

**PENERAPAN LOGIKA FUZZY PADA TOWER DI  
GAME TOWER DEFENCE**

**SKRIPSI**



**Disusun Oleh :  
MUHAMMAD NUR HADI  
12.18.213**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG  
2016**

---

**LEMBAR PERSETUJUAN**

**PENERAPAN LOGIKA FUZZY PADA TOWER DI  
GAME TOWER DEFENCE**

**SKRIPSI**

*Disusun dan Diajukan untuk melengkapi dan memenuhi persyaratan guna  
mencapai Gelar Sarjana Komputer Strata Satu (S-1)*

Disusun Oleh :

**MUHAMMAD NUR HADI**

**NIM : 12.18.213**

**Diperiksa dan Disetujui**

**Dosen Pembimbing I**

**Dosen Pembimbing II**

**Joseph Dedy Irawan, ST, MT**  
**NIP. 197404162005011002**

**Sonny Prasetyo, ST, MT**  
**NIP.P. 1031000433**

**Mengetahui**

**Kema Program Studi Teknik Informatika S-1**

**Joseph Dedy Irawan, ST, MT**  
**NIP. 197404162005011002**

**PROGRAM STUDI TEKNIK INFORMATIKA S-1  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG**

**2016**

## LEMBAR KEASLIAN

### PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan dibawah ini :

Nama : MUHAMMAD NUR HADI

NIM : 12.18.213

Program Studi : Teknik Informatika S-1

Fakultas : Fakultas Teknologi Industri

Menyatakan dengan sesungguhnya bahwa Skripsi saya yang berjudul :

**“Penerapan Logika Fuzzy Pada Tower Di *Game Tower Defence*”**

Adalah skripsi sendiri bukan duplikasi serta mengutip atau menyadur seluruhnya karya orang lain kecuali dari sumber aslinya.

Malang, 8 Februari 2016

Yang membuat pernyataan



Muhammad Nur Hadi

## Abstrak

*Tower Defence salah satu sub genre dari game bergenre RTS (Real Time Strategy). Pada game bergenre ini, player harus menahan atau melindungi daerahnya dari serangan musuh. Tower merupakan NPC yang harus dilindungi dan sekaligus dapat menyerang musuh. Namun game pada umumnya, Tower tidak memiliki kemampuan untuk membedakan target prioritas musuh yang tingkat bahayanya tinggi dan tingkat bahaya yang rendah.*

*Penulis membuat game Tower Defence yang towernya memiliki kecerdasan buatan menggunakan logika fuzzy mamdani. Library untuk logika fuzzy menggunakan library DotFuzzy. Game Engine yang digunakan adalah Unity.*

*Berdasarkan hasil pengujian, tower yang menggunakan fuzzy mamdani dan tower yang tidak menggunakan fuzzy mamdani dapat membunuh creep dengan jumlah yang sama, yaitu 3 creep. Tower yang menggunakan fuzzy mamdani, 5,7% dapat bertahan hidup lebih lama daripada tower tanpa fuzzy mamdani. Tower yang menggunakan fuzzy akan menembak berdasarkan target priority sedangkan tower yang tidak menggunakan fuzzy akan menembak berdasarkan urutan creep masuk ke area tembak tower.*

**Kata kunci:** *Tower defence, kecerdasan buatan, logika fuzzy, mamdani, unity*

## KATA PENGANTAR

Puji dan Syukur penyusun panjatkan kehadirat Tuhan Yang Maha Esa, karena telah memberikan rahmat-Nya sehingga penyusun dapat menyelesaikan skripsi dengan judul **PENERAPAN LOGIKA FUZZY PADA TOWER DI GAME TOWER DEFENCE** sesuai dengan waktu yang ditentukan.

Penyusunan skripsi ini merupakan salah satu persyaratan untuk menyelesaikan program pendidikan Strata Satu (S-1) Teknik Informatika, Fakultas Teknologi Industri di Institut Teknologi Nasional Malang. Penyusun mengucapkan terima kasih kepada:

1. Allah SWT yang selalu menguatkan dan meridhoi.
2. Bapak Muhammad Nuruddin dan Ibu Rasinah selaku orang tua yang selalu mendukung dalam hal material dan immaterial .
3. Dr. Ir. Lalu Mulyadi, MT selaku Rektor ITN Malang.
4. Ir. Anang Subardi, MT selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
5. Joseph Dedy Irawan, ST, MT selaku Ketua Jurusan Teknik Informatika Institut Teknologi Nasional Malang.
6. Sonny Prasetyo, ST, MT selaku Sekertaris Jurusan Teknik Informatika Institut Teknologi Nasional Malang.
7. Michael Ardita, ST, MT selaku dosen wali.
8. Bapak Joseph Dedy Irawan, ST, MT selaku pembimbing I dan Bapak Sonny Prasetyo, ST, MT, selaku pembimbing II yang telah memberikan bimbingan dan arahan sehingga skripsi ini dapat diselesaikan.
9. Semua sahabat yang telah membantu penulisan skripsi ini dengan masukan – masukan yang positif.

Penulis menyadari bahwa skripsi ini masih belum sempurna, oleh karena itu penulis mengharapkan kritik dan saran untuk kesempurnaan skripsi ini.

Malang, 8 Februari 2016

(Muhammad Nur Hadi)

## DAFTAR ISI

ABSTRAK .....	iv
KATA PENGANTAR .....	v
DAFTAR ISI .....	vi
DAFTAR GAMBAR .....	viii
DAFTAR TABEL .....	x
BAB I PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	2
1.3. Batasan Masalah .....	2
1.4. Tujuan .....	3
1.5. Metodologi Penelitian .....	3
1.6. Sistematika Penulisan .....	4
BAB II LANDASAN TEORI .....	5
2.1 Tower Defence .....	5
2.2 Logika Fuzzy .....	6
2.2.1 Fungsi Keanggotaan .....	6
2.2.2 Metode Mamdani .....	9
2.3 Library DotFuzzy .....	12
2.4 Unity .....	12
2.4.1 Interface Unity .....	13
2.4.2 IDE Visual Studio C# Integration .....	15
2.4.3 Script .....	15
2.5 Matlab Fuzzy Logic Toolbox .....	16
BAB III ANALISIS DAN PERANCANGAN .....	17
3.1 Blok Diagram Menu .....	17
3.2 Perancangan <i>Scene</i> .....	17
3.2.1 <i>Scene</i> Menu .....	17
3.2.2 <i>Scene</i> Mulai .....	17
3.2.3 <i>Scene</i> Nilai .....	18
3.2.4 <i>Scene</i> Tentang .....	19
3.2.5 Pause dan Cheat .....	19
3.3 Perancangan <i>Gameplay</i> .....	20
3.3.1 <i>Hero</i> .....	20
3.3.2 Tower .....	20
3.3.3 <i>Creep</i> .....	20
3.3.4 Hole Spawn .....	21
3.3.5 <i>Item</i> .....	21

3.3.6 Perhitungan <i>Score</i> .....	21
3.4 Atribut <i>NPC</i> dan <i>Player</i> .....	21
3.4.1 <i>Player Hero</i> .....	21
3.4.2. <i>NPC Tower</i> .....	22
3.4.3. <i>NPC Creep</i> .....	23
3.4.4. <i>Item</i> .....	24
3.5 Analisis Himpunan Fuzzy.....	25
3.6 Rule Fuzzy.....	30
3.7 Analisis Menggunakan <i>Fuzzy Toolbox Matlab</i> .....	31
BAB IV IMPLEMENTASI DAN PENGUJIAN.....	32
4.1 Implementasi.....	32
4.1.1 Implementasi <i>Scene Menu</i> .....	32
4.1.2 Implementasi <i>Scene Mulai</i> .....	32
4.1.3 Implementasi <i>Scene Nilai</i> .....	34
4.1.4 Implementasi <i>Scene Tentang</i> .....	35
4.1.5 Ketika <i>Creep Menyerang</i> .....	35
4.1.6 <i>Potion</i> .....	36
4.1.7 Fitur <i>Pause dan Cheat</i> .....	37
4.2 PENGUJIAN.....	38
4.2.1 Pengujian Perhitungan Logika Fuzzy.....	38
4.2.2 Pengujian Tower Menembak <i>Creep</i> Berdasarkan <i>Priority Target</i> .....	40
4.2.3 Pengujian Tower Menggunakan Logika Fuzzy dan Tanpa Logika Fuzzy.....	41
BAB V PENUTUP.....	43
5.1 KESIMPULAN.....	43
5.2 SARAN.....	43
DAFTAR PUSTAKA.....	44

## DAFTAR GAMBAR

Gambar 2.1 Grafik keanggotaan kurva linear naik.....	7
Gambar 2.2 Grafik keanggotaan kurva linear turun.....	7
Gambar 2.3 Grafik keanggotaan kurva segitiga.....	8
Gambar 2.4 Grafik keanggotaan kurva trapesium.....	8
Gambar 2.5 Grafik keanggotaan kurva bahu pada variabel umur.....	9
Gambar 2.6 Grafik keanggotaan kurva trapesium milik <i>moisture</i> .....	10
Gambar 2.7 <i>Center of Gravity</i> atau <i>Centroid</i> .....	11
Gambar 2.8 Logo <i>Unity</i> .....	12
Gambar 2.9 Tampilan Muka <i>Unity</i> .....	13
Gambar 2.10 <i>Project Browser</i> .....	13
Gambar 2.11 <i>Hierarchy</i> .....	13
Gambar 2.12 <i>Toolbar</i> .....	14
Gambar 2.13 <i>Scene View</i> .....	14
Gambar 2.14 <i>Game View</i> .....	14
Gambar 2.15 <i>Inspector</i> .....	15
Gambar 3.1 Blok Diagram Menu.....	17
Gambar 3.2 <i>Scene Menu</i> .....	17
Gambar 3.3 <i>Scene</i> Mulai level 1.....	18
Gambar 3.4 <i>Scene</i> Mulai level 2.....	18
Gambar 3.5 <i>Scene</i> Mulai level 3.....	18
Gambar 3.6 <i>Scene</i> Nilai.....	19
Gambar 3.7 <i>Scene</i> Tentang.....	19
Gambar 3.8 <i>Pause dan Cheat</i> .....	20
Gambar 3.9 Fungsi Keanggotaan Variabel <i>Speed</i> .....	26
Gambar 3.10 Fungsi Keanggotaan Variabel <i>Aspeed</i> .....	27
Gambar 3.11 Fungsi Keanggotaan Variabel <i>Health Point</i> .....	28
Gambar 3.12 Fungsi Keanggotaan Variabel <i>Damage</i> .....	28
Gambar 3.13 Fungsi Keanggotaan Variabel <i>Priority Target</i> .....	29
Gambar 3.14 Analisis menggunakan Fuzzy Tool box Matlab.....	31
Gambar 4.1 <i>Scene Menu</i> .....	32
Gambar 4.2 <i>Scene</i> Mulai Level 1.....	33

Gambar 4.3 <i>Scene</i> Mulai Level 2.....	33
Gambar 4.4 <i>Scene</i> Mulai Level 3.....	34
Gambar 4.5 <i>Scene</i> Nilai .....	34
Gambar 4.6 <i>Scene</i> Tentang.....	35
Gambar 4.7 <i>Creep</i> sebelum menyerang.....	35
Gambar 4.8 <i>Creep</i> sedang menyerang.....	36
Gambar 4.9 <i>Potion</i> yang jatuh .....	36
Gambar 4.10 <i>Hero</i> mengambil <i>red potion</i> .....	37
Gambar 4.11 Fitur <i>Pause</i> dan <i>Cheat</i> .....	37
Gambar 4.12 Hasil <i>cheat</i> menambah <i>damage</i> .....	38

## DAFTAR TABEL

Tabel 2.1 Rule untuk <i>Sprinkler Control System</i> .....	10
Tabel 3.1 Tabel Atribut <i>NPC</i> dan <i>Player</i> .....	24
Tabel 3.2 Tabel Atribut <i>Item</i> .....	25
Tabel 3.3 Tabel Himpunan Fuzzy.....	25
Tabel 3.4 Tabel <i>Rule Fuzzy</i> .....	30
Tabel 4.1 Pengujian Perhitungan.....	38
Tabel 4.2 Pengujian Tower Menembak Creep .....	40
Tabel 4.3 Pengujian Tower Menggunakan Fuzzy Mamdani dan Tanpa Fuzzy Mamdani .....	41

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

*Tower Defence* secara harfiah tower berarti menara dan *Defence* berarti bertahan. Namun makna secara istilah *tower defence* memiliki arti *game* yang *player*-nya harus mempertahankan wilayah yang sudah ditentukan agar *player* lain atau musuh (*NPC*) tidak dapat melewati atau menghancurkan area tersebut. *Tower defence* adalah salah satu sub *genre* dari *game* ber-*genre* *RTS (Real Time Strategy)*.

Contoh *game tower defence* adalah *game Plant vs Zombie*. Di *game* tersebut yang disebut tower adalah *plant*. *Plant* dapat menembak *zombie*. Pemain harus membendung serangan *zombie* yang selalu *spawning*. Permainan dinyatakan kalah bila *zombie* menghancurkan *plant* dan melewati area yang telah ditentukan. Sebaliknya permainan akan dinyatakan menang, bila pemain dapat bertahan dari serangan *zombie* dalam kurun waktu tertentu.

Pada *Game DoTa 2* yang disebut tower adalah bangunan yang menyerupai tower. Tower dapat menembak musuh yang telah masuk ke area tembaknya. Tower memiliki spesifikasi untuk menembak berdasarkan prioritasnya. Namun prioritas tembak tidak menggunakan Fuzzy. Melainkan dengan memprioritaskan musuh dengan jarak terdekat dari tower.

Tower pada *game*, umumnya tidak memiliki kecerdasan buatan yang dapat berpikir. Tidak dapat membedakan musuh yang memiliki tingkat bahaya yang tinggi dan rendah. Biasanya tower hanya menembak musuh berdasarkan urutan masuk area tembak saja. Setiap jenis musuh memiliki nilai atribut yang berbeda – beda. Hal ini membuat tower tidak dapat bertahan lebih lama.

Logika fuzzy adalah salah satu cabang dari kecerdasan buatan. Kecerdasan buatan merupakan representasi kecerdasan manusia yang ditiru dan diterapkan pada komputer. Logika fuzzy dapat mengenali diantara *true* dan *false*. Logika fuzzy dapat digunakan di berbagai bidang, termasuk dalam *game*. Logika Fuzzy menggunakan Metode *Mamdani*.

*Unity* adalah *game engine* untuk membuat *game multi platform*. *Unity* mampu digunakan untuk membuat *3D* dan *2D game multiplatform*. Platform yang didukung adalah *BlackBerry 10, Windows Phone 8, Windows, OS X, Android, iOS, Unity Web Player (termasuk Facebook), Adobe Flash, PlayStation 3, PlayStation 4, PlayStation Vita, Xbox 360, Xbox One, Wii U, New 3DS dan Wii*. *Unity* dapat digunakan untuk membuat berbagai macam *game*. Selain untuk membuat *game* bergenre *First Person Shooting (FPS)*, *Unity* juga dapat digunakan untuk membuat *game* bergenre *RPG, RTS*, dan lain sebagainya.

Dengan demikian penulis mengembangkan *game tower defence* yang towernya memiliki kecerdasan buatan. Kecerdasan buatan yang digunakan adalah Logika Fuzzy Mamdani. *Game Engine* yang digunakan adalah *Unity*.

### 1.2. Rumusan Masalah

Rumusan masalah pada skripsi ini adalah:

1. Bagaimana mengembangkan *game* ber-genre *tower defence* yang towernya memiliki kecerdasan buatan?
2. Bagaimana menerapkan logika fuzzy pada *Non-Player Character (NPC)* tower?
3. Bagaimana tower dapat menembak musuh berdasarkan *priority target*?

### 1.3. Batasan Masalah

Batasan masalah dari pembuatan *game* ini adalah :

1. Menggunakan *Game Engine Unity 2D* yang dijalankan pada *desktop Windows Operating System*.
  2. Kecerdasan buatan yaitu logika fuzzy menggunakan *library DotFuzzy*.
  3. Metode penalaran fuzzy yang digunakan adalah metode Mamdani.
  4. Penerapan logika fuzzy pada tower.
  5. *Gameplay* berdasarkan rancangan penulis.
  6. *Asset spritesheet, background, dan background sound* menggunakan *free license* dari internet.
  7. Variabel *input* fuzzy yang digunakan adalah *speed, aspeed, health point, dan damage*.
-

8. Variabel *speed* dan *aspeed* terdiri dari tiga himpunan fuzzy, yaitu lambat, normal dan cepat.
9. Variabel *health point* terdiri dari tiga himpunan fuzzy, yaitu sedikit, sedang dan banyak.
10. Variabel *demage* terdiri dari tiga himpunan fuzzy, yaitu lemah, normal dan kuat.
11. Variabel *priority target* sebagai variabel *output* terdiri dari lima himpunan fuzzy, yaitu sangat rendah, rendah, normal, tinggi dan sangat tinggi.

#### 1.4. Tujuan

Adapun yang menjadi tujuan penulisan dalam penyusunan skripsi adalah berikut:

1. Mengembangkan game *Tower Defence* yang towernya memiliki kecerdasan buatan.
2. Menerapkan fuzzy pada tower.
3. Tower dapat menembak musuh berdasarkan *priority target*.

#### 1.5. Metodologi Penelitian

Untuk mencapai mengembangkan game *tower defence*, maka langkah-langkahnya sebagai berikut :

1. Studi Literatur *Logika Fuzzy* dan *Game Tower Defence*

Pengumpulan data dilakukan dengan cara mengambil sumber referensi dari buku teks ataupun dari situs internet mengenai cara kerja logika fuzzy. Dan mencari referensi tentang *game tower defence* terutama pada game *Plant vs Zombie* dan *Dota 2*.

2. Konsep Permainan *Game Tower Defence*

Perancangan *Gameplay*, perancangan karakter dan *balancing* pada permainan berdasarkan perancangan penulis.

3. Desain Karakter

Pemilihan desain karakter yang akan digunakan pada game. Penulis menggunakan *free asset* yang didapat dari internet.

---

#### 4. Pembuatan *Game Tower Defence*

Pada tahap ini *game tower defence* yang telah dirancang kemudian diimplementasikan menggunakan *Unity* sebagai *game engine*-nya.

#### 5. Pengujian *Game*

Pengujian ini dengan menjalankan *game* dengan tujuan menemukan kesalahan serta menguji hasil dari penerapan logika fuzzy pada *game*.

### 1.6 Sistematika Penulisan

Sistematika penulisan pada penulisan skripsi ini sebagai berikut :

#### BAB I : PENDAHULUAN

Bab ini, membahas mengenai latar belakang pemilihan judul, perumusan masalah, tujuan, pembatasan masalah, metodologi penelitian serta sistematika penulisan.

#### BAB II : LANDASAN TEORI

Bab ini berisi penjelasan teori dari berbagai sumber pustaka. Teori – teori berhubungan dengan *game tower defence*, logika fuzzy, *library dotfuzzy*, *unity*, dan *matlab* akan dijelaskan pada bab ini.

#### BAB III : ANALISIS DAN PERANCANGAN

Berisi mengenai proses perancangan *gameplay*, atribut *NPC* pada *game*, dan analisis logika fuzzy yang diterapkan pada tower di *game*.

#### BAB IV : IMPLEMENTASI DAN PENGUJIAN

Bab ini membahas mengenai hasil dari *game* yang telah dikembangkan, tampilan *game* dan hasil dari pengujian.

#### BAB V : PENUTUP

Bab ini membahas tentang kesimpulan dan saran. Kesimpulan akan menjelaskan hasil dari evaluasi *game* dan saran menjelaskan tentang pengembangan *game* selanjutnya.

---

## BAB II

### LANDASAN TEORI

#### 2.1 Tower Defence

*Tower Defence* merupakan sub genre dari *Real Time Strategy* yang berfokus pada pengaturan sumber daya yang dimiliki pemain dan penempatan menara pertahanan (tower). Tower ini yang akan membendung laju musuh sehingga tidak dapat memasuki area tertentu atau merusak area tertentu. Pemilihan jenis tower dan penempatan tower adalah dasar strategi dalam permainan ini. (Satriawan, 2014)

Contoh *Game* dari *Tower Defence* adalah *Plant vs Zombie* dan *Dota 2*. Di *game Plant vs Zombie* yang disebut tower adalah *plant*. *Plant* dapat menembak *zombie* (*NPC* musuh). Pemain harus membendung serangan *zombie*. *Zombie* selalu *spawning* memasuki area permainan dan menghancurkan *plant*. Permainan dinyatakan kalah bila *zombie* menghancurkan *plant* dan melewati area yang telah ditentukan. Sebaliknya permainan akan dinyatakan menang, bila pemain dapat bertahan dari serangan *zombie* dalam kurun waktu tertentu.

Sementara itu, pada *Dota 2*, tower memiliki target prioritas untuk menembak musuh berdasarkan dari urutan prioritas sebagai berikut :

1. Unit musuh terdekat atau *hero* musuh menyerang *hero player* (mode menyerang otomatis).
2. Unit musuh terdekat atau *hero* musuh menyerang tower *player* (mode menyerang otomatis).
3. Unit musuh terdekat atau *hero* musuh menyerang unit *player* (mode menyerang otomatis).
4. Unit musuh terdekat.
5. *Hero* musuh terdekat.
6. Pelontar musuh terdekat.

Tower mengganti targetnya berdasarkan beberapa hal, salah satunya adalah jika unit musuh tertarget atau *hero* musuh tertarget keluar dari *range* tembak tower dan jika unit musuh tertarget atau *hero* musuh tertarget mati. (*Dota 2 Wiki*, 2015)

## 2.2 Logika Fuzzy

Logika Fuzzy adalah salah satu kecerdasan buatan. Logika fuzzy merupakan peningkatan dari logika *boolean* yang berhadapan dengan konsep kebenaran sebagian. Saat logika klasik menyatakan bahwa segala hal dapat diekspresikan dalam istilah *biner* (0 atau 1, hitam atau putih, ya atau tidak), logika fuzzy menggantikan kebenaran *boolean* dengan tingkat kebenaran.

Logika fuzzy memungkinkan nilai keanggotaan antara 0 dan 1, tingkat keabuan dan juga hitam dan putih, dan dalam bentuk linguistik, konsep tidak pasti seperti "sedikit", "lumayan", dan "sangat". Logika ini berhubungan dengan *set fuzzy* dan teori kemungkinan. Logika fuzzy diperkenalkan oleh *Dr. Lotfi Zadeh* dari *Universitas California, Berkeley* pada 1965.

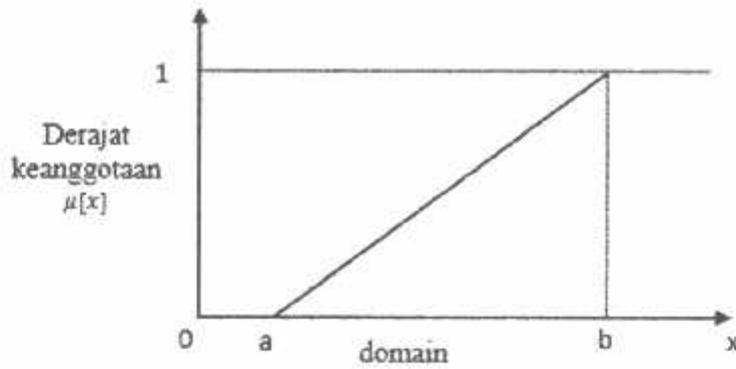
Bila dibandingkan dengan logika konvensional, kelebihan logika fuzzy adalah kemampuannya dalam proses penalaran secara bahasa sehingga dalam perancangannya tidak memerlukan persamaan matematik yang rumit. (Sutojo *et.all*, 2011)

### 2.2.1 Fungsi Keanggotaan

Fungsi keanggotaan merupakan yang mewakili besar dari derajat keanggotaan masing – masing variabel input yang berada dalam interval 0 dan 1. Derajat keanggotaan sebuah variabel  $x$  dilambangkan dengan simbol  $\mu(x)$ . Aturan – aturan menggunakan nilai keanggotaan sebagai faktor bobot untuk menentukan pengaruhnya pada saat melakukan inferensi untuk menarik kesimpulan. Beberapa contoh fungsi keanggotaan menurut (Sutojo *et.all*, 2011).

#### 1. Grafik Keanggotaan Kurva Linear

Pada grafik keanggotaan linear, sebuah variabel input dipetakan ke derajat keanggotaannya dengan digambarkan sebagai suatu garis lurus. Ada 2 grafik keanggotaan linear. Pertama, grafik keanggotaan kurva linear naik, yaitu kenaikan himpunan fuzzy dimulai pada nilai domain yang memiliki derajat keanggotaan nol [0] bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi. Grafik keanggotaan kurva linear naik ditujukan pada Gambar 2.1 sebagai berikut.

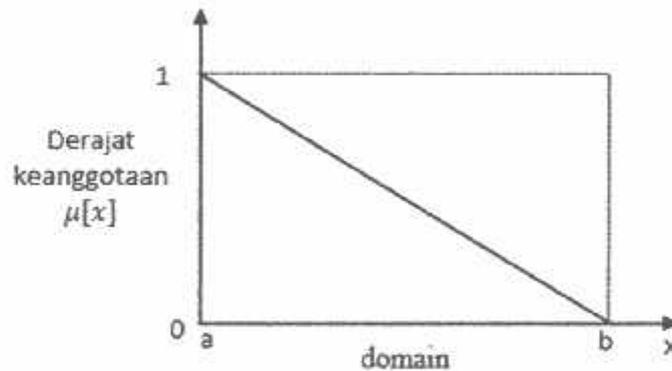


Gambar 2.1 Grafik keanggotaan kurva linear naik

Fungsi Keanggotaan:

$$\mu(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & x \geq b \end{cases}$$

Kedua, grafik keanggotaan kurva linear turun. Yaitu himpunan fuzzy dimulai dari nilai domain derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah seperti pada Gambar 2.2.



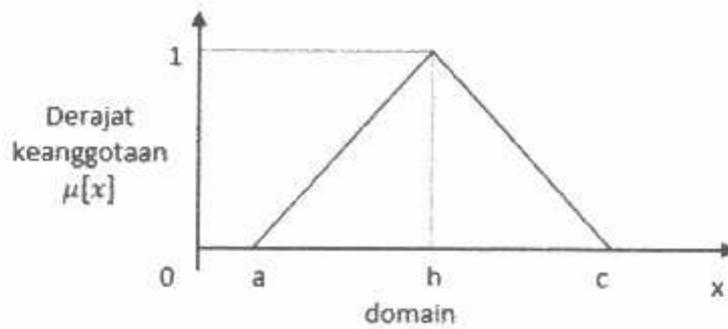
Gambar 2.2 Grafik keanggotaan kurva linear turun

Fungsi Keanggotaan:

$$\mu(x) = \begin{cases} \frac{b-x}{b-a}, & a \leq x \leq b \\ 0, & x \geq b \end{cases}$$

## 2. Grafik Keanggotaan Kurva Segitiga

Grafik keanggotaan segitiga pada dasarnya merupakan gabungan antara 2 garis (linear) seperti terlihat pada Gambar 2.3.



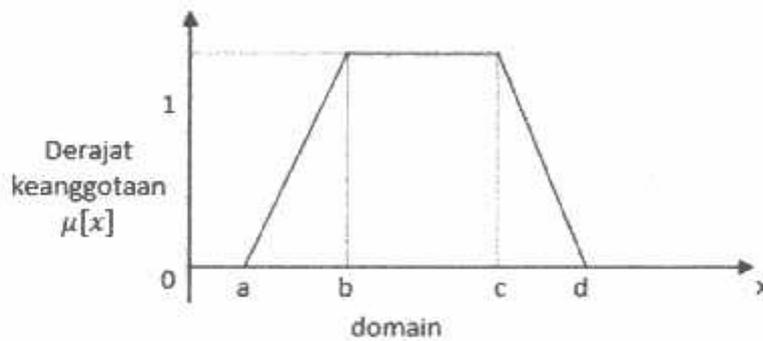
Gambar 2.3 Grafik keanggotaan kurva segitiga

Fungsi Keanggotaan:

$$\mu(x) = \begin{cases} 0, & x \leq a \text{ atau } x \geq c \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{b-x}{c-b}, & b \leq x \leq c \end{cases}$$

### 3. Grafik Keanggotaan Kurva Trapesium

Grafik keanggotaan kurva trapesium pada dasarnya seperti bentuk segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan 1. Grafik keanggotaan kurva trapesium ditujukan pada Gambar 2.4 sebagai berikut.



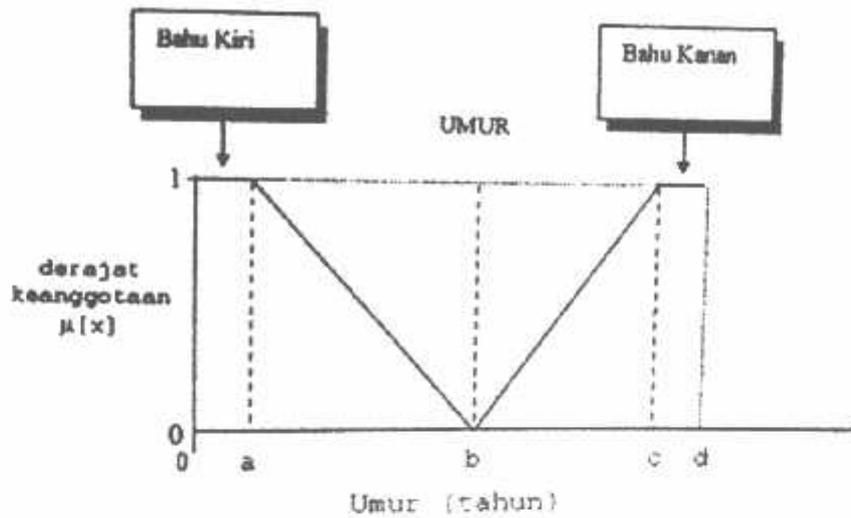
Gambar 2.4 Grafik keanggotaan kurva trapesium

Fungsi Keanggotaan

$$\mu(x) = \begin{cases} 0, & x \leq a \text{ atau } x \geq d \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \end{cases}$$

#### 4. Grafik Keanggotaan Kurva Bentuk Bahu

Grafik keanggotaan kurva “bahu” digunakan untuk mengakhiri variabel suatu daerah fuzzy yang nilai derajat keanggotaannya adalah konstan (biasanya 1). Kurva bentuk bahu ditunjukkan pada Gambar 2.5 sebagai berikut.



Gambar 2.5 Grafik keanggotaan kurva bahu pada variabel umur

Fungsi Keanggotaan

$$\mu(x) = \begin{cases} 1, & 0 \leq x \leq a \text{ atau } c \leq x \leq d \\ \frac{b-x}{b-a}, & a \leq x \leq b \\ \frac{x-b}{c-b}, & b \leq x \leq c \end{cases}$$

#### 2.2.2 Metode Mamdani

Metode Mamdani paling sering digunakan dalam aplikasi – aplikasi karena strukturnya yang sederhana, yaitu menggunakan operasi *MIN-MAX*. (Sutojo et.all, 2011) Teori metode *min-max* menurut (Lee, 2005) adalah sebagai berikut :

1. *Union (Disjunction)* : Nilai dari himpunan x dalam *union* mengambil nilai terbesar antara himpunan A dan B.

$$\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)]$$

2. *Intersection (Conjunction)* : *Intersection* dalam set fuzzy A dan B mengambil nilai terkecil dari himpunan antara A dan B.

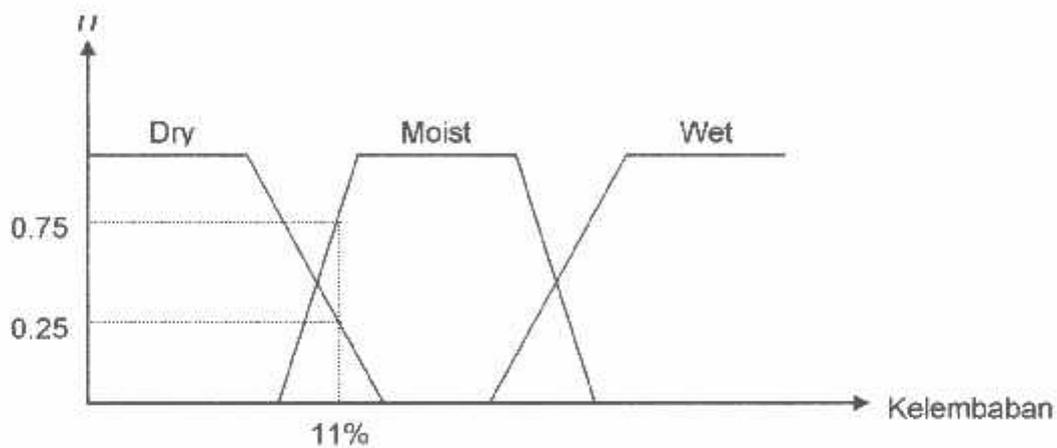
$$\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)]$$

Contoh untuk mendapatkan *crisp output* (bilangan tegas hasil defuzzyfikasi) dalam studi kasus *Sprinkler Control System* diperlukan 3 tahapan (Suyanto, 2012)

## 1. Fuzzyfikasi

Proses untuk mengubah input sistem yang mempunyai nilai tegas (*crisp*) menjadi variabel linguistik menggunakan fungsi keanggotaan seperti pada sub bab 2.2.1.

Membership Function untuk *Temperature* sudah diketahui yaitu temperature udara 92°F menjadi Warm = 0.2 Hot = 0.46. Membership Function untuk *Moisture* adalah trapesium. Kelembaban 11% menjadi Dry = 0.25, Moist = 0.75, Wet = 0. Gambar 2.6 merupakan kurva grafik keanggotaan dari *Moisture* (Kelembaban).



Gambar 2.6 Grafik keanggotaan kurva trapesium milik *moisture*

## 2. Rule Evaluation

Aturan yang diterapkan pada logika fuzzy berdasarkan keterkaitan antar *set fuzzy*. *Rule evaluation* menentukan besaran derajat keanggotaan set fuzzy. Rule yang telah ditetapkan pada contoh ini adalah seperti pada Tabel 2.1.

Tabel 2.1 Rule untuk *Sprinkler Control System*

		Temperature				
		Cold	Cool	Normal	Warm	Hot
Moisture	Dry	Long	Long	Long	Long	Long
	Moist	Long	Medium	Medium	Medium	Medium
	Wet	Short	Short	Short	Short	Short

Berdasarkan Tabel 2.1, *fuzzy rule* untuk *Sprinkler* adalah sebagai berikut:

[R1] IF Temperature is Hot (0.46) AND Soil is Dry (0.25) THEN watering duration is Long.

[R2] IF Temperature is Hot (0.46) AND Soil is Moist (0.75) THEN watering duration is Medium.

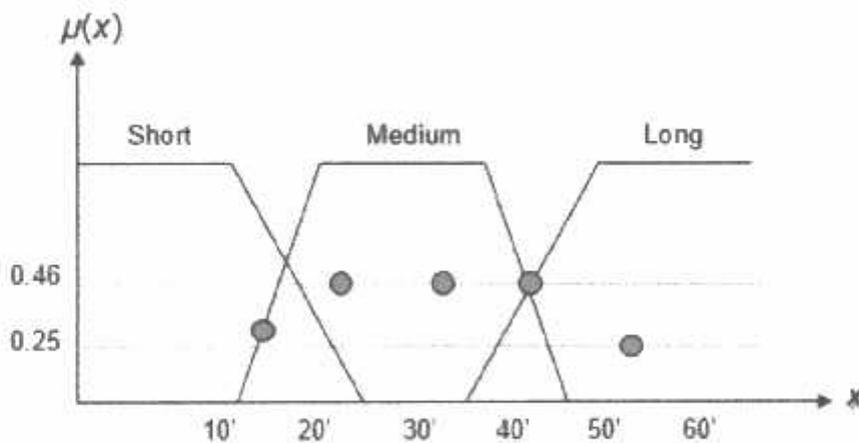
[R3] IF Temperature is Warm (0.2) AND Soil is Moist (0.75) THEN watering duration is Medium.

[R4] IF Temperature is Warm (0.2) AND Soil is Dry (0.25) THEN watering duration is Long.

Fuzzy Output: Water Duration is Long (0.25) dan Medium (0.46)

### 3. Proses Defuzzyfikasi

Metode yang digunakan: *Center of Gravity (CoG)* atau *Centroid Method*. Fuzzy Output yang berupa  $\mu$  dipotongkan ke grafik menggunakan metode *Lamda Cut ( $\lambda$ -Cut)*. *Lamda-Cut* adalah membatasi maximum kebenaran dari himpunan fuzzy atau  $\mu$ . Untuk setiap  $\mu_A$  berlaku  $\mu(x) = \min_A(\mu(x), \lambda\text{-Cut}_A)$ . Gambar 2.7 merupakan ilustrasi metode *centroid*.



Gambar 2.7 *Center of Gravity* atau *Centroid*

Pada metode ini, solusi crisp diperoleh dengan cara mengambil titik pusat daerah fuzzy. Secara umum untuk diskrit dirumuskan :

$$z_0 = \frac{\sum_{j=a}^b \mu(x) \cdot x}{\sum_{x=a}^b \mu(x)}$$

$$= \frac{((15 * 0.3) + (25 * 0.46) + (35 * 0.46) + (45 * 0.46) + (55 * 0.25))}{(0.3 + 0.46 + 0.46 + 0.46 + 0.25)} = 34.5$$

### 2.3 Library DotFuzzy

*DotFuzzy* dibuat oleh oleh *Michele Bertoli*. *Library* ini dibuat untuk menyelesaikan program *bachelor*-nya pada tahun 2008. Namun sejak saat itu, *library* ini tidak pernah di *update* atau *maintenance*.

*DotFuzzy* merupakan *open source class library* untuk logika *fuzzy*. *Library* ini dibangun dengan *C#*. Karena hal itu, *library* ini dapat digunakan oleh semua bahasa lingkungan NET. Dibangun Berorientasi Obyek, sehingga *library* ini mudah digunakan. *DotFuzzy* dirancang untuk digunakan secara fleksibel. *DotFuzzy* mengimplementasikan *fuzzification*, *rules validation / evaluation* dan *defuzzification* dengan metode *centroid*. (Bertoli, 2014)

*Library DotFuzzy* menggunakan Mode *B-FITA* (*first infer, then aggregate*) untuk defuzzifikasinya. Mode *B-FITA* menurut (Cordon *et.all*, 2001) adalah teknik pengerjaan defuzzifikasi mamdani dengan cara menyimpulkan terlebih dahulu setelah itu menggabungkannya.

### 2.4 Unity

*Unity Technologies* dibangun sejak tahun 2004. *Game Engine* adalah *software* yang digunakan untuk membuat *game* digital. *Unity* adalah *game engine* untuk membuat *game multi platform*.

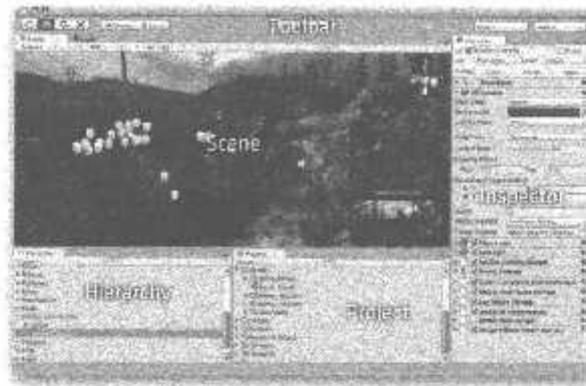
*Unity* mampu digunakan untuk membuat *3D* dan *2D game multiplatform*. *Platform* yang didukung adalah *BlackBerry 10*, *Windows Phone 8*, *Windows*, *OS X*, *Android*, *iOS*, *Unity Web Player* (termasuk *Facebook*), *Adobe Flash*, *PlayStation 3*, *PlayStation 4*, *PlayStation Vita*, *Xbox 360*, *Xbox One*, *Wii U*, *New 3DS* dan *Wii*.

*Unity* dapat digunakan untuk membuat berbagai macam *game*. Secara *default*, *Unity* telah diatur untuk pembuatan *game* ber-genre *First Person Shooting (FPS)*. Namun *Unity* juga dapat digunakan untuk membuat *game* bergenre *RPG*, *RTS*, dan lain sebagainya. (Roedavan, 2014). Gambar 2.8 adalah *Logo Unity*.



Gambar 2.8 Logo Unity

### 2.4.1 Interface Unity

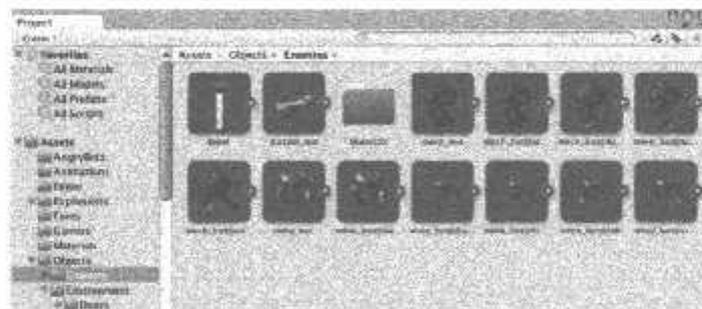


Gambar 2.9 Tampilan Muka *Unity*

Tampilan muka pada *Unity* yang dijelaskan pada (Unity3d, 2015) adalah: *Project Browser*, *Hierarchy*, *Toolbar*, *Scene View*, *Game View*, *Inspector*. Pada halaman ini pengguna dapat mengembangkan *game*. Tampilannya seperti pada Gambar 2.9.

#### a. *Project Browser*

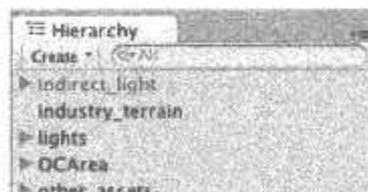
Pada tampilan *Project Browser*, digunakan untuk mengakses dan memajemen semua *assets* pada *project*. Seperti pada Gambar 2.10.



Gambar 2.10 *Project Browser*

#### b. *Hierarchy*

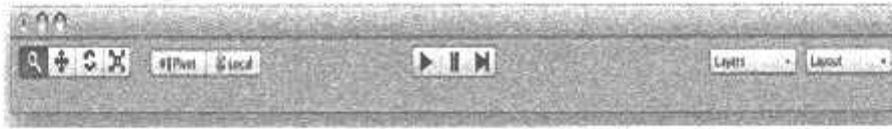
*Hierarchy* berisi semua *GameObject* yang terdapat pada *Scene*. Seperti pada Gambar 2.11.



Gambar 2.11 *Hierarchy*

c. *Toolbar*

*Toolbar* secara *default* menyediakan lima kontrol dasar. Masing – masing kontrol memiliki kegunaan tersendiri. Seperti pada Gambar 2.12.



Gambar 2.12 *Toolbar*

d. *Scene View*

*Scene View* digunakan untuk memilih dan mengatur posisi lingkungan, *player*, kamera, musuh, dan *GameObject*. Seperti pada Gambar 2.13.



Gambar 2.13 *Scene View*

e. *Game View*

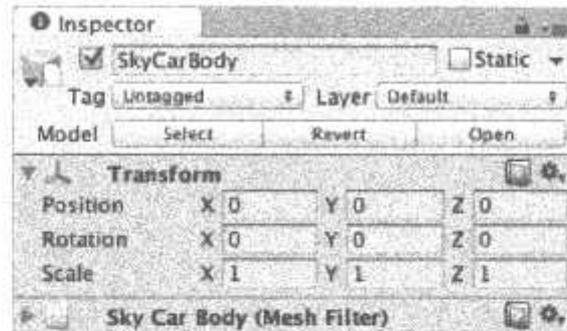
*Game View* di-*rendering* dari kamera *game* yang dibuat. Hal ini merepresentasikan yang sudah ter-*publish*. Seperti pada Gambar 2.14.



Gambar 2.14 *Game View*

#### f. Inspector

*Game* di *Unity* dibuat berdasarkan *multiple GameObject* dimana memiliki *meshes, scripts, sounds*, atau elemen grafis lainnya seperti cahaya. *Inspector* merupakan fitur dari *Unity* untuk mengatur atribut yang terdapat pada *GameObject*. Tampilannya seperti pada Gambar 2.15.



Gambar 2.15 *Inspector*

#### 2.4.2 IDE Visual Studio C# Integration

*Unity* dapat terintegrasi dengan *Visual Studio* untuk editor script *C#*. Kemudahan yang didapat menggunakan *Visual Studio* adalah *smart autocompletion, smart syntax highlighting*, dan lain lain. Untuk mengatur integrasi *Unity* dengan *Visual Studio* memilih "*Assets->Sync MonoDevelop Project*".

#### 2.4.3 Script

*Unity* dapat digunakan menggunakan dua bahasa pemrograman yaitu *C#* dan *javascript*. Contoh default script *C#* adalah :

```
using UnityEngine;
using System.Collections;
public class MainPlayer : MonoBehaviour {
    void Start () {
    }
    void Update () {
    }
}
```

Sedangkan untuk *JavaScript* seperti berikut :

```
#pragma strict
function Start () {
}
function Update () {
}
```

## 2.5 Matlab Fuzzy Logic Toolbox

*Fuzzy Logic Toolbox*™ menyediakan fungsi, aplikasi, dan blok *Simulink*® untuk menganalisis, merancang, dan sistem simulasi berdasarkan logika fuzzy. Produk ini memandu melalui langkah-langkah mendesain sistem inferensi fuzzy. Fungsi yang disediakan banyak, yaitu metode yang umum pada fuzzy, termasuk *fuzzy clustering* dan *adaptive neurofuzzy learning*. (Fuzzy Logic Toolbox, 2015)

*Toolbox* ini memungkinkan untuk memodelkan perilaku sistem yang kompleks menggunakan aturan logika sederhana. Kemudian menerapkan aturan ini dalam sistem inferensi fuzzy.

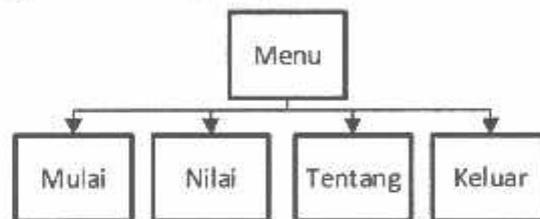
---

## BAB III

### ANALISIS DAN PERANCANGAN

#### 3.1 Blok Diagram Menu

Saat Permainan dijalankan, terdapat tampilan utama yaitu Menu. Pemain dapat memilih “Mulai” untuk bermain, “Nilai” untuk melihat nilai terakhir dari permainan, “Tentang” untuk melihat informasi *game*, “Keluar” untuk keluar dari permainan. Blok diagram menu seperti pada Gambar 3.1.



Gambar 3.1 Blok Diagram Menu

#### 3.2 Perancangan Scene

##### 3.2.1 Scene Menu

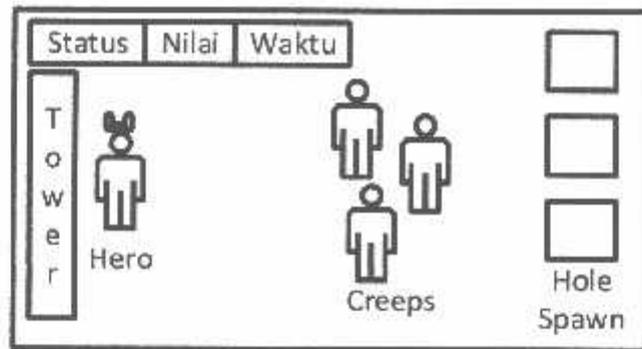
Tampilan layout Menu utama. Memiliki 4 tombol yaitu “Mulai”, “Nilai”, “Tentang” dan “Keluar”. Seperti pada Gambar 3.2 berikut.



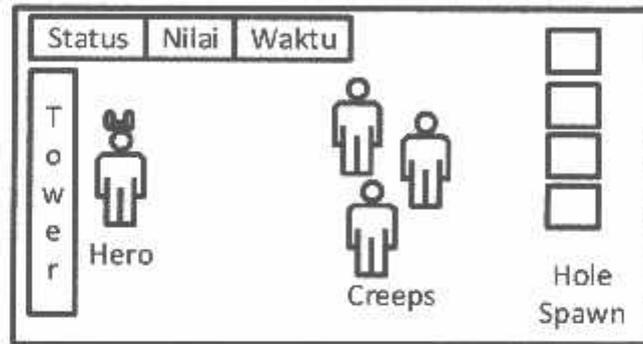
Gambar 3.2 Scene Menu

##### 3.2.2 Scene Mulai

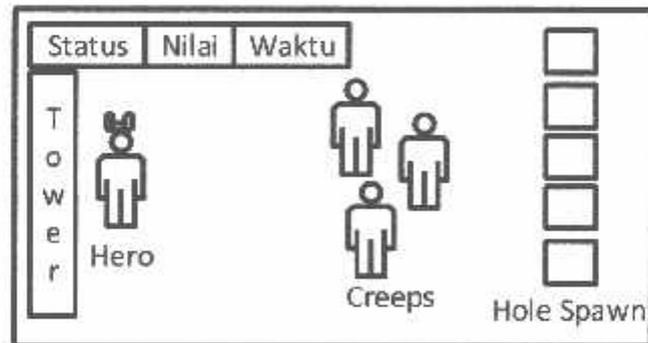
*Scene* Mulai seperti pada Gambar 3.3, Gambar 3.4, dan Gambar 3.5, terdapat status permainan dibagian atas. Tower dan *Hero* berada di sebelah kiri, *Hole Spawn* dan *Creep* di sebelah kanan. Terdapat tiga level permainan. Level 1 memiliki 3 *Hole Spawn*, level 2 memiliki 4 *Hole Spawn*, dan level 3 memiliki 5 *Hole Spawn*.



Gambar 3.3 Scene Mulai level 1



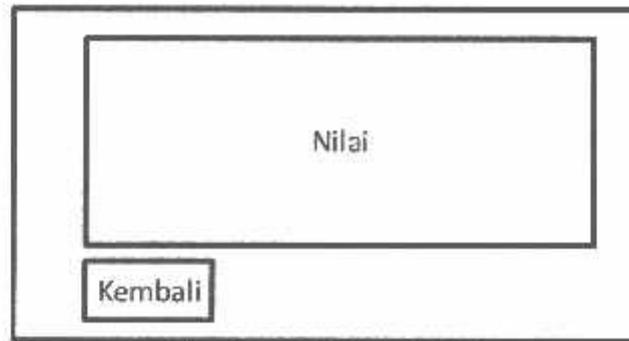
Gambar 3.4 Scene Mulai level 2



Gambar 3.5 Scene Mulai level 3

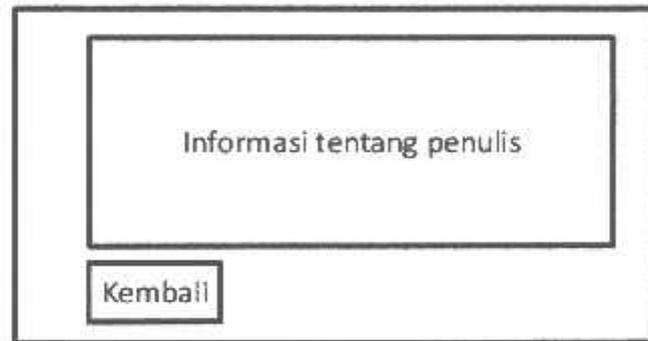
### 3.2.3 Scene Nilai

*Scene Nilai* menampilkan nilai, pemain harus ke menu utama dan memilih tombol "Nilai". Nilai terakhir dari permainan yang pernah dimainkan yang akan muncul. Namun nilai tersebut akan menjadi nol (*default*) kembali jika keluar dari *game*. *Scene Nilai* seperti pada Gambar 3.6.

Gambar 3.6 *Scene Nilai*

### 3.2.4 *Scene Tentang*

*Scene Tentang* berisi tentang informasi *about us* dan informasi *game*. Informasi *about us* menjelaskan siapa pembuat *game*. Informasi *game* menjelaskan bagaimana cara bermain. Tombol kembali untuk kembali ke menu utama seperti pada Gambar 3.7.

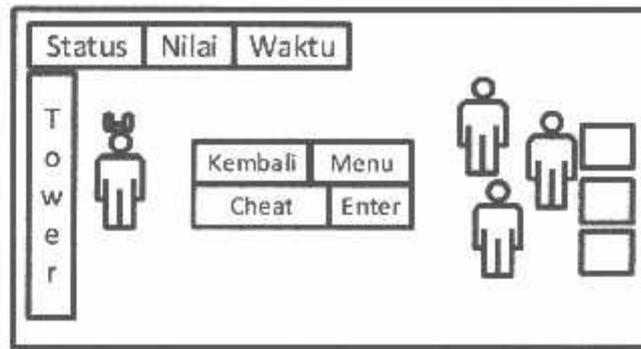
Gambar 3.7 *Scene Tentang*

### 3.2.5 *Pause dan Cheat*

Pemain harus menekan tombol ESC pada keyboard untuk mengaktifkan fitur *pause* dan *cheat*. Tombol “kembali” untuk kembali bermain, tombol “Menu” untuk kembali ke menu utama, “*cheat*” untuk menulis *cheat*, dan tombol “Enter” untuk mengaktifkan *cheat* seperti Gambar 3.8

Daftar *cheat* yang dapat digunakan adalah sebagai berikut:

1. Nyawa tidak terbatas: *unlimhp*
2. Menambah *damage hero* sebesar 1000: *inc damage*
3. Menambah kecepatan *hero* sebesar 12: *inc speed*
4. Menambah kecepatan serang *hero* sebesar 20: *inc aspeed*



Gambar 3.8 *Pause dan Cheat*

### 3.3 Perancangan *Gameplay*

#### 3.3.1 *Hero*

Pemain mengendalikan *Hero*. Pemain harus mempertahankan tower miliknya. Setiap *hero* terbunuh akan mengurangi score dan 5 detik kemudian akan respawn di tempat dimana hero mati berada. Sehingga untuk memainkannya membutuhkan strategi.

Level yang harus diselesaikan oleh hero adalah sebagai berikut:

- a. Level 1, *Hero* dan Tower harus dapat bertahan selama 5 menit. Jika berhasil, maka permainan akan naik ke level 2.
- b. Level 2, *Hero* dan Tower harus dapat bertahan selama 7 menit. Jika berhasil, maka permainan akan naik ke level 3.
- c. Level 3, *Hero* dan Tower harus dapat bertahan selama 9 menit. Jika berhasil, maka permainan berakhir.

#### 3.3.2 *Tower*

Tower adalah *NPC* protagonis. Tugasnya adalah membantu pemain untuk menembak *creep* berdasarkan *priority target*. Permainan berakhir ketika tower hancur.

#### 3.3.3 *Creep*

*Creep* adalah *Non Player Character (NPC)* musuh. Terdapat 3 jenis *Creep* berdasarkan besar *damage*-nya yaitu *Creep A*, *Creep B*, dan *Creep C*. *Creep* spawning dari *Hole Spawn*. Atribut *Creep* bersifat *random*.

### 3.3.4 Hole Spawn

*Hole Spawn* adalah *NPC* yang tidak dapat diserang maupun menyerang. Fungsinya adalah tempat untuk *spawning Creep*. Waktu (*cooldown*) untuk *spawning* bersifat *random* dari 10 detik sampai dengan 25 detik. Jenis *Creep* yang *spawning* juga bersifat *random*. Pada Level 1 terdapat 3 *Hole Spawn*. Pada Level 2 terdapat 4 *Hole Spawn*. Pada Level 3 terdapat 5 *Hole Spawn*.

### 3.3.5 Item

*Item* merupakan benda yang jatuh dari hasil membunuh *Creep*. Terdapat 3 jenis *item* berdasarkan kegunaannya. *Green Potion* untuk menambah *Health Point (HP) Tower*. *Red Potion* untuk menambah *Health Point (HP) Hero*. *Purple Potion* untuk menambah *Damage Hero*.

### 3.3.6 Perhitungan Score

Setiap berhasil membunuh *creep*, pemain akan mendapatkan *score*. Perhitungan *score* menggunakan rumus

$$\text{score} = \left( \sum \text{creep terbunuh} \times 1 \right) - \left( \sum \text{hero terbunuh} \times 2 \right)$$

Misal *player* telah membunuh 10 *creep* dan *hero* terbunuh 1 kali, maka :

$$\text{score} = (10 \times 1) - (1 \times 2)$$

$$\text{score} = 8$$

## 3.4 Atribut NPC dan Player

### 3.4.1 Player Hero

Dikendalikan oleh *player* menggunakan *control keyboard (A-W-S-D)* dan "L" untuk menyerang. Tugasnya adalah menjaga tower agar tidak hancur dengan cara melawan *creeps*. Hero dapat mati dan *respawn* kembali 5 detik kemudian. Hero memiliki atribut:

#### a. Health Point (HP)

Nyawa yang dimiliki oleh *hero*. Jika *health point hero* habis maka akan mati.

Nilai : 1000

b. *Damage*

*Damage* adalah angka *hit point* untuk mengurangi angka *health point* milik *creep*.

Nilai : 5% dari *max HP Creep* yaitu  $5\% * 500 = 25$

c. *Speed*

*Speed* adalah kecepatan gerak berpindah tempat.

Nilai : Panjang area lintasan 12 *pixel* skala *Unity*. Diharapkan 15 detik dapat menyelesaikan area lintasan. Maka didapatkan nilai  $12 \text{ pixel} / 15 \text{ detik} = 0.8 \text{ pixel} / \text{detik}$  atau setara dengan  $8 \text{ pixel} / 10 \text{ detik}$ .

d. *Aspeed / Attack Speed*

*Hero* memiliki kecepatan menyerang.

Nilai : Diharapkan dalam 1 detik *Hero* menyerang 1 kali. Maka didapat nilai.  $1 \text{ hit} / \text{detik}$  atau setara dengan  $10 \text{ hit} / 10 \text{ detik}$ .

### 3.4.2. NPC Tower

*NPC Tower* bertugas untuk membunuh musuh sekaligus yang harus dilindungi oleh pemain. Jika *Tower* hancur maka permainan selesai. *NPC tower* memiliki atribut:

a. *Health Point (HP)*

Nyawa yang dimiliki oleh *Tower*. Jika *health point* *tower* habis maka akan hancur. Dan permainan berakhir

Nilai : 300% dari *HP Hero* yaitu  $300\% * 1000 = 3000$

b. *Damage*

*Damage* adalah angka *hit point* untuk mengurangi angka *health point* milik musuh.

Nilai : 200% dari *Damage Hero* yaitu  $200\% * 25 = 50$

c. *Range*

*Tower* memiliki jangkauan tembak. Ketika ada musuh yang memasuki jangkauan tembaknya maka *tower* akan menembak.

Nilai : 75% dari skala area lintasan yaitu  $75\% * 12 = 9$

d. *Aspeed / Attack Speed*

*Tower* memiliki kecepatan menyerang.

Nilai : 50% dari *Aspeed Hero* yaitu  $50\% * 1 = 0,5 \text{ hit / detik}$  atau setara dengan  $5 \text{ hit / 10 detik}$ .

### 3.4.3. NPC Creep

*Creep* adalah *NPC* musuh yang berjalan menuju tower. *Creep spawning* dari *hole spawn*. Tugas dari *creep* adalah menghancurkan tower dan melawan *hero*. Semua atribut yang dimiliki *creep* bernilai *random*. Terdapat 3 jenis *Creep* berdasarkan besar *damage*-nya yaitu : *Creep A*, *Creep B*, dan *Creep C*. *NPC Creep* memiliki atribut :

a. *Health Point (HP)*

Nyawa yang dimiliki oleh *creep*. Jika *health point* *creep* habis maka *creep* akan mati.

Nilai : 0,1% sampai 50% dari *HP Hero*. Nilai  $\text{min} = 0,1\% * 1000 = 1$  dan  $\text{max} = 50\% * 1000 = 500$

b. *Damage*

*Damage* adalah angka *hit point* untuk mengurangi angka *health point* milik Tower dan *Hero*.

Nilai : 1% sampai 4% dari *HP Hero*. Nilai  $\text{min} = 1\% * 1000 = 10$  dan  $\text{max} = 4\% * 1000 = 40$ . Pembagian *Creep* menurut besar *damage*-nya adalah : *Creep A* = 10-20, *Creep B* = 21-30, *Creep C* = 31-40

c. *Speed*

*Speed* adalah kecepatan gerak berpindah tempat.

Nilai : Diharapkan 10 sampai 20 detik dapat menyelesaikan area lintasan. Maka di dapatkan nilai  $\text{min} = 12 \text{ pixel / 20 detik} = 0,6 \text{ pixel/detik}$  atau setara dengan  $6 \text{ pixel / 10 detik}$  dan  $\text{max} = 12 \text{ pixel / 10 detik} = 1,2 \text{ pixel/detik}$  atau setara dengan  $12 \text{ pixel / 10 detik}$ .

e. *Aspeed / Attack Speed*

*Creep* memiliki kecepatan menyerang.

Nilai : 80% sampai 200% dari *Aspeed Hero*. Nilai  $\text{min} = 80\% * 1 = 0,8 \text{ hit / detik}$  atau setara dengan  $8 \text{ hit / 10 detik}$  dan  $\text{max} = 200\% * 1 = 2 \text{ hit / detik}$  atau setara dengan  $20 \text{ hit / 10 detik}$ .

### 3.4.4. Item

*Item* merupakan benda jatuh yang didapat dari melawan *creep*. Terdapat 3 jenis *item* berdasarkan kegunaannya, yaitu *Green Potion*, *Red Potion*, dan *Purple Potion*. Satu *Creep* hanya dapat menjatuhkan satu jenis *item* saja. Masing – masing *item* memiliki probabilitas jatuh sebesar 1/10.

a. *Green Potion*

Untuk menambah *HP Tower*.

Nilai : 50% dari *max Damage Creep* yaitu  $50\% * 40 = 20$

b. *Red Potion*

Untuk menambah *HP Hero*.

Nilai : 50% dari *max Damage Creep* yaitu  $50\% * 40 = 20$

c. *Purple Potion*

Untuk menambah *Damage Hero*.

Nilai : 0.2% dari *max HP Creep* yaitu  $0.2\% * 500 = 1$

Untuk algoritma probabilitas *item* jatuh adalah sebagai berikut :

- Bangkitkan angka *random* dari 1 sampai 10.
- Jika angka *random* tersebut adalah 1 maka *item* yang jatuh adalah *Green Potion*.
- Jika angka *random* tersebut adalah 2 maka *item* yang jatuh adalah *Red Potion*.
- Jika angka *random* tersebut adalah 3 maka *item* yang jatuh adalah *Purple Potion*.
- Jika angka *random* tersebut adalah 4 sampai 10 maka *creep* tidak menjatuhkan *item*.

Dapat dirangkum menjadi Tabel 3.1 dan Tabel 3.2

Tabel 3.1 Tabel Atribut *NPC* dan *Player*

	Hero	Tower	Creeps A		Creeps B		Creeps C	
			min	max	min	max	min	max
HP	1000	3000	1	500	1	500	1	500
Damage	25	50	10	20	21	30	31	40
Speed	8	-	6	12	6	12	6	12
Range	-	9	-	-	-	-	-	-

	Hero	Tower	Creeps A		Creeps B		Creeps C	
			min	max	min	max	min	max
ASpeed	10	5	8	20	8	20	8	20

Tabel 3.2 Tabel Atribut *Item*

	Green Potion	Red Potion	Purple Potion
HP Tower	-20	-	-
HP Hero	-	+20	-
Damage Hero	-	-	+1
Probabilitas drop	1/10	1/10	1/10

### 3.5 Analisis Himpunan Fuzzy

Himpunan fuzzy yang diterapkan pada *game* diterapkan pada *Non Player Character (NPC) tower*. Himpunan fuzzy ditunjukkan pada tabel 3.3 berikut.

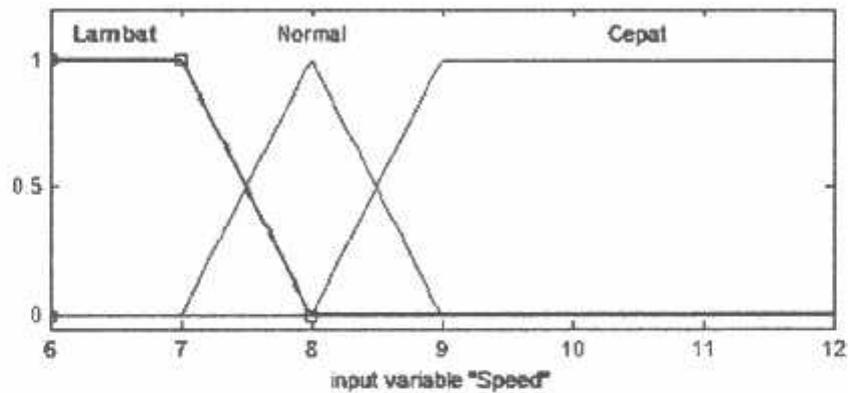
Tabel 3.3 Tabel Himpunan Fuzzy

Fungsi	Nama Variabel	Keterangan	Himpunan	Domain	Parameter
Input	Speed [6 - 12]	Kecepatan gerak (berjalan) milik creep	Lambat	[6 - 8]	[6;6;7;8]
			Normal	[7 - 9]	[7;8;8;9]
			Cepat	[8 - 12]	[8;9;12;12]
	ASpeed [8 - 20]	Kecepatan Serang creep	Lambat	[8 - 13]	[8;8;10;13]
			Normal	[10 - 18]	[10;14;14;18]
			Cepat	[15 - 20]	[15;18;20;20]
	Health Point [1 - 500]	Health Point milik creep	Sedikit	[1 - 200]	[1;1;90;200]
			Sedang	[100 - 400]	[100;250; 250;400]
			Banyak	[300 - 500]	[300;400;500;500]
Damage [10 - 40]	Damage milik creep	Lemah	[10 - 18]	[10;10;13;18]	
		Normal	[15 - 33]	[15;24;24;33]	
		Kuat	[30 - 40]	[30;35;40;40]	
Output	Priority Target [0-120]	Prioritas target tembak	Sangat Rendah	[0 - 40]	[0;0;20;40]
			Rendah	[20 - 60]	[20;40;40;60]
			Normal	[40 - 80]	[40;60;60;80]
			Tinggi	[60 - 100]	[60;80;80;100]
			Sangat Tinggi	[80 - 120]	[80;100;120;120]

Berdasarkan Tabel 3.3 Tabel himpunan fuzzy, maka pemetaan *input – output* adalah sebagai berikut :

1. Variabel *Speed*

Variabel *Speed* merupakan kecepatan gerak (berjalan) milik *Creep*. Variabel ini terbagi menjadi tiga himpunan fuzzy yaitu : Lambat, Normal, Cepat. Berikut ini fungsi keanggotaan untuk variabel *speed* pada Gambar 3.9.



Gambar 3.9 Fungsi Keanggotaan Variabel *Speed*

a. Himpunan Fuzzy Lambat

$$\mu_{Lambat}(x) = \begin{cases} 1, & x \leq 7 \\ \frac{8-x}{8-7}, & 7 < x \leq 8 \\ 0, & x > 8 \end{cases}$$

b. Himpunan Fuzzy Normal

$$\mu_{Normal}(x) = \begin{cases} 0, & x < 7 \text{ atau } x > 9 \\ \frac{x-7}{8-7}, & 7 \leq x \leq 8 \\ \frac{9-x}{9-8}, & 8 < x \leq 9 \end{cases}$$

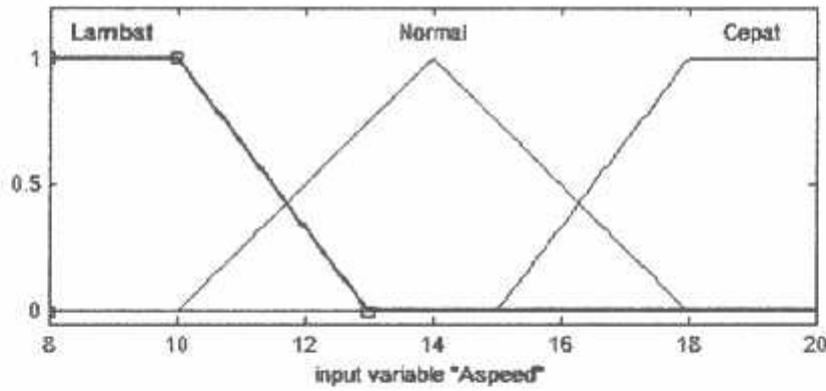
c. Himpunan Fuzzy Cepat

$$\mu_{Cepat}(x) = \begin{cases} 0, & x < 8 \\ \frac{x-8}{9-8}, & 8 \leq x < 9 \\ 1, & x \geq 9 \end{cases}$$

2. Variabel *Aspeed*

Variabel *Aspeed (Attack Speed)* merupakan kecepatan untuk menyerang milik *Creep*. Variabel ini terbagi menjadi tiga himpunan fuzzy yaitu :

Lambat, Normal, Cepat. Berikut ini fungsi keanggotaan untuk variabel *aspeed* pada Gambar 3.10.



Gambar 3.10 Fungsi Keanggotaan Variabel *Aspeed*

a. Himpunan Fuzzy Lambat

$$\mu_{Lambat}(x) = \begin{cases} 1, & x \leq 10 \\ \frac{13-x}{13-10}, & 10 < x \leq 13 \\ 0, & x > 13 \end{cases}$$

b. Himpunan Fuzzy Normal

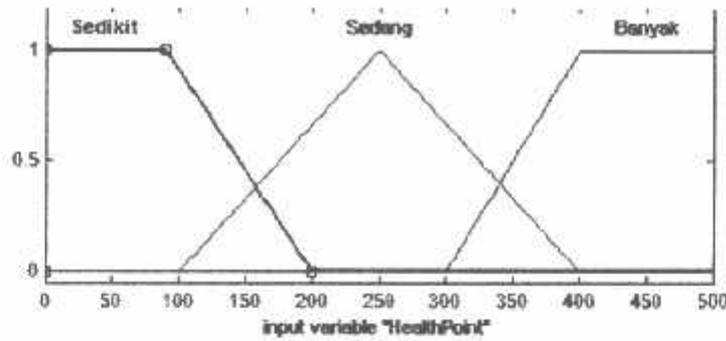
$$\mu_{Normal}(x) = \begin{cases} 0, & x < 10 \text{ atau } x > 18 \\ \frac{x-10}{14-10}, & 10 \leq x \leq 14 \\ \frac{18-x}{18-14}, & 14 < x \leq 18 \end{cases}$$

c. Himpunan Fuzzy Cepat

$$\mu_{Cepat}(x) = \begin{cases} 0, & x < 15 \\ \frac{x-15}{18-15}, & 15 \leq x < 18 \\ 1, & x \geq 18 \end{cases}$$

### 3. Variabel *Health Point*

Variabel *Health Point* merupakan nyawa milik *Creep*. Variabel ini terbagi menjadi tiga himpunan fuzzy yaitu : Sedikit, Sedang, Banyak. Berikut ini fungsi keanggotaan untuk variabel *health point* pada Gambar 3.11.

Gambar 3.11 Fungsi Keanggotaan Variabel *Health Point*

a. Himpunan Fuzzy Sedikit

$$\mu_{Sedikit}(x) = \begin{cases} 1, & x \leq 90 \\ \frac{200 - x}{200 - 90}, & 90 < x \leq 200 \\ 0, & x > 200 \end{cases}$$

b. Himpunan Fuzzy Sedang

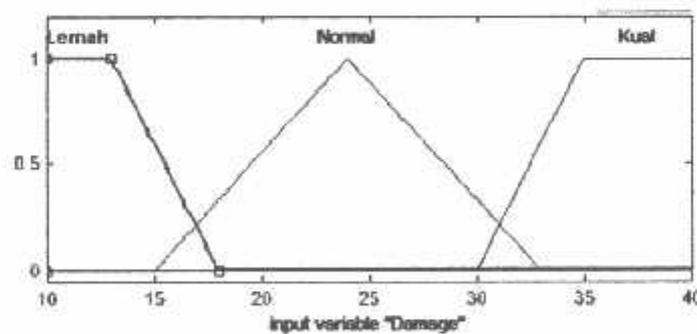
$$\mu_{Sedang}(x) = \begin{cases} 0, & x < 100 \text{ atau } x > 400 \\ \frac{x - 100}{250 - 100}, & 100 \leq x \leq 250 \\ \frac{400 - x}{400 - 250}, & 250 < x \leq 400 \end{cases}$$

c. Himpunan Fuzzy Banyak

$$\mu_{Banyak}(x) = \begin{cases} 0, & x < 300 \\ \frac{x - 300}{400 - 300}, & 300 \leq x < 400 \\ 1, & x \geq 400 \end{cases}$$

#### 4. Variabel *Damage*

Variabel *Damage* merupakan nilai *damage* milik *Creep*. Variabel ini terbagi menjadi tiga himpunan fuzzy yaitu : Lemah, Normal, Kuat. Berikut ini fungsi keanggotaan untuk variabel *damage* pada Gambar 3.12.

Gambar 3.12 Fungsi Keanggotaan Variabel *Damage*

a. Himpunan Fuzzy Lemah

$$\mu_{Lemah}(x) = \begin{cases} 1, & x \leq 13 \\ \frac{18-x}{18-13}, & 13 < x \leq 18 \\ 0, & x > 18 \end{cases}$$

b. Himpunan Fuzzy Normal

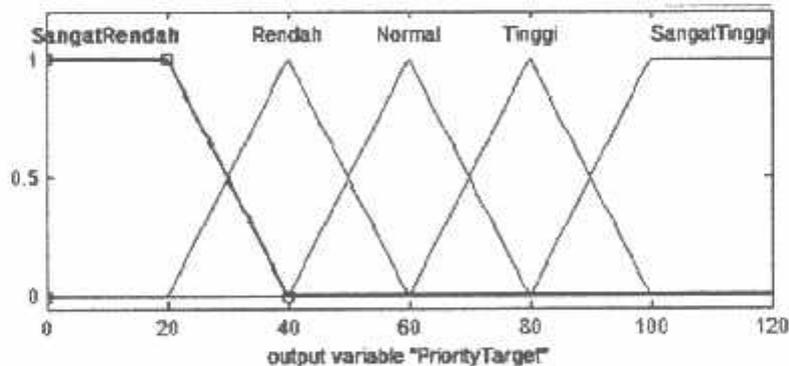
$$\mu_{Normal}(x) = \begin{cases} 0, & x < 15 \text{ atau } x > 33 \\ \frac{x-15}{24-15}, & 15 \leq x \leq 24 \\ \frac{33-x}{33-24}, & 24 < x \leq 33 \end{cases}$$

c. Himpunan Fuzzy Kuat

$$\mu_{Kuat}(x) = \begin{cases} 0, & x < 30 \\ \frac{x-30}{35-30}, & 30 \leq x < 35 \\ 1, & x \geq 35 \end{cases}$$

##### 5. Variabel *Priority Target*

Untuk variabel *Priority Target*, penulis mengadopsi hasil penelitian sebelumnya yang dilakukan oleh (Satriawan, 2014), yang menyatakan bahwa “Variabel *Priority Target* merupakan variabel *output*, nilai dari variabel ini menjadi acuan *NPC Tower* untuk menentukan target tembak. Variabel *Priority Target* memiliki lima himpunan fuzzy, yaitu : sangat rendah, rendah, normal, tinggi dan sangat tinggi”. Pada Gambar 3.13 untuk lebih jelasnya.



Gambar 3.13 Fungsi Keanggotaan Variabel *Priority Target*

a. Himpunan Fuzzy Sangat Rendah

$$\mu_{SangatRendah}(x) = \begin{cases} 1, & x \leq 20 \\ \frac{40-x}{40-20}, & 20 < x \leq 40 \\ 0, & x > 40 \end{cases}$$

b. Himpunan Fuzzy Rendah

$$\mu_{Rendah}(x) = \begin{cases} 0, & x < 20 \text{ atau } x > 60 \\ \frac{x-20}{40-20}, & 20 \leq x \leq 40 \\ \frac{60-x}{60-40}, & 40 < x \leq 60 \end{cases}$$

c. Himpunan Fuzzy Normal

$$\mu_{Normal}(x) = \begin{cases} 0, & x < 40 \text{ atau } x > 80 \\ \frac{x-40}{60-40}, & 40 \leq x \leq 60 \\ \frac{80-x}{80-60}, & 60 < x \leq 80 \end{cases}$$

d. Himpunan Fuzzy Tinggi

$$\mu_{Tinggi}(x) = \begin{cases} 0, & x < 60 \text{ atau } x > 100 \\ \frac{x-60}{80-60}, & 60 \leq x \leq 80 \\ \frac{100-x}{100-80}, & 80 < x \leq 100 \end{cases}$$

e. Himpunan Fuzzy Sangat Tinggi

$$\mu_{SangatTinggi}(x) = \begin{cases} 0, & x < 80 \\ \frac{x-80}{100-80}, & 80 \leq x < 100 \\ 1, & x \geq 100 \end{cases}$$

### 3.6 Rule Fuzzy

Aturan yang diterapkan pada logika fuzzy berdasarkan keterkaitan antar *set fuzzy*. *Rule fuzzy* menentukan besaran derajat keanggotaan set fuzzy. *Rule fuzzy* yang digunakan dari atribut *creep*. Untuk *rule fuzzy* pada Tabel 3.4.

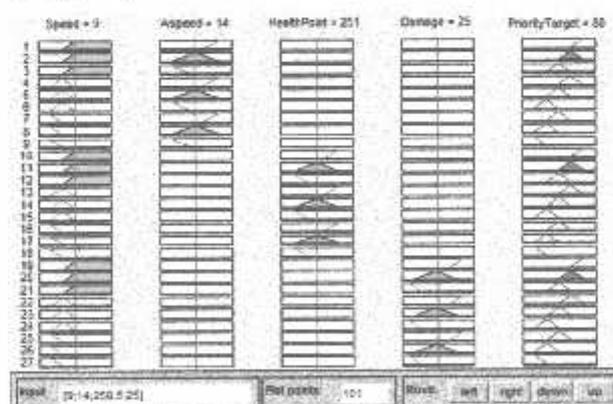
Tabel 3.4 Tabel *Rule Fuzzy*

ANTESEDEN				KONSEKUEN	
Operator	Speed	Operator	Aspeed	Operator	Prioritas
IF	Cepat	AND	Cepat	THEN	Sangat Tinggi
			Normal		Tinggi
			Lambat		Normal
	Normal		Cepat		Tinggi
			Normal		Normal

ANTESEDEN				KONSEKUEN				
Operator	Speed	Operator	Aspeed	Operator	Prioritas			
	Lambat		Lambat		Rendah			
			Cepat		Normal			
			Normal		Rendah			
			Lambat		Sangat Rendah			
ANTESEDEN				KONSEKUEN				
Operator	Speed	Operator	Health Point	Operator	Prioritas			
IF	Cepat	AND	Banyak	THEN	Sangat Tinggi			
			Sedang		Tinggi			
			Sedikit		Normal			
	Normal		Banyak		Tinggi			
			Sedang		Normal			
			Sedikit		Rendah			
	Lambat		Banyak		Normal			
			Sedang		Rendah			
			Sedikit		Sangat Rendah			
	ANTESEDEN				KONSEKUEN			
	Operator		Speed		Operator	Damage	Operator	Prioritas
	IF		Cepat		AND	Kuat	THEN	Sangat Tinggi
Normal		Tinggi						
Lemah		Normal						
Normal		Kuat	Tinggi					
		Normal	Normal					
		Lemah	Rendah					
Lambat		Kuat	Normal					
		Normal	Rendah					
		Lemah	Sangat Rendah					

### 3.7 Analisis Menggunakan Fuzzy Toolbox Matlab

Analisis perhitungan menggunakan fuzzy tool box matlab versi 2014.  $Speed = 9$ ,  $Aspeed = 14$ ,  $Health Point = 251$ , dan  $Damage = 25$ . Menghasilkan  $Priority Target$  sebesar 80. Seperti pada Gambar 3.14.



Gambar 3.14 Analisis menggunakan Fuzzy Tool box Matlab

## BAB IV

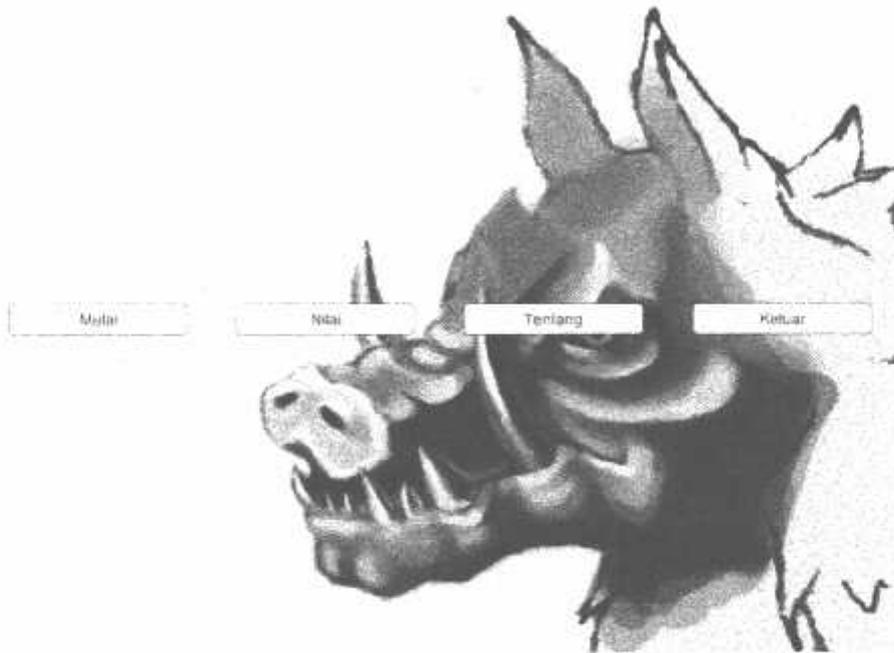
### IMPLEMENTASI DAN PENGUJIAN

#### 4.1 Implementasi

Implementasi adalah proses menerapkan rancangan *game* yang telah dibuat. Pada sub bab ini, menjelaskan tentang fitur – fitur yang terdapat pada *game*.

##### 4.1.1 Implementasi *Scene* Menu

Terdapat 4 tombol yaitu “Mulai”, “Nilai”, “Tentang”, “Keluar”. Seperti pada Gambar 4.1.



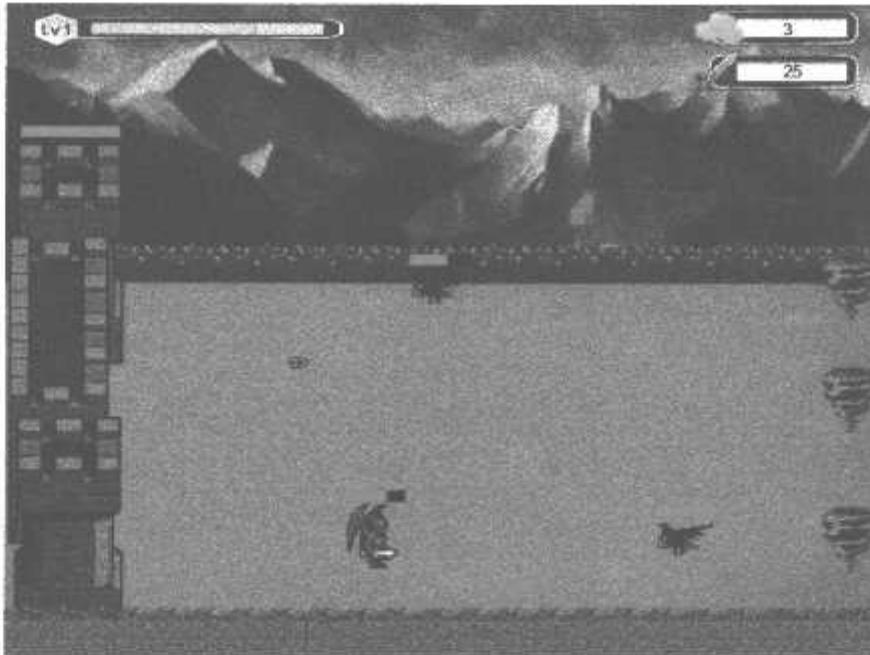
Gambar 4.1 *Scene* Menu

##### 4.1.2 Implementasi *Scene* Mulai

*Scene* Mulai adalah *scene* yang dapat dimainkan oleh pemain. Terdapat 3 level yaitu level 1, 2, dan terakhir level 3.

###### a. Level 1

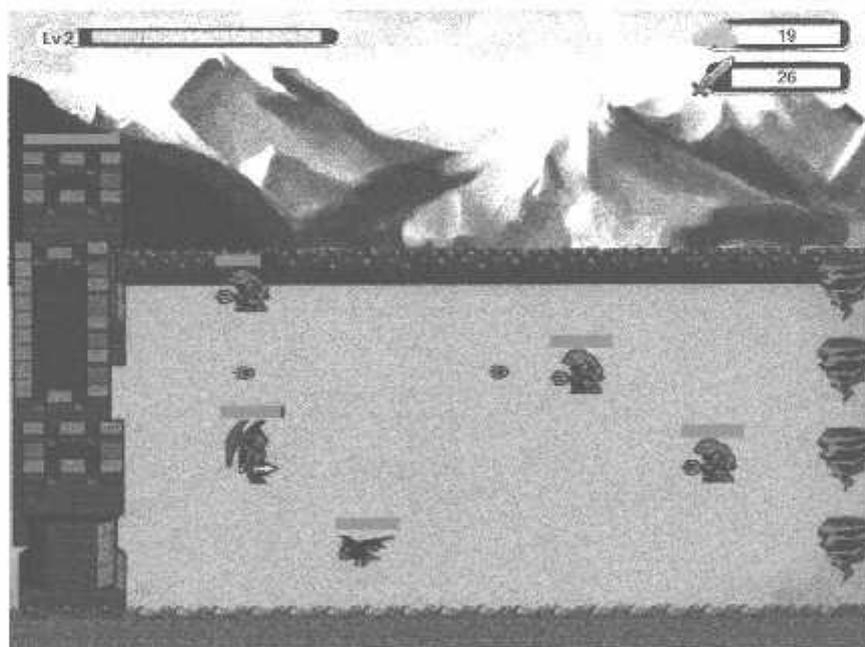
Pada level 1 terdapat 3 *hole spawn* musuh seperti pada Gambar 4.2.



Gambar 4.2 Scene Mulai Level 1

b. Level 2

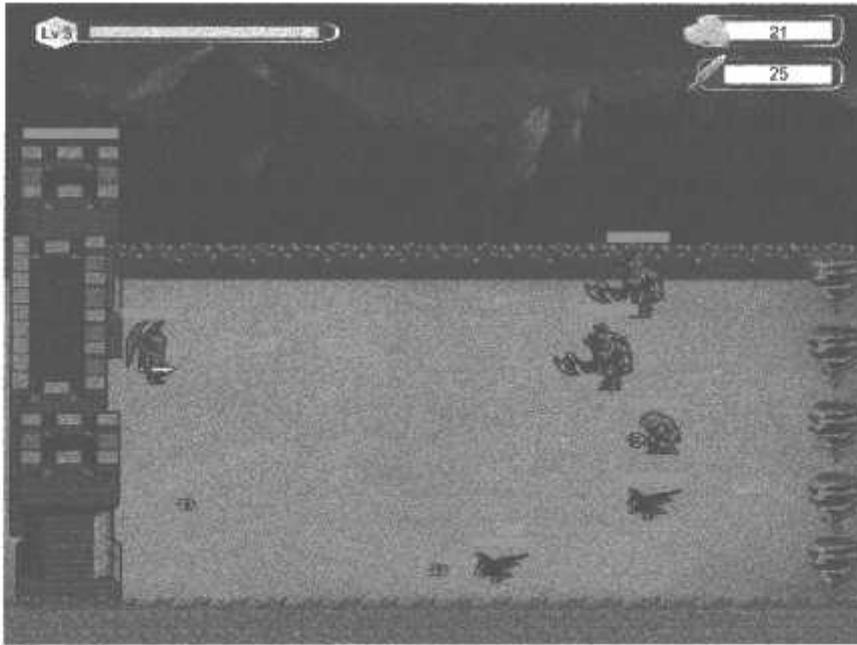
Pada level 2 terdapat 4 *hole spawn* musuh seperti pada Gambar 4.3.



Gambar 4.3 Scene Mulai Level 2

c. Level 3

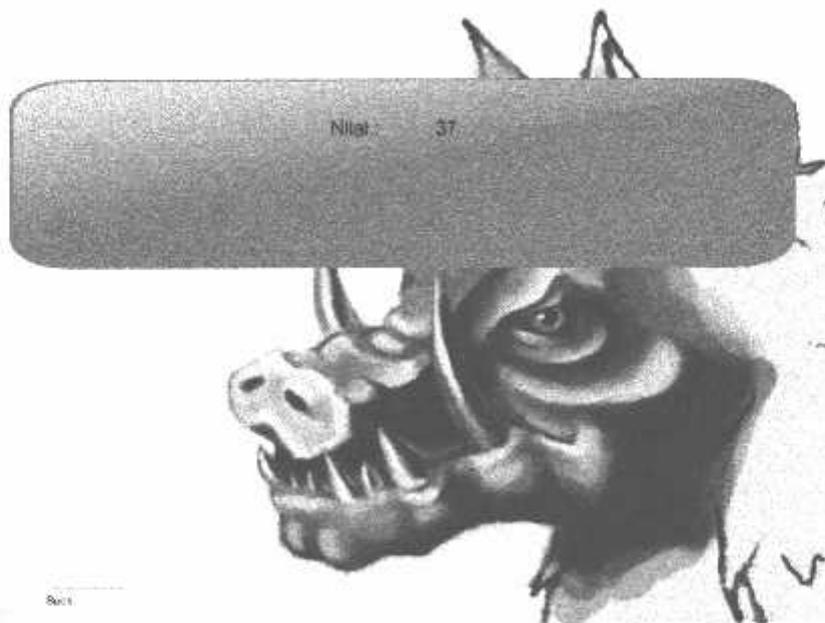
Pada level 3 terdapat 5 *hole spawn* musuh seperti pada Gambar 4.4.



Gambar 4.4 *Scene* Mulai Level 3

#### 4.1.3 Implementasi *Scene* Nilai

*Scene* Nilai menampilkan nilai terakhir pada permainan seperti pada Gambar 4.5.



Gambar 4.5 *Scene* Nilai

#### 4.1.4 Implementasi *Scene* Tentang

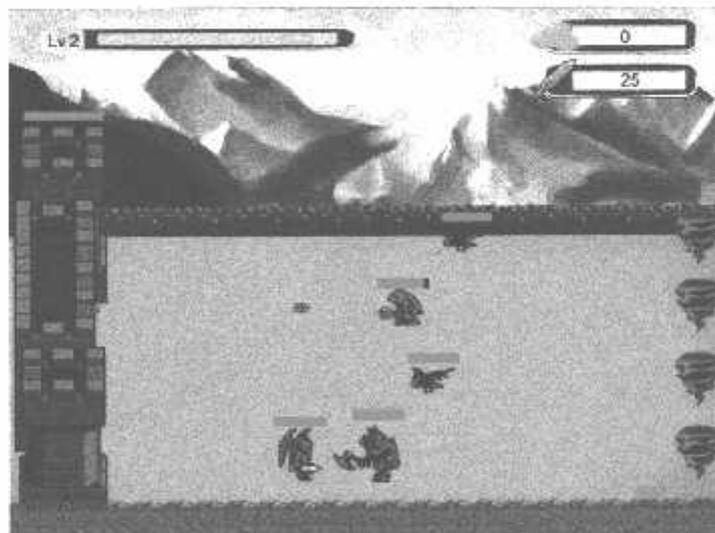
*Scene* Tentang berisi informasi penulis dan informasi game seperti pada Gambar 4.6.



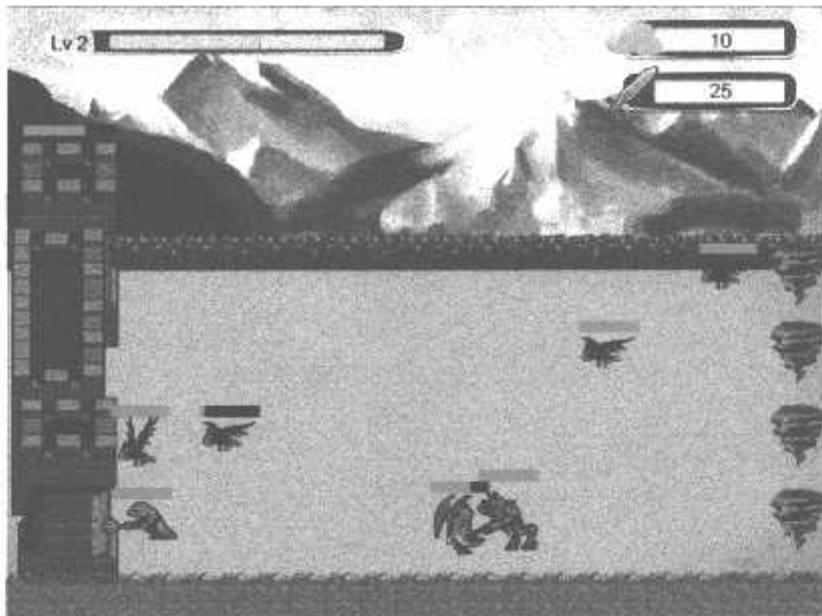
Gambar 4.6 *Scene* Tentang

#### 4.1.5 Ketika *Creep* Menyerang

Gambar 4.7 adalah Gambar sebelum *creep* menyerang. Gambar 4.8 nyawa *Hero* akan berkurang ketika *creep* menyerang.



Gambar 4.7 *Creep* sebelum menyerang



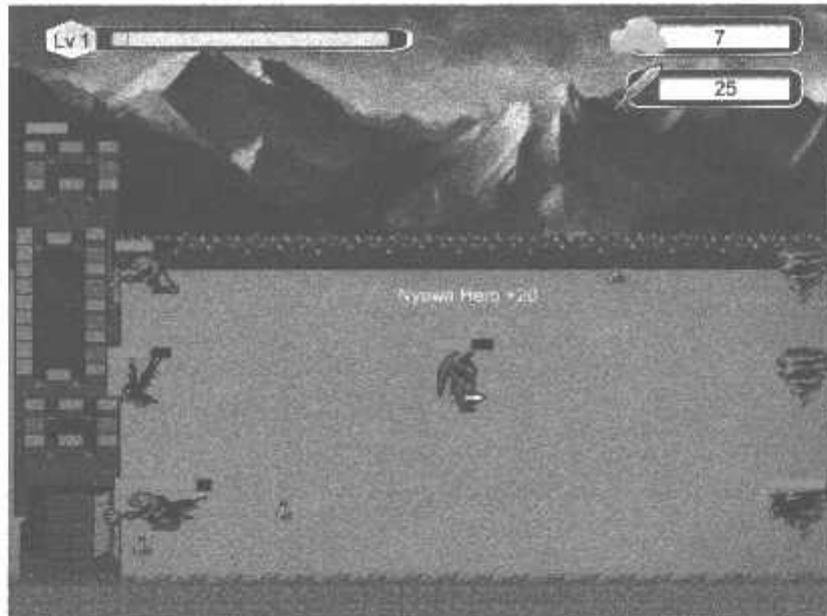
Gambar 4.8 *Creep* sedang menyerang

#### 4.1.6 Potion

Setelah berhasil membunuh *creep*, *potion* akan jatuh dari *creep*. Untuk menggunakan *potion*, *Hero* harus mengambilnya. Pada Gambar 4.9 terdapat 4 *potion* yang jatuh. Gambar 4.10 *Hero* sesaat setelah mengambil *red potion*. Manfaat *red potion* adalah menambah nyawa *Hero* sebesar 20 *Health Point*.



Gambar 4.9 *Potion* yang jatuh



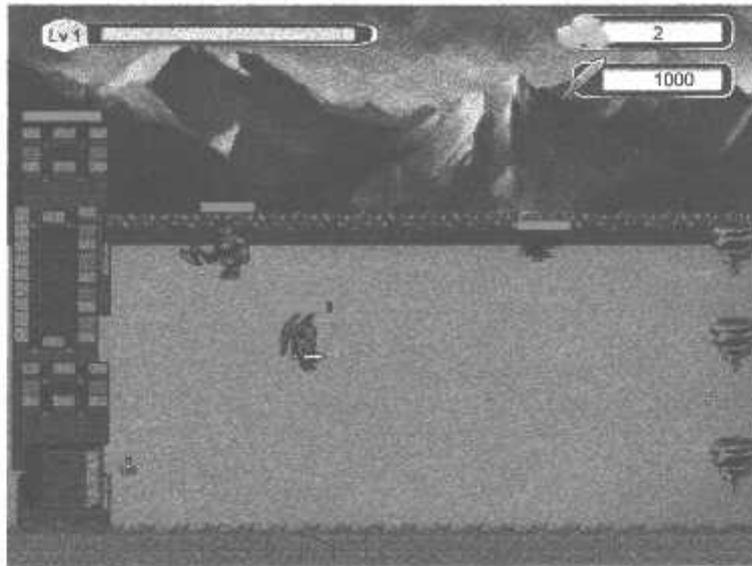
Gambar 4.10 Hero mengambil red potion

#### 4.1.7 Fitur *Pause* dan *Cheat*

Setelah tombol ESC ditekan fitur *pause* dan *cheat* akan aktif seperti Gambar 4.11. Penerapan *cheat*, setelah menuliskan "inc damage" tanpa tanda kutip pada kolom *cheat*, maka *damage hero* akan bertambah menjadi 1000 seperti Gambar 4.12.



Gambar 4.11 Fitur *Pause* dan *Cheat*



Gambar 4.12 Hasil *cheat* menambah *damage*

## 4.2 PENGUJIAN

### 4.2.1 Pengujian Perhitungan Logika Fuzzy

Pengujian perhitungan logika fuzzy, penulis membandingkan hasil logika fuzzy pada *game* dengan perhitungan matematis dan hasil logika fuzzy pada *Matlab* yang dapat dibaca pada Tabel 4.1.

Tabel 4.1 Pengujian Perhitungan

Percobaan	Atribut	Priority Target		
		Game Tower Defence	Perhitungan Matematis	Matlab
1	Speed = 7 Aspeed = 10 Health Point = 474 Damage = 39	37.78	37.78	32.9
2	Speed = 6 Aspeed = 17 Health Point = 410 Damage = 18	55	55	53.3
3	Speed = 10 Aspeed = 12 Health Point = 68 Damage = 37	81.78	81.78	86.4
4	Speed = 8 Aspeed = 17 Health Point = 370 Damage = 10	56.92	56.92	59.1
5	Speed = 8 Aspeed = 14 Health Point = 94 Damage = 13	50	50	50

Perhitungan logika fuzzy secara matematis pada percobaan ke satu adalah sebagai berikut :

**Tahap 1 : Fuzzifikasi**

Speed (7)

$$Lambat = 1$$

$$Normal = 0$$

$$Cepat = 0$$

Aspeed (10)

$$Lambat = 1$$

$$Normal = 0$$

$$Cepat = 0$$

Health Point (474)

$$Sedikit = 0$$

$$Sedang = 0$$

$$Banyak = 1$$

Damage (39)

$$Lemah = 0$$

$$Normal = 0$$

$$Kuat = 1$$

**Tahap 2 : Pembentukan Rule**

Pembentukan rule mengacu pada Tabel 3.4.

**Tahap 3 : Inferensi**

IF Speed IS Lambat(1) AND Aspeed IS Lambat(1) THEN PriorityTarget IS SangatRendah(1);

IF Speed IS Lambat(1) AND HealthPoint IS Banyak(1) THEN PriorityTarget IS Normal(1);

IF Speed IS Lambat(1) AND Damage IS Kuat(1) THEN PriorityTarget IS Normal(1);

SangatRendah (1); Rendah (0); Normal (1); Tinggi (0); SangatTinggi (0)

**Tahap 4 : Defuzzyfikasi**

$$PriorityTarget = \frac{\sum x \times \mu}{\sum \mu} = \frac{(15.56 \times 1) + (60 \times 1)}{1 + 1} = 37.78$$

Berdasarkan Tabel 4.1, hasil dari perhitungan fuzzy pada game dan perhitungan matematis adalah sama. Untuk hasil dari perhitungan fuzzy pada game

dan hasil fuzzy pada *matlab* berbeda, dikarenakan perbedaan perhitungan nilai  $x$  pada rumus defuzzyfikasi ( $\frac{\sum \mu \times x}{\sum \mu}$ ) pada *library DotFuzzy* dan *Matlab*.

#### 4.2.2 Pengujian Tower Menembak *Creep* Berdasarkan *Priority Target*

Pengujian ini dilakukan pada *game* yang sedang berjalan. Pengujian ini bertujuan untuk mengetahui apakah tower dapat menembak *Creep* berdasarkan *Priority Target* atau tidak. Hasil dari pengujian dapat dibaca pada Tabel 4.2.

Tabel 4.2 Pengujian Tower Menembak *Creep*

Perco- baan	Unit Musuh	Priorit y Target (PT)	Yang Diharapkan	Yang Dicapai	Kesim- pulan
1	<b>Creep 1</b> Speed = 7 Aspeed = 10 Health Point = 474 Damage = 39	37.78	Tower menembak Creep1 hingga mati.	Tower menembak Creep1 hingga mati	Berhasil
2	<b>Creep 1</b> Speed = 7 Aspeed = 10 Health Point = 474 Damage = 39	37.78	Tower akan menembak creep dengan urutan tembak Creep2 ditembak terlebih dahulu sebanyak 6 kali, setelah itu tower menembak Creep1 hingga mati. Dilanjutkan Tower menembak Creep2 hingga mati.	Tower menembak dengan urutan Creep1, Creep2	Berhasil
	<b>Creep 2</b> Speed = 6 Aspeed = 17 Health Point = 410 Damage = 18	55			
3	<b>Creep 1</b> Speed = 7 Aspeed = 10 Health Point = 474 Damage = 39	37.78	Tower akan menembak creep dengan urutan tembak Creep3 ditembak terlebih dahulu hingga mati. Setelah itu menembak Creep2 sebanyak 6 kali, setelah itu tower menembak Creep1 hingga mati. Dilanjutkan Tower menembak Creep2 hingga mati.	Tower menembak dengan urutan Creep3, Creep2, Creep1, Creep2	Berhasil
	<b>Creep 2</b> Speed = 6 Aspeed = 17 Health Point = 410 Damage = 18	55			
	<b>Creep 3</b> Speed = 10 Aspeed = 12 Health Point = 68 Damage = 37	81.78			
4	<b>Creep 1</b> Speed = 7 Aspeed = 10	37.78	Tower akan menembak creep	Tower menembak	Berhasil

Percobaan	Unit Musuh	Prioritas Target (PT)	Yang Diharapkan	Yang Dicapai	Kesimpulan
	Health Point = 474 Damage = 39				
	<b>Creep 2</b> Speed = 6 Aspeed = 17 Health Point = 410 Damage = 18	55	dengan urutan tembak Creep3 ditembak terlebih dahulu hingga mati. Setelah itu menembak Creep4 hingga mati.	dengan urutan Creep3, Creep4, Creep2, Creep1, Creep2	
	<b>Creep 3</b> Speed = 10 Aspeed = 12 Health Point = 68 Damage = 37	81.78	Setelah itu menembak Creep2 sebanyak 6 kali, setelah itu tower menembak Creep1 hingga mati.		
	<b>Creep 4</b> Speed = 8 Aspeed = 17 Health Point = 370 Damage = 10	56.92	Dilanjutkan Tower menembak Creep2 hingga mati.		

Kesimpulan berdasarkan Tabel 4.2 adalah Tower dapat menembak *Creep* berdasarkan *Priority Target* dengan tingkat keberhasilan adalah

$$\text{Keberhasilan} = \frac{4}{4} \times 100\% = 100\%$$

#### 4.2.3 Pengujian Tower Menggunakan Logika Fuzzy dan Tanpa Logika Fuzzy

Pengujian ini dilakukan untuk mengetahui seberapa efektif logika fuzzy mamdani yang diterapkan pada Tower di *Game Tower Defence* dibandingkan dengan tower yang tidak menggunakan logika fuzzy mamdani. Dan hasilnya pada Tabel 4.3.

Tabel 4.3 Pengujian Tower Menggunakan Fuzzy Mamdani dan Tanpa Fuzzy Mamdani

	Tower menggunakan Logika Fuzzy	Tower tanpa menggunakan Logika Fuzzy
Health Point Tower	3000	3000
Jumlah Creep Terbunuh	3	3
Tower Lama Bertahan (detik)	56	53
Prosentase Lama Tower Bertahan	$\frac{(56 - 53)}{53} \times 100\% = 5.7\%$	

Kesimpulan berdasarkan Tabel 4.3 adalah tower yang menggunakan fuzzy mamdani dan tower yang tidak menggunakan fuzzy mamdani dapat membunuh

*creep* dengan jumlah yang sama, yaitu 3 *creep*. Tower yang menggunakan fuzzy mamdani, 5,7% dapat bertahan hidup lebih lama daripada tower tanpa fuzzy mamdani. Tower yang menggunakan fuzzy akan menembak berdasarkan *target priority* sedangkan tower yang tidak menggunakan fuzzy akan menembak berdasarkan urutan *creep* masuk ke area tembak tower.

---

## BAB V

### PENUTUP

#### 5.1 KESIMPULAN

Kesimpulan dari penelitian ini adalah sebagai berikut :

1. Hasil dari perhitungan logika fuzzy pada game dibandingkan dengan perhitungan logika fuzzy pada perhitungan matematis adalah sama.
2. Terdapat perbedaan hasil perhitungan logika fuzzy pada *game* dibandingkan dengan perhitungan fuzzy pada *Matlab*. Hal ini dikarenakan perbedaan perhitungan nilai  $x$  pada rumus defuzzyfikasi ( $\frac{\sum \mu \times x}{\sum \mu}$ ) pada *library Dot'uzzy* dan *Matlab*.
3. Tower dapat menembak *creep* berdasarkan *Priority Target* dengan tingkat keberhasilan 100%.
4. Tower yang menggunakan fuzzy mamdani dan tower yang tidak menggunakan fuzzy mamdani dapat membunuh *creep* dengan jumlah yang sama, yaitu 3 *creep*.
5. Tower yang menggunakan fuzzy mamdani, 5,7% dapat bertahan hidup lebih lama daripada tower tanpa fuzzy mamdani.
6. Tower yang menggunakan fuzzy mamdani akan menembak berdasarkan *target priority* sedangkan tower yang tidak menggunakan fuzzy mamdani akan menembak berdasarkan urutan *creep* masuk ke area tembak tower.

#### 5.2 SARAN

Saran untuk pengembangan lebih lanjut adalah :

1. Menambah kecerdasan buatan *Path Finding*, misal  $A^*$ , pada navigasi *Creep*.
2. Menjadikan Game dapat dimainkan secara *multiplayer*, layaknya *Game Dota*.

## DAFTAR PUSTAKA

- Bertoli, M. (2014, Januari 14). *DotFuzzy*. Diakses pada November 7, 2015, tersedia pada DotFuzzy: <https://github.com/MicheleBertoli/DotFuzzy>
- Cordon, O., Herrera, F., Hoffmann, F., & Magdalena, L. (2001). *Genetic Fuzzy Systems Evolutionary Tuning And Learning Of Fuzzy Knowledge Bases*. Singapore: World Scientific.
- Dota 2 Wiki*. (2015, Oktober 21). Diakses pada November 30, 2015, from Game Pedia: <http://dota2.gamepedia.com/Aggro>
- Fuzzy Logic Toolbox*. (2015). Diakses pada November 30, 2015, from Mathworks: <http://www.mathworks.com/products/fuzzy-logic/index.html>
- Lee, K. H. (2005). *First Course on Fuzzy Theory and Applications*. Berlin: Springer.
- Roedavan, R. (2014). *Unity Tutorial Game Engine*. Bandung: Informatika.
- Satriawan, S. (2014). *Penerapan Logika Fuzzy Untuk Optimalisasi Perilaku Tower Pada Game Tower Defense*. Universitas Komputer Indonesia.
- Sutojo, T., Mulyanto, F., & Suhartono, V. (2011). *Kecerdasan Buatan*. Yogyakarta: ANDI.
- Suyanto. (2012). *Intelijensia Buatan*. Bandung.
- Unity3d*. (2015). Diakses pada November 30, 2015, tersedia pada Unity3d: <http://docs.unity3d.com/Manual/>

## LAMPIRAN

---

**BERITA ACARA UJIAN SKRIPSI**  
**FAKULTAS TEKNOLOGI INDUSTRI**

Nama : Muhammad Nur Hadi  
NIM : 12.18.213  
Program Studi: Teknik Informatika S-1  
Judul : Penerapan Logika Fuzzy pada Tower di *Game Tower Defence*

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari : Jum'at  
Tanggal : 15 Januari 2016  
Nilai : 92.47 (A)

Panitia Ujian Skripsi :

**Ketua Majelis Penguji**



**Joseph Dedy Irawan, ST, MT**  
**NIP. 197404162005011002**

Anggota Penguji :

**Dosen Penguji I**



**Sandy Nataly Mantia, S.Kom**  
**NIP.P.1030800418**

**Dosen Penguji II**



**Hani Zulfia Zahro', S.Kom, M.Kom**  
**NIP.P. 1031500480**

## FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata I Program Studi Teknik Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa:

NAMA : Muhammad Nur Hadi  
NIM : 1218213  
JURUSAN : Teknik Informatika S-1  
JUDUL : Penerapan Logika Fuzzy pada Tower di Game Tower Defence

No	Penguji	Tanggal	Uraian	Paraf
1.	Penguji I	15 Januari 2016	1. Perbaiki penulisan pada abstrak 2. Perbaikan penutup	
2.	Penguji II	15 Januari 2016	1. Perbaiki penulisan. Sitasi, halaman pada bab, huruf kapital pada gambar dan tabel 2. Penambahan cheat pada game	

Dosen Penguji I



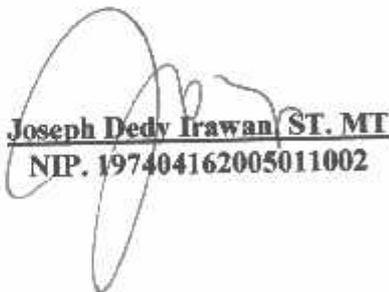
Sandy Nataly Mantia, S.Kom  
NIP.P.1030800418

Dosen Penguji II



Hani Zulfia Zahro', S.Kom, M.Kom  
NIP.P. 1031500480

Dosen Pembimbing I



Joseph Dedy Irawan, ST. MT  
NIP. 197404162005011002

Dosen Pembimbing II



Sonny Prasctio, ST. MT  
NIP. 1031000433



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG  
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI  
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN  
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG  
BANK NIAGA MALANG

Kampus I : J. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145  
Kampus II : J. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 23 Oktober 2015

Nomor : ITN-593/L.INF/TA/2015

Lampiran : ---

Perihal : Bimbingan Skripsi

Kepada : Yth. Bpk/Ibu Sonny prasetio ST,MT  
Dosen Pembina Program Studi Teknik Informatika S-1  
Institut Teknologi Nasional  
Malang

Dengan Hormat,  
Sesuai dengan permohonan dan persetujuan dalam proposal skripsi untuk mahasiswa :

Nama : MUHAMMAD NUR HADI  
Nim : 1218213  
Prodi : Teknik Informatika S-1  
Fakultas : Teknologi Industri

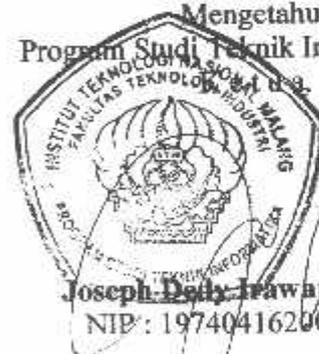
Maka dengan ini pembimbingan kami serahkan sepenuhnya kepada Saudara/i selama waktu 6 (enam) bulan, terhitung mulai tanggal :

**23 Oktober 2015 S/D 23 Maret 2016**

Sebagai satu syarat untuk menempuh Ujian Akhir Sarjana Teknik, Program Studi Teknik Informatika S-1.

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui  
Program Studi Teknik Informatika S-1



Joseph Dedy Irawan, ST., MT.  
NIP : 197404162905021002

Form 5-4a

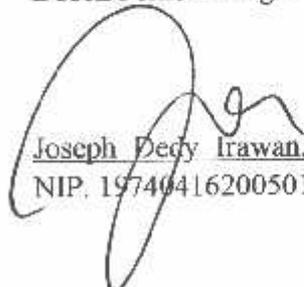
## Lampiran 2 : Lembar asistensi dosen pembimbing I

### LEMBAR BIMBINGAN SKRIPSI

Nama : Muhammad Nur Hadi  
NIM : 1218213  
Masa Bimbingan : 23 Oktober 2015 s/d 23 Marct 2016  
Judul Skripsi : Penerapan Logika Fuzzy Pada Game Di Game Tower Defence

No	Tanggal	Uraian	Paraf
1	4 November 2015	ACC Design Game.	
2	7 November 2015	Bab I Revisi: Penulisan latar belakang dan batasan masalah.	
3	9 November 2015	Bab I: OK. Bab II revisi: Teori logika fuzzy diperjelas, terutama tentang mamdani.	
4	14 November 2015	Bab II: OK. Bab III revisi: Perancangan menggunakan matlab. Dan perhitungan matematis logika fuzzy mamdani.	
5	25 November 2015	ACC seminar progres pada 28 November 2015.	
6	5 Desember 2015	Bab III: OK. Demo Game revisi: penambahan health bar dan animasi saat menyerang musuh.	
7	10 Desember 2015	Demo Game: OK. Bab IV: revisi: Penjelasan fitur – fitur yang ada pada game dan screenshot-nya.	
8	11 Desember 2015	ACC seminar hasil pada 14 Desember 2015.	
9	12 Januari 2016	Bab IV dan Bab V: OK. Pembedulan daftar isi, daftar gambar, daftar tabel. ACC kompre pada 15 Januari 2016.	

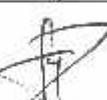
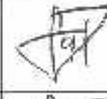
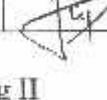
Dosen Pembimbing I

  
Joseph Decy Irawan, ST, MT  
NIP. 197404162005011002

## Lampiran 2 : Lembar asistensi dosen pembimbing 2

### LEMBAR BIMBINGAN SKRIPSI

Nama : Muhammad Nur Hadi  
NIM : 1218213  
Masa Bimbingan : 23 Oktober 2015 s/d 23 Maret 2016  
Judul Skripsi : Penerapan Logika Fuzzy Pada Game Di Game Tower Defence

No	Tanggal	Uraian	Paraf
1	9 November 2015	Revisi Bab III Perancangan perhitungan nilai dan atribut player dan NPC.	
2	21 November 2015	ACC Bab III.	
3	23 November 2015	Bab I revisi: Tambahkan latar belakang dibagian "kenapa game tower defence dan contoh pada game umumnya"	
4	25 November 2015	ACC seminar progres pada 28 November 2015.	
5	8 Desember 2015	Bab II revisi: Tambahkan tinjauan pustaka tentang game tower defence dan tambahkan sitasi.	
6	9 Desember 2015	Bab I: ACC Bab II: ACC	
7	10 Desember 2015	Bab IV: ACC. Bab V revisi : Kesimpulan berdasarkan hasil pengujian dan rumusan masalah.	
8	11 Desember 2015	ACC seminar hasil pada 14 Desember 2015.	
9	22 Desember 2015	Revisi: Penomoran gambar dan tabel disetiap bab.	
10	4 Januari 2016	Revisi daftar pustaka dan sitasi.	
11	5 Januari 2016	ACC Kompre pada 15 Januari 2016	

Dosen Pembimbing II



Sonny Prasetio, S.T. MT  
NIP.P. 1031000433

## Lampiran Source Code Class PenerapanFuzzy.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using DotFuzzy;

namespace DotFuzzy
{
    public class PenerapanFuzzy
    {
        FuzzyEngine fuzzyEngine;

        public PenerapanFuzzy(double _speed, double _aspeed,
double _healthPoint, double _damage)
        {

            //----- INPUT -----//

                LinguisticVariable speed = new
LinguisticVariable("Speed");
                speed.MembershipFunctionCollection.Add(new
MembershipFunction("Lambat", 6, 6, 7, 8));
                speed.MembershipFunctionCollection.Add(new
MembershipFunction("Normal", 7, 8, 8, 9));
                speed.MembershipFunctionCollection.Add(new
MembershipFