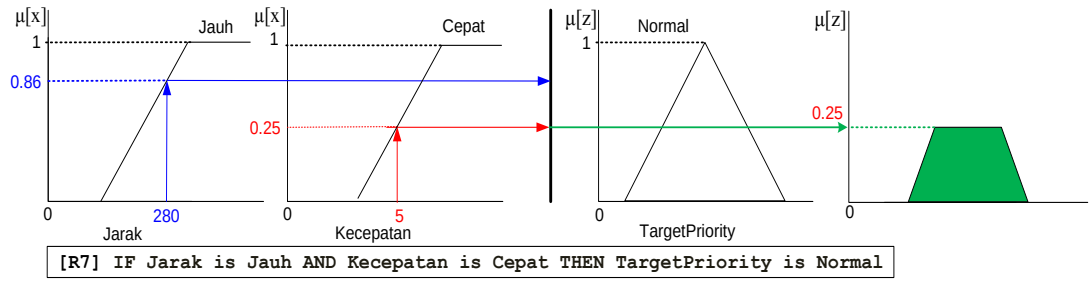


Gambar 3. 13 Aplikasi Fungsi Implikasi R14

[R7] **IF** Jarak **IS** Jauh **AND** Kecepatan **IS** Cepat **THEN** TargetPriority **IS** Normal

$$\alpha - Predikat_7 = \mu_{JrkJauh \cap KcpCepat}$$
$$= \min(\mu_{JrkJauh(280)};\ \mu_{KcpCepat(5)})$$
$$= \min(0.86; 0.25)$$
$$= 0.25$$

Untuk lebih jelasnya bisa dilihat pada Gambar 3.14.

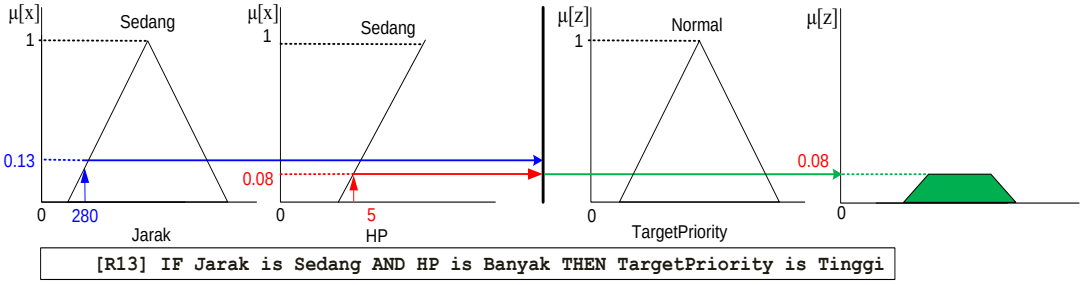


Gambar 3. 14 Aplikasi Fungsi Implikasi R7

[R13] **IF** Jarak **IS** Sedang **AND** HP **IS** Banyak **THEN** TargetPriority **IS** Tinggi

$$\alpha - Predikat_{13} = \mu_{JrkSedang \cap HPBanyak}$$
$$= \min(\mu_{JrkSedang(280)};\ \mu_{HPBanyak(540)})$$
$$= \min(0.13; 0.08)$$
$$= 0.08$$

Untuk lebih jelasnya bisa dilihat pada Gambar 3.15.

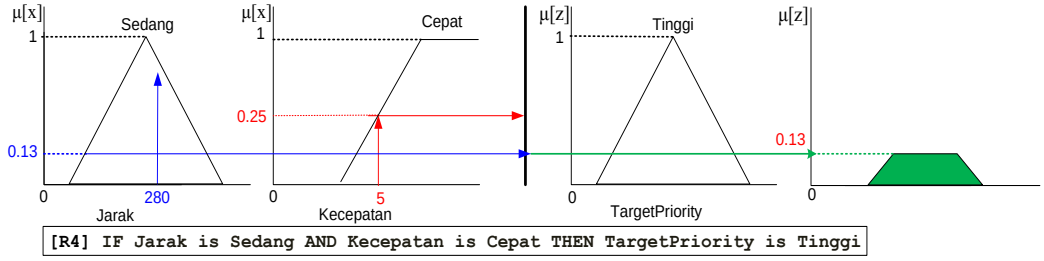


Gambar 3. 15 Aplikasi Fungsi Implikasi R13

[R4] **IF** Jarak **IS** Sedang **AND** Kecepatan **IS** Cepat **THEN** TargetPriority **IS** Tinggi

$$\alpha - Predikat_4 = \mu_{JrkSedang \cap KcpCepat}$$
$$= \min(\mu_{JrkSedang(280)};\ \mu_{KcpCepat(5)})$$
$$= \min(0.13; 0.25)$$
$$= 0.13$$

Untuk lebih jelasnya bisa dilihat pada Gambar 3.16.



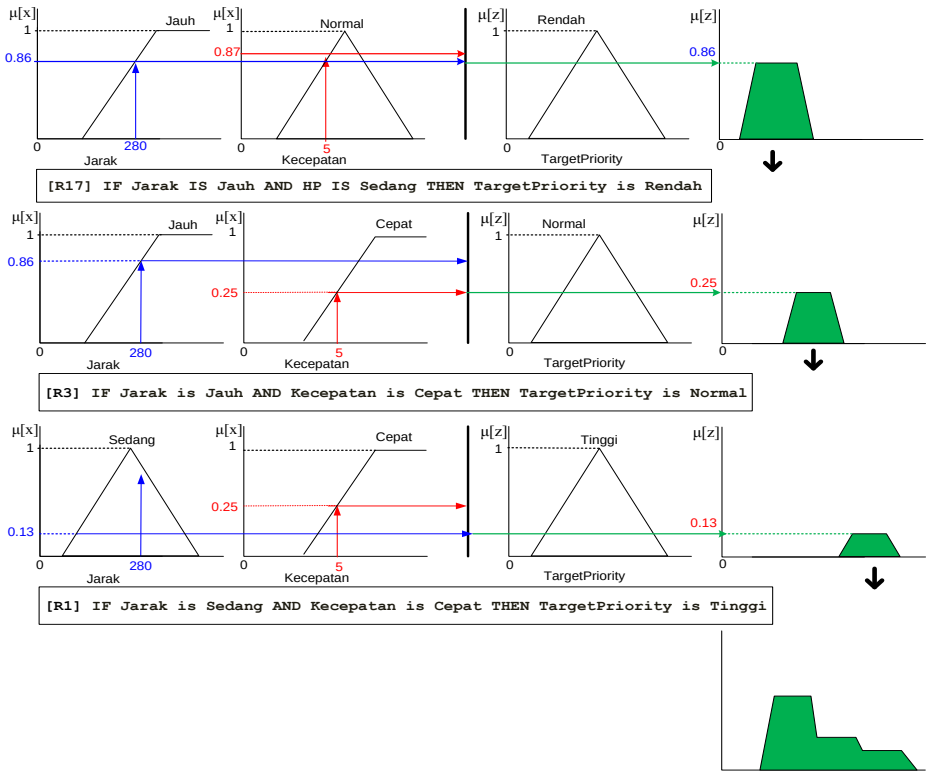
Gambar 3. 16 Aplikasi Fungsi Implikasi R4

3.4.2.3 Komposisi Aturan

Dari hasil aplikasi fungsi implikasi dari tiap aturan, digunakan metode MAX untuk melakukan komposisi antar semua aturan. Berikut ini adalah daerah solusi fuzzy yang didapat dari proses fungsi implikasi.

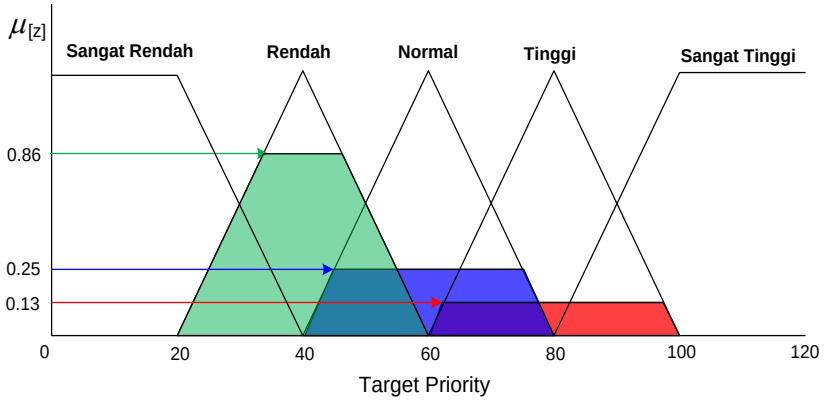
$$\mu_{sf}(z) = \mu_{rendah(z) \cup} \mu_{normal(z) \cup} \mu_{tinggi(z)}$$
$$= maks\{0.86, 0.25, 0.13\}$$

Untuk lebih jelasnya dapat dilihat pada Gambar 3.17.



Gambar 3. 17 Komposisi Aturan *Fuzzy* Menggunakan Metode Max

Setelah di dapat dan menggabungkan dari semua kesimpulan masing – masing aturan. Hasilnya bisa dilihat pada Gambar 3.18.



Gambar 3. 18 Daerah Hasil Komposisi Aturan Metode Max

Proses selanjutnya adalah menentukan fungsi keanggotaan dari himpunan *Fuzzy* rendah, normal dan tinggi.

1. Rendah

   ketika nilai $\mu_{\text{Rendah}(z)= 0.86,}$maka dapat ditentukan nilai z dengan mengacu pada persamaan (3.11) :

   $\frac{z-20}{40-20}$ = 0.86 maka nilai z = 37.2

2. Normal

   Ketika nilai $\mu_{Normal(z)= 0.25,}$maka dapat ditentukan nilai z dengan mengacu pada persamaan (3.12) .

   $\frac{z-40}{60-40}$ = 0.25 maka nilai z = 45 dan $\frac{60-z}{80-60}$ = 0.25 maka nilai z = 55

3. Tinggi

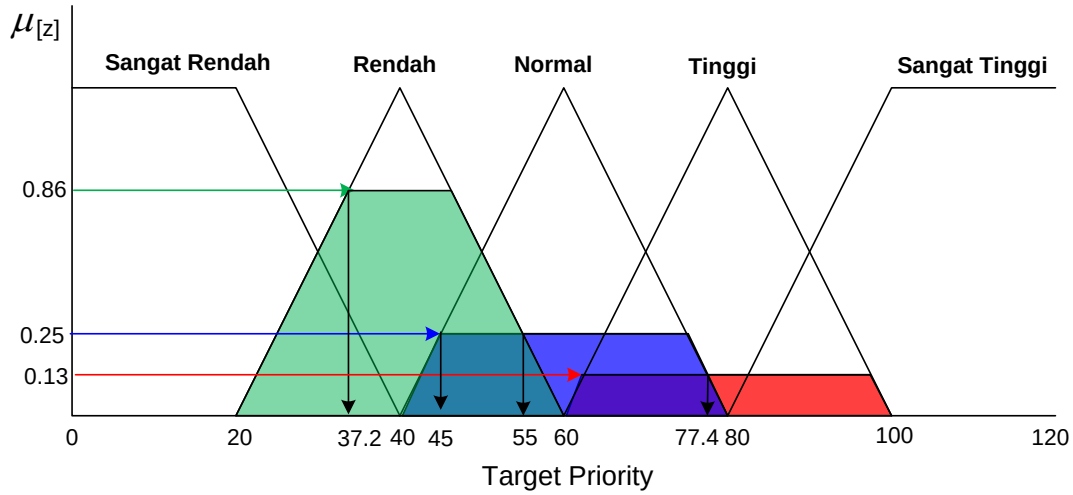
   Ketika nilai $\mu_{Tinggi(z)= 0.13,}$ maka dapat ditentukan nilai z dengan mengacu pada persamaan (3.12).

   $\frac{80-z}{100-80}$ = 0.13 maka nilai z = 77.4

Dengan demikian fungsi keanggotaan hasil komposisi aturan adalah sebagai berikut:

$$\mu_{TargetPriority=}\begin{cases} 0.86; & 20 \le z \le 37.2 \\ \dfrac{z-40}{60-40}; & 37.2 \le z \le 45 \\ 0.25; & 45 \le z \le 55 \\ \dfrac{80-z}{100-80}; & 55 \le z \le 77.4 \\ 0.13; & 77.4 \le z \le 100 \end{cases}$$

Untuk lebih jelasnya dapat dilihat pada Gambar 3.19.



Gambar 3. 19 Daerah Solusi *Fuzzy* Rendah, Tinggi dan Normal

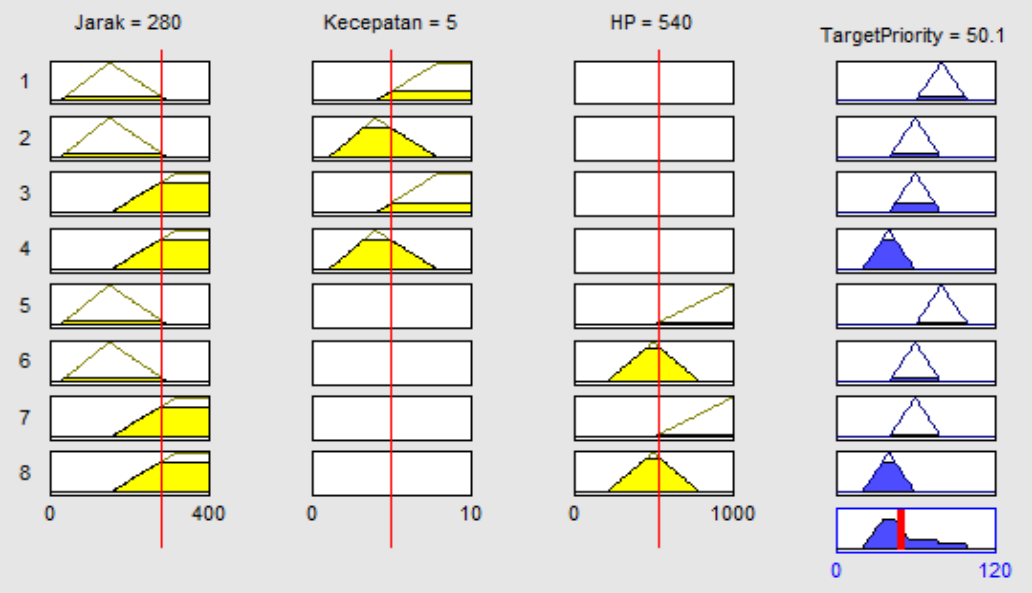
3.4.2.4 Penegasan (De*ffuzy*)

Berikut ini adalah perhitungan deffuzifikasi menggunakan metode *centroid*.

$$Z = \frac{\int_{20}^{37.2} 0.86\ z\ dz + \int_{37.2}^{45} \frac{z-40}{60-40} z\ dz + \int_{45}^{55} 0.25\ z\ dz + \int_{55}^{77.4} \frac{80-z}{100-80} z\ dz + \int_{77.4}^{100} 0.13\ z\ dz}{\int_{20}^{37.2} 0.86\ dz + \int_{37.2}^{45} \frac{z-40}{60-40}\ dz + \int_{45}^{55} 0.25\ dz + \int_{55}^{77.4} \frac{80-z}{100-80} dz + \int_{77.4}^{100} 0.13\ dz}$$

$$= \frac{1804.61}{36.11} = 49.97$$

Jadi nilai yang dihasilkan berdasarkan input nilai jarak [280] , kecepatan [5] dan HP [540] adalah 49.97. Perhitungan berdasarkan program simulasi *Matlab* dengan menggunakan fasilitas *Fuzzy Logic Toolbox* ditunjukan pada Gambar 3.20.



Gambar 3. 20 Perhitungan Menggunakan *Fuzzy Logic Toolbox Matlab*

Dari gambar diatas, garis vertikal merah tebal pada variabel *output TargetPriority* menunjukkan nilai target musuh yaitu sebesar 50.1. Nilai target tersebut termasuk ke dalam keanggotaan himpunan *Fuzzy TargetPriority* Rendah, sehingga jarak sebesar 280, kecepatan 5 dan Hp sebesear 540 termasuk ke dalam *TargetPriority* Rendah.

3.5 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen-komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukan yang diperlukan sistem, keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk mengolah masukan sehingga menghasilkan suatu keluaran yang diinginkan. Pada analisis kebutuhan sistem non fungsional ini dijelaskan analisis kebutuhan perangkat keras dan analisis kebutuhan perangkat lunak.

3.5.1 Analisis Kebutuhan Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan dalam mengimplementasikan algoritma pada penelitian ini adalah sebagai berikut seperti ditunjukkan pada Tabel 3.7.

Tabel 3. 7 Analisis Kebutuhan Perangkat Keras

Nama Perangkat	Spesifikasi
Processor	Core 2 duo 2.80 GHz
Memory	DDR3 4 GB
HDD	650 GB
Display	Resolusi 1366 x 768
Monitor	1 Unit
Mouse	1 Unit
Keyboard	1 Unit

3.5.2 Analisis Kebutuhan Perangkat Lunak (*Software*)

Perangkat lunak yang dipergunakan dalam mengimplementasikan algoritma pada penelitian ini adalah sebagai berikut:

Tabel 3. 8 Analisis Kebutuhan Perangkat Lunak

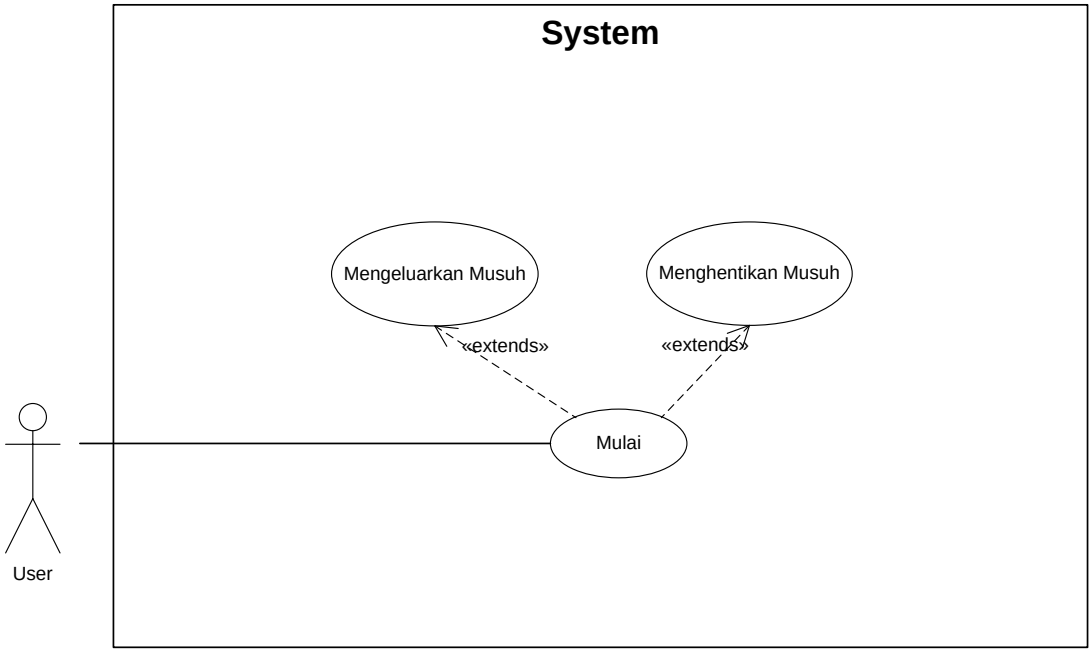
Nama Perangkat Lunak	Jenis
Sistem Operasi	Windows 7
Adobe Flash	CS 5

3.6 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*). Dimana tahap-tahap analisis yang dilakukan dalam mengimplementasikan algoritma pada penelitian ini antara lain *Use Case Diagram, Activity Diagram, Sequence Diagram* dan *Class Diagram*.

3.6.1 *Use Case Diagram*

Use case Diagram merupakan gambaran fungsional dari interaksi antara aktor dengan sistem. Sebuah *Diagram use case* menggambarkan keterhubungan antara aktor dengan aksi yang dapat dilakukannya terhadap sistem. Berikut adalah perancangan *use case* pada penelitian ini dapat dilihat pada Gambar 3.21.



Gambar 3. 21 *Use Case Diagram*

3.6.2 Definisi *Use Case*

Definisi *use case* berfungsi untuk menjelaskan proses yang terdapat pada setiap *use case*.

Tabel 3. 9 Definisi *Use Case*

No	*Use Case*	Deskripsi
1	Mulai	Merupakan proses untuk menampilkan pergerakan unit musuh
2	Mengeluarkan Musuh	Merupakan proses untuk mengeluarkan musuh
3	Menghentikan *Musuh*	Merupakan proses untuk menghentikan musuh dengan cara menempatkan *tower*

3.6.3 Skenario *Use Case*

Skenario adalah bagian pada *use case* menunjukkan proses apa yang terjadi pada setiap *use case* tersebut, dimana aktor memberikan perintah pada setiap bagian *use case*, dan respon apa yang diberikan oleh sistem kepada aktor setelah aktor memberikan perintah pada setiap bagian-bagian *use case*. Adapun skenario *use case* pada penelitian ini adalah sebagai berikut.

1. Skenario *Use Case* Mulai

Skenario *use case* mulai menjelaskan interaksi antara *user* dengan system dan akan dijelaskan pada Tabel 3.10.

Tabel 3. 10 Skenario *Use Case* Mulai

Identifikasi	
Nama	Mulai
Tujuan	Menampilkan Pergerakan Unit Musuh
Aktor	*User*
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih tombol *start*	
	2. Set jalur musuh
	3. Menampilkan pergerakan musuh
	4. Menampilkan Proses Fuzzy
5. Memilih tombol *restart*	
	6. Set Kondisi permainan atau simulasi ke kondisi awal
Kondisi Akhir	Menampilkan Pergerakan Unit Musuh

2. Skenario *Use Case* Mengeluarkan Musuh

Skenario mengeluarkan musuh menjelaskan interaksi antara *user* dengan sistem dan akan dijelaskan pada Tabel 3.11.

Tabel 3. 11 *Use Case* Mengeluarkan Musuh

Identifikasi	
Nama	Mengeluarkan Musuh
Tujuan	Mengeluarkan atau memanggil musuh ke arena permainan
Aktor	*User*
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Mengisi *health point* musuh	
2. Mengisi kecepatan musuh	
3. Memilih tombol reset	
	4. Menghapus nilai hp dan kecepatan pada *textfield*
5. Memilih tombol submit	
	6. Set nilai hp dan kecepatan musuh
7. Memilih tombol *wave*	
	8. Menampilkan musuh di arena permainan
Kondisi Akhir	Menampilkan unit musuh di arena permainan

3. Skenario *Use Case* Menghentikan Musuh

Skenario use case menghentikan musuh menjelaskan interaksi antara *user* dengan sistem dan akan dijelaskan pada Tabel 3.12.

Tabel 3. 12 Skenario *Use Case* Menghentikan musuh

Identifikasi	
Nama	Menghentikan pergerakan musuh
Tujuan	Menghentikan pergerakan musuh menuju *goal point*
Aktor	Sistem
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**

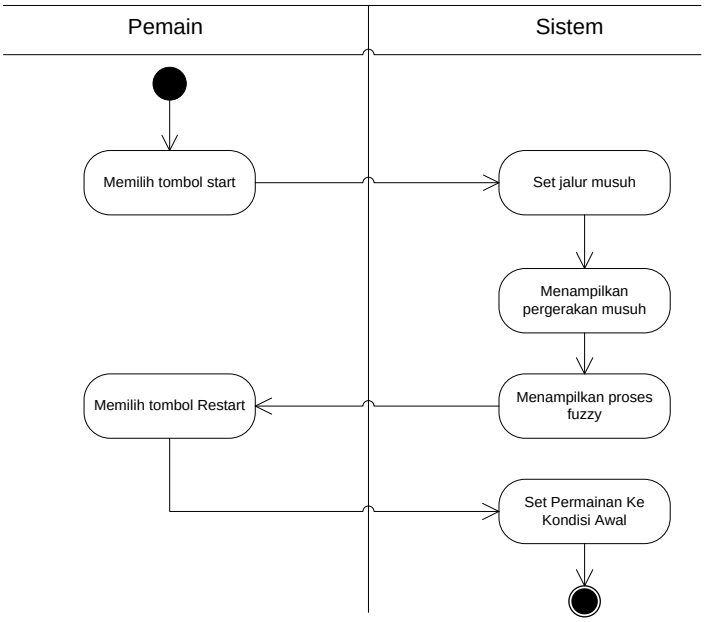
1. Menempatkan *tower* di arena permainan	
	2. Menampilkan *tower* di arena permainan
	3. Memeriksa jarak musuh sesuai jarak tembak *tower*
	4. Inisialisasi variabel Fuzzy (jarak, kecepatan dan hp)
	5. Melakukan proses fuzzyfikasi
	6. Menembak musuh
Kondisi Akhir	Menampilkan kondisi *tower* menyerang musuh

3.6.4 *Activity Diagram*

Activity Diagram menggambarkan berbagai alir aktifitas dalam sistem yang sedang di rancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi dan bagaimana alir berakhir.

1. *Activity Diagram* Mulai

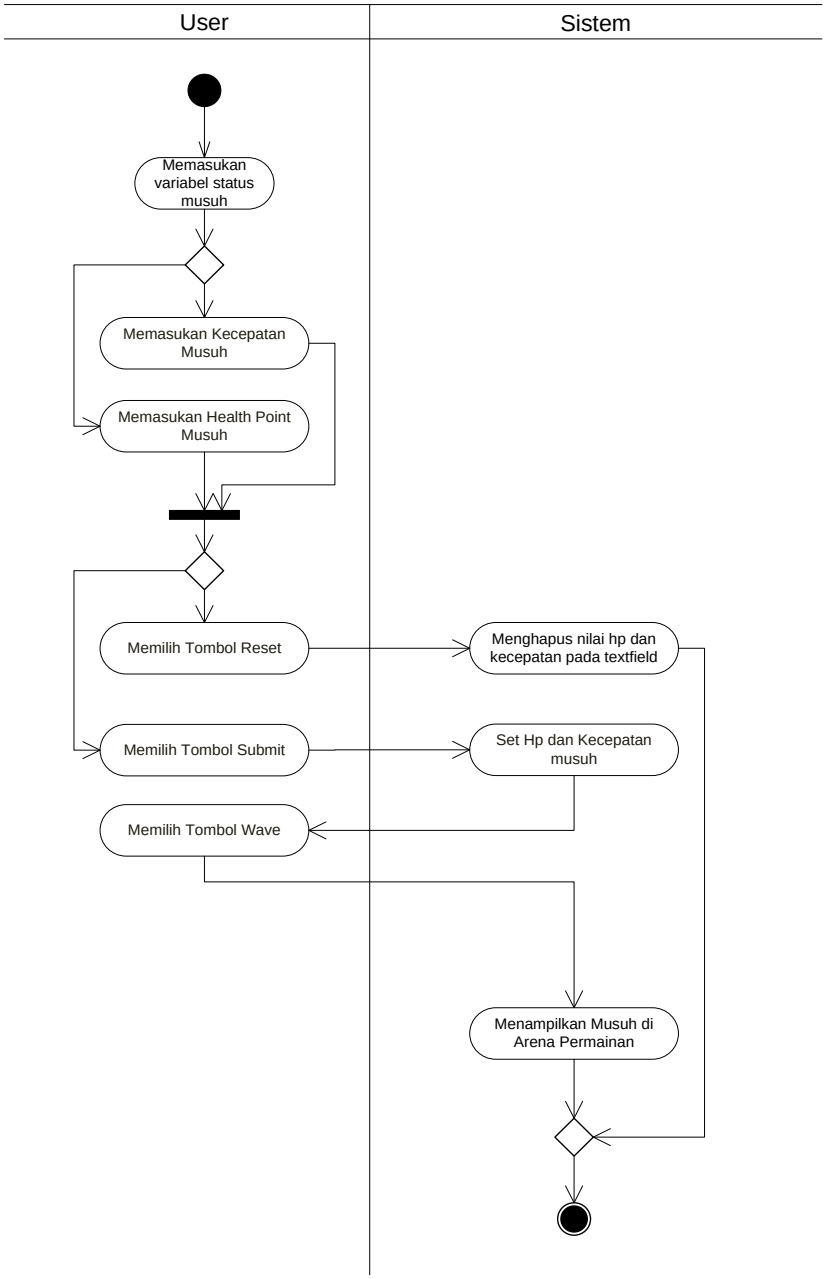
Activity Diagram ini menjelaskan tentang proses memulai permainan atau simulasi pada sistem yang akan dibangun.



Gambar 3. 22 *Activity Diagram* Mulai

2. *Activity Diagram* Mengeluarkan Musuh

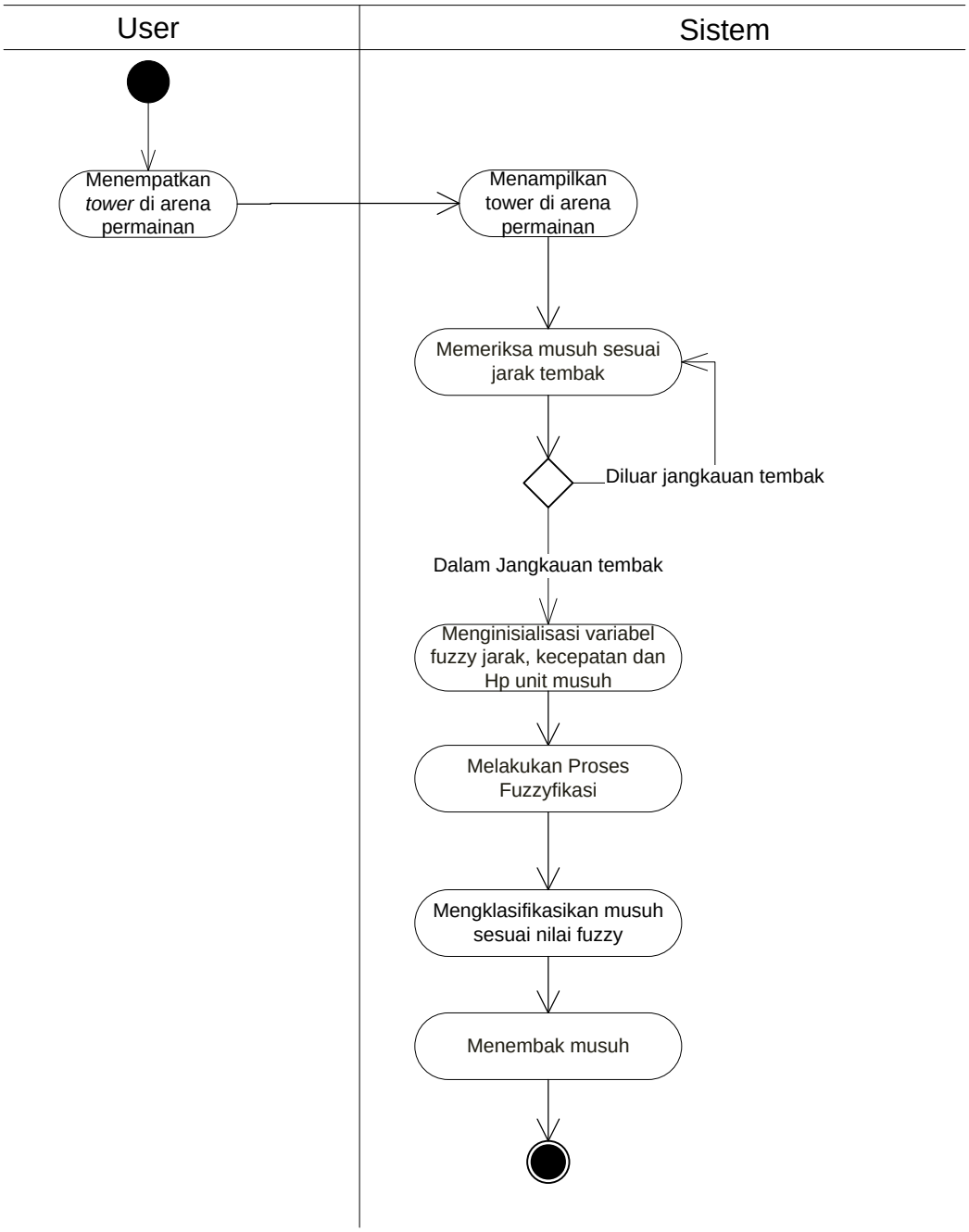
Activity Diagram ini menjelaskan tentang proses mengeluarkan unit musuh. Untuk lebih jelasnya dapat dilihat pada Gambar 3.23.



Gambar 3. 23 *Activity Diagram* Mengeluarkan Musuh

3. *Activity Diagram* Menempatkan *Tower*

Activity Diagram ini menjelaskan tentang proses menempatkan *tower*. Untuk lebih jelasnya dapat dilihat pada Gambar 3.24.



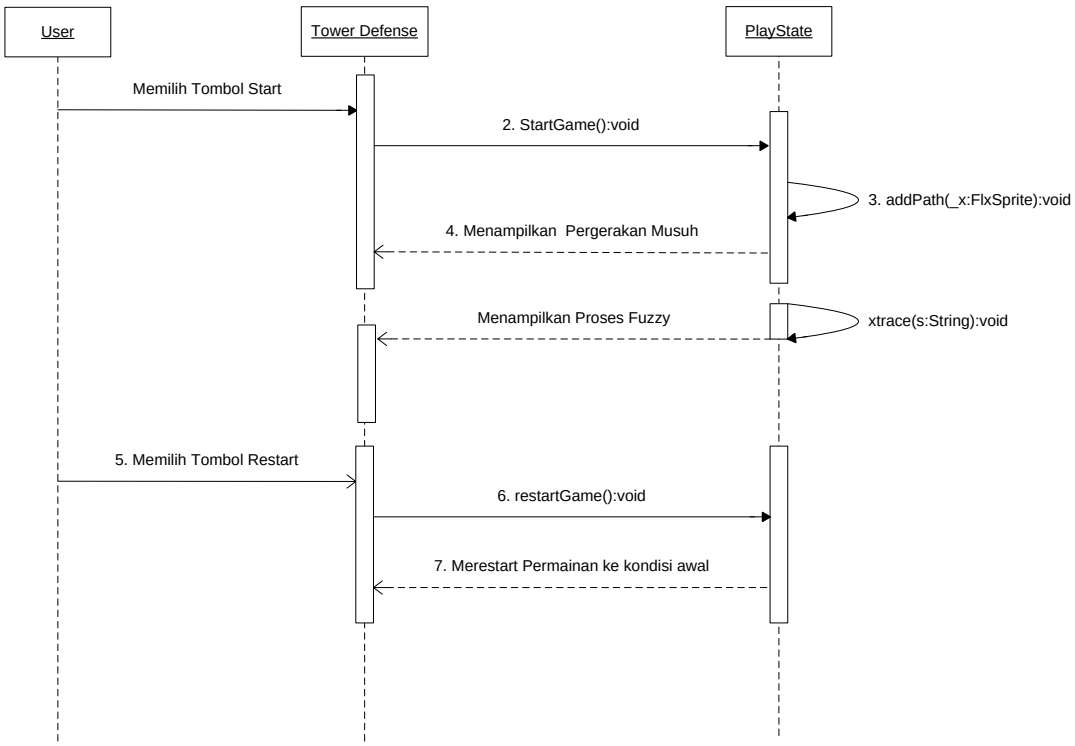
Gambar 3. 24 *Activity Diagram* Menempatkan *Tower*

3.6.5 *Sequence Diagram*

Pada penelitian ini, *sequence Diagram* digunakan untuk menggambarkan serangkaian langkah-langkah yang dilakukan *user* atau pengguna terhadap sistem dan sebaliknya.

1. *Sequence Diagram* Mulai

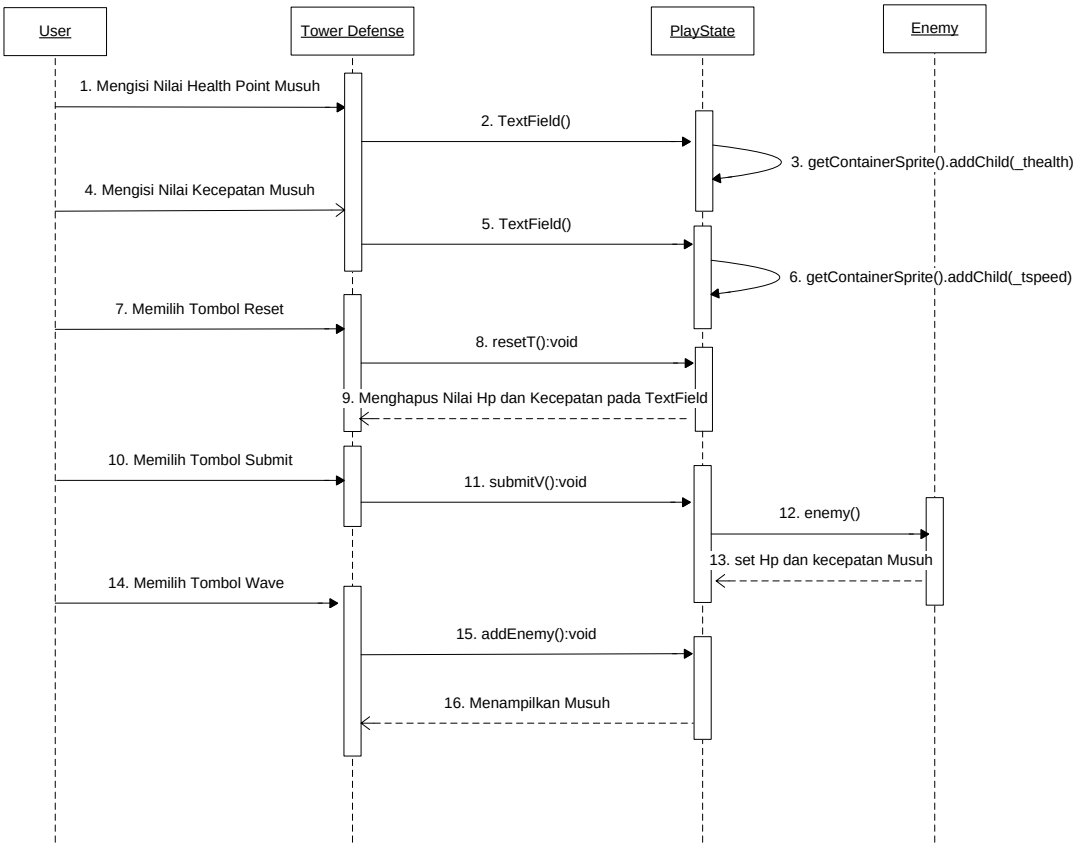
Sequence Diagram mulai menampilkan proses komunikasi pemain dan *method* pada kelas-kelas yang berhubungan dengan proses memulai permainan. Untuk lebih jelasnya bisa dilihat pada Gambar 3.25.



Gambar 3. 25 *Sequence Diagram* Mulai

2. *Sequence Diagram* Mengeluarkan Musuh

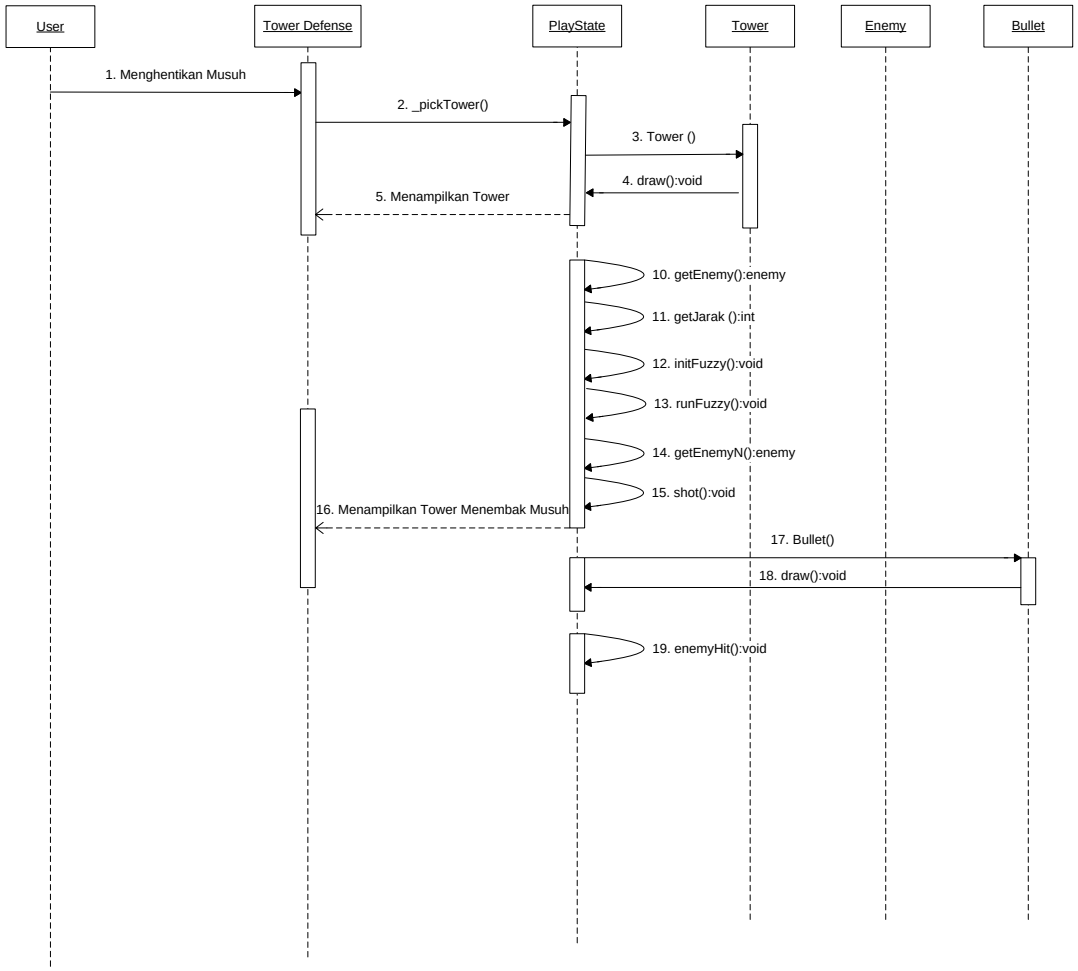
Sequence Diagram menghentikan musuh menampilkan proses komunikasi pemain dan *method* pada kelas-kelas yang berhubungan dengan proses mengeluarkan musuh. Untuk lebih jelasnya bisa dilihat pada Gambar 3.26.



Gambar 3. 26 *Sequence Diagram* Mengeluarkan Musuh

3. *Sequence Diagram* Menghentikan Musuh

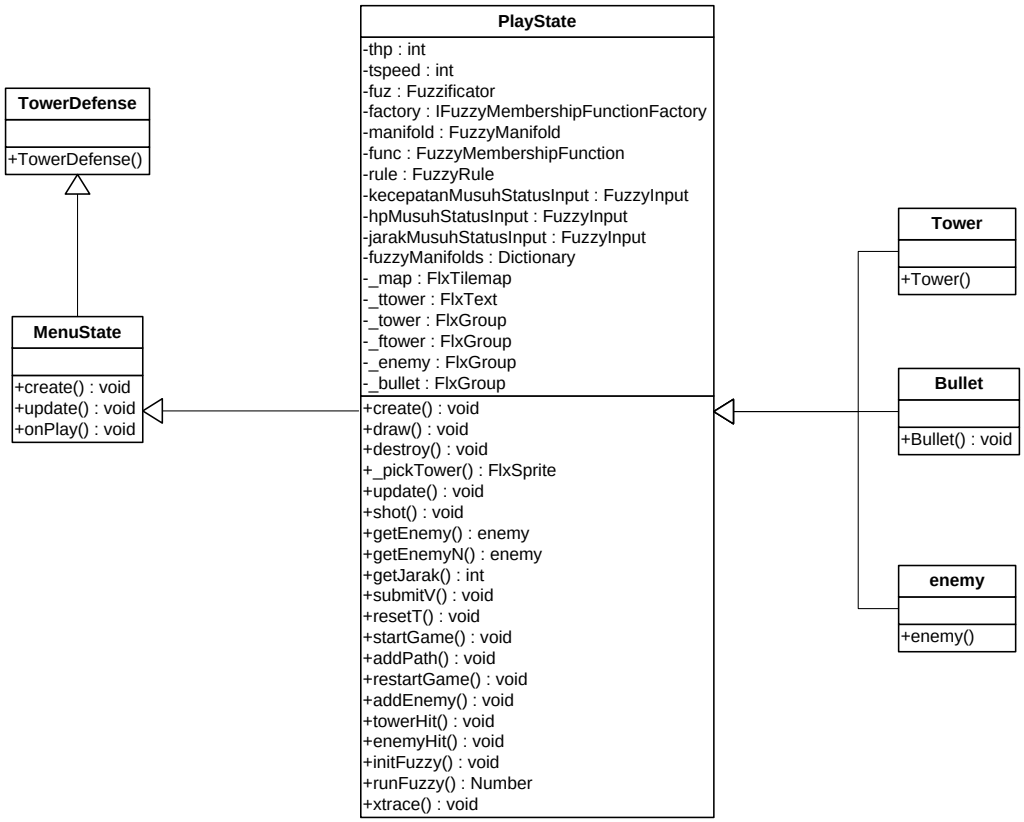
Sequence Diagram menghentikan musuh menampilkan proses komunikasi pemain dan *method* pada kelas-kelas yang berhubungan dengan proses menghentikan musuh. Untuk lebih jelasnya bisa dilihat pada Gambar 3.27.



Gambar 3. 27 *Sequence Diagram* Menghentikan Musuh

3.6.6 *Class Diagram*

Class *Diagram* adalah suatu *Diagram* yang menggambarkan struktur dari suatu sistem. Sistem tersebut akan menampilkan sistem kelas, atribut, dan hubungan antar kelas. Berikut ini adalah perancangan struktur sistem pada penelitian ini, yang digambarkan dengan class *Diagram* yang dapat dilihat pada Gambar 3.28.



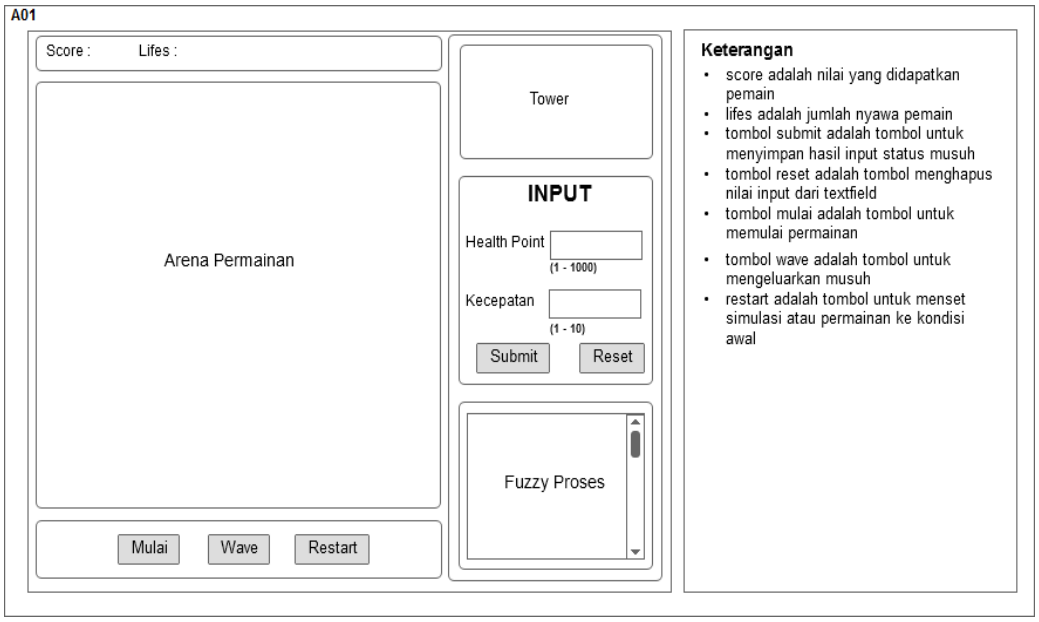
Gambar 3. 28 *Class Diagram*

3.7 Perancangan Sistem

Perancangan sistem adalah proses merancang suatu sistem yang isinya adalah langkah-langkah operasi dalam proses pengolahan data dan proses prosedur-prosedur untuk mendukung operasi sistem pada aplikasi yang akan dibuat.

3.7.1 Perancangan Antar Muka

Perancangan antar muka dibutuhkan untuk mewakili keadaan sebenarnya dari sebuah sistem yang akan dibangun. Perancangan antar muka pada penelitian ini bisa dilihat pada Gambar 3.29.



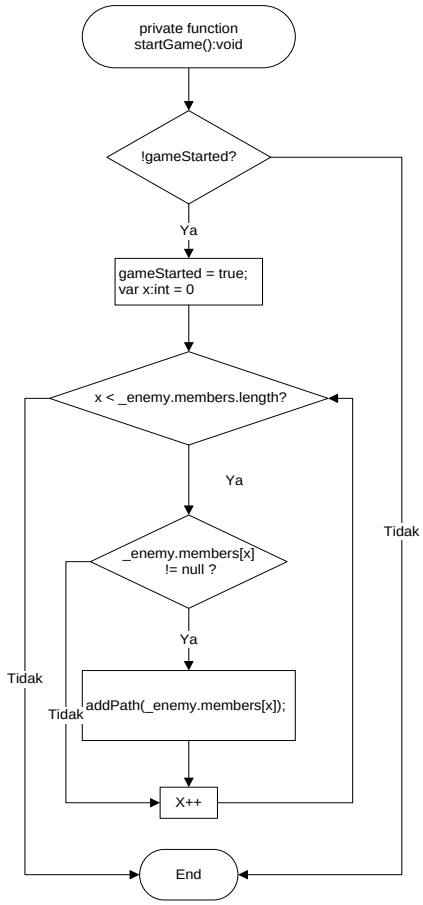
Gambar 3. 29 Perancangan Antar Muka

3.7.2 Perancangan *Method*

Perancangan *method* merupakan perancangan yang berfungsi untuk mendeskripskan *method-method* yang berada di dalam aplikasi. Adapun *method-method* yang digunakan pada penelitian ini adalah sebagai berikut :

3.7.2.1 Perancangan *Method startGame*

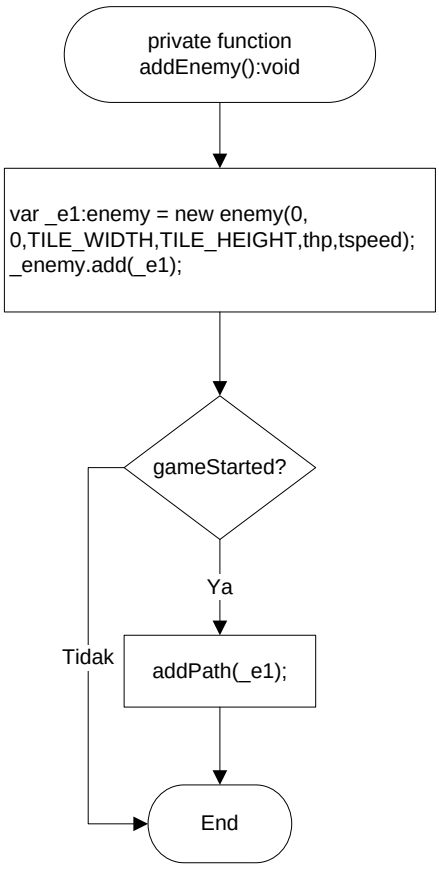
Method startGame merupakan *method* untuk memulai simulasi atau permainan, jika sebelumnya *user* telah menambah unit musuh di arena permainan. Untuk lebih jelasnya dapat dilihat pada Gambar 3.30.



Gambar 3. 30 Perancangan *Method startGame*

3.7.2.2 Perancangan *Method addEnemy*

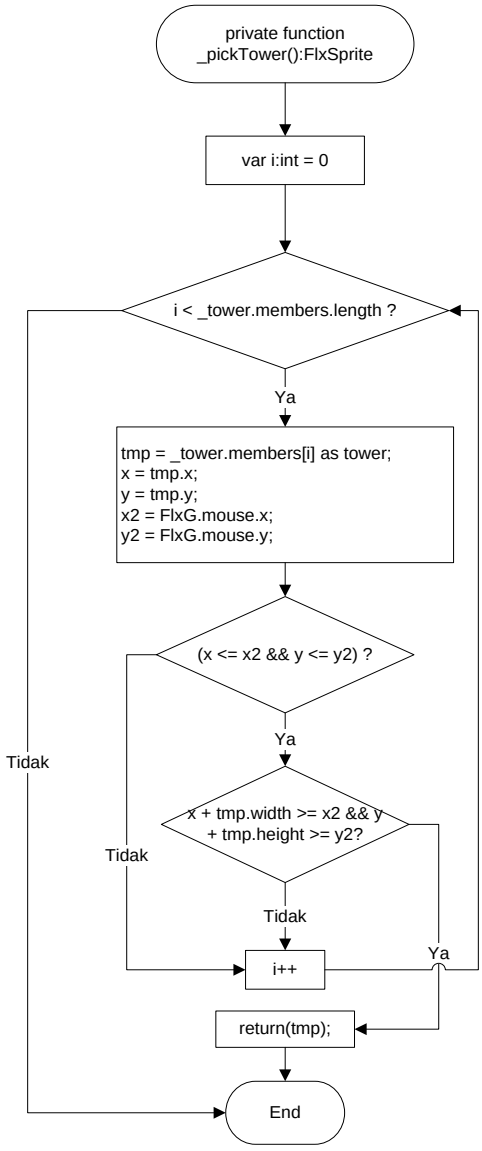
Method addEnemy merupakan *method* untuk menampilkan unit musuh di arena permainan. Untuk lebih jelasnya dapat dilihat pada Gambar 3.31.



Gambar 3. 31 Perancangan *Method addEnemy*

3.7.2.3 Perancangan *Method _pickTower*

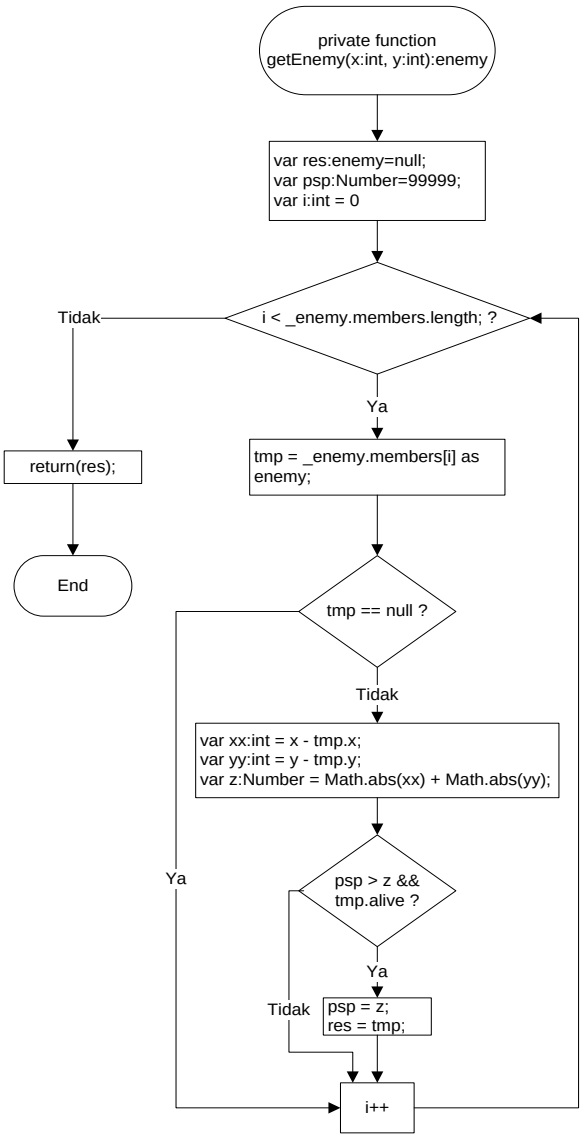
Method _pickTower merupakan *method* untuk menempatkan *tower* di arena permainan. Untuk lebih jelasnya dapat dilihat pada Gambar 3.32.



Gambar 3. 32 Perancangan *Method _pickTower*

3.7.2.4 Perancangan *Method getEnemy*

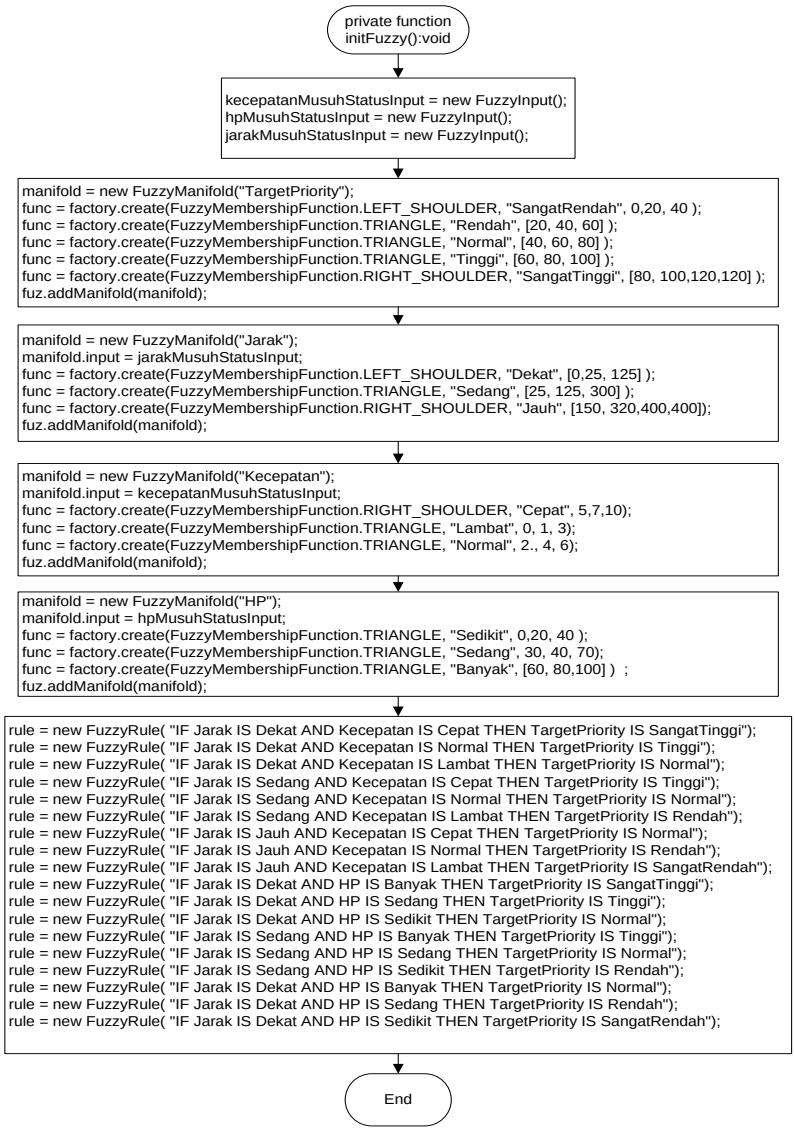
Method getEnemy merupakan *method* yang digunakan untuk mengidentifikasi unit musuh berdasarkan jarak dengan *tower*. Untuk lebih jelasnya dapat dilihat pada Gambar 3.33.



Gambar 3. 33 Perancangan *Method getEnemy*

3.7.2.5 Perancangan *Method initFuzzy*

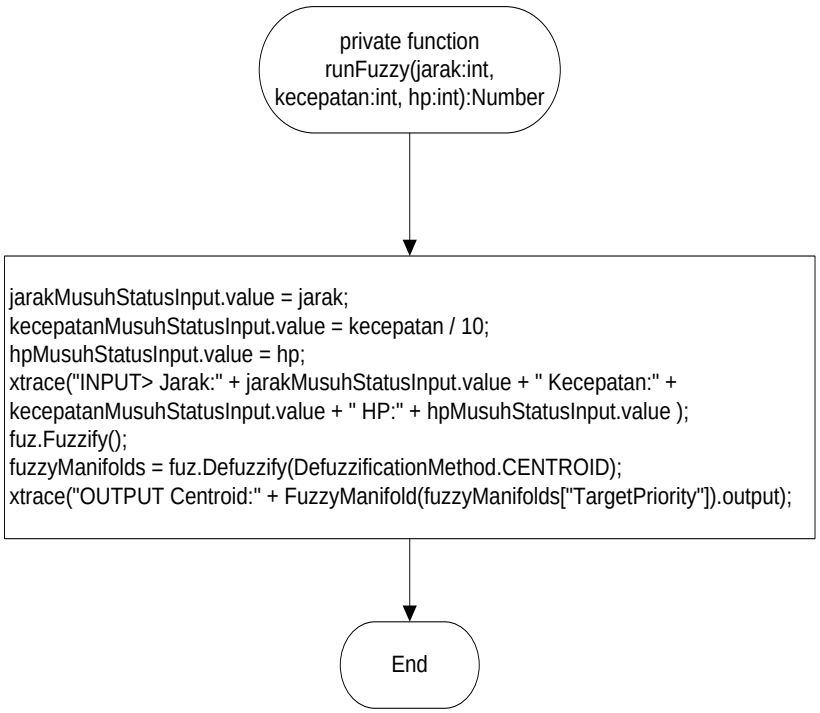
Method runFuzzy merupakan *method* yang digunakan untuk melakukan proses inisialisasi variabel masukan *Fuzzy*, pembentukan fungsi keanggotaan dan aturan *Fuzzy* setelah sebelumnya *tower* mengidentifikasi jarak, kecepatan dan hp pada unit musuh. Untuk lebih jelasnya dapat dilihat pada Gambar 3.34.



Gambar 3. 34 Perancangan *Method initFuzzy*

3.7.2.6 Perancangan *Method runFuzzy*

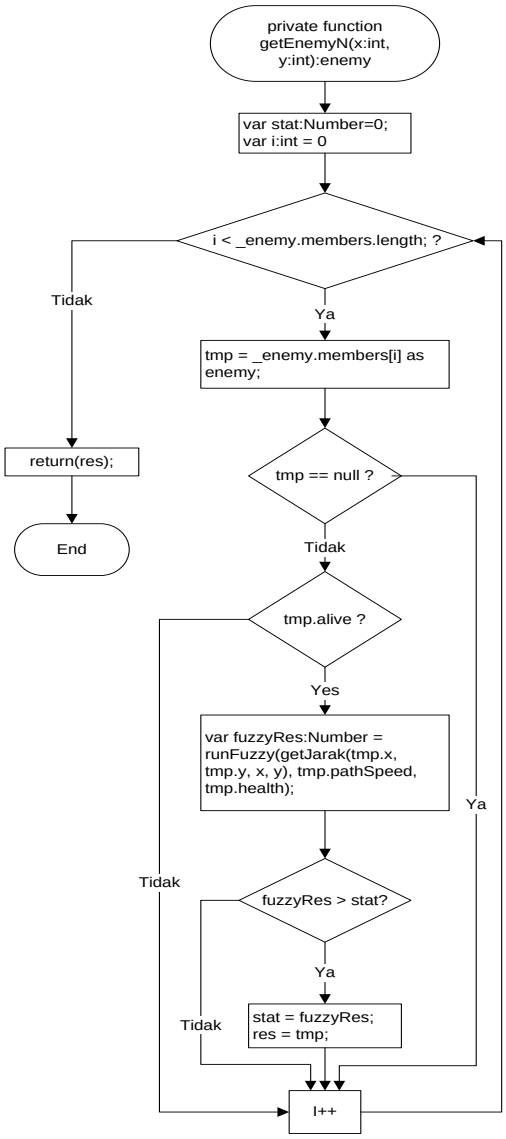
Method runFuzzy merupakan *method* yang digunakan untuk melakukan proses fuzzyfikasi setelah sebelumnya telah menginisialisasi variabel masukan *fuzzy*. Untuk lebih jelasnya dapat dilihat pada Gambar 3.35.



Gambar 3. 35 Perancangan *Method runFuzzy*

3.7.2.7 Perancangan *Method getEnemyN*

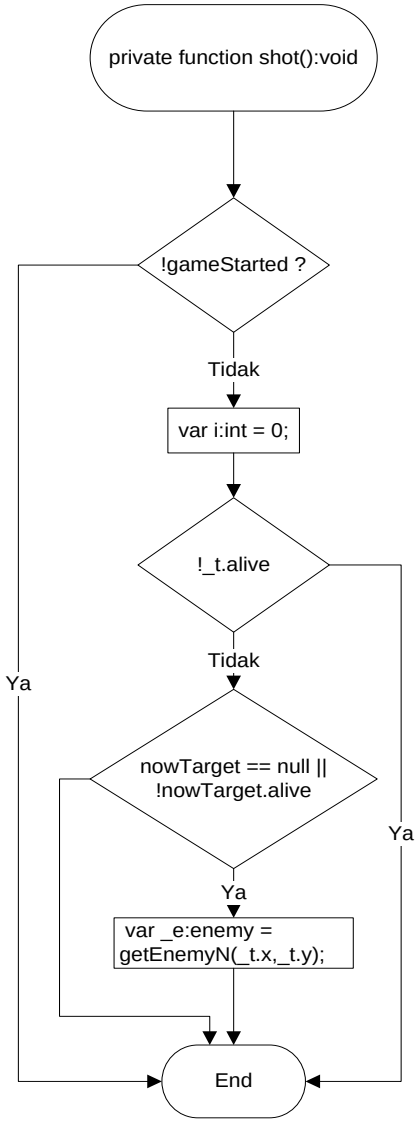
Method getEnemyN merupakan *method* yang digunakan untuk mengidentifikasi nilai target *priority* dari tiap unit musuh berdasarkan hasil dari perhitungan fuzzyfikasi. Untuk lebih jelasnya dapat dilihat pada Gambar 3.36.



Gambar 3. 36 Perancangan *Method getEnemyN*

3.7.2.8 Perancangan *Method shot*

Method shot merupakan *method* untuk menembak musuh sesuai nilai target *priority* hasil dari proses fuzzyfikasi. Untuk lebih jelasnya dapat dilihat pada Gambar 3.37.



Gambar 3. 37 Perancangan *Method Shot*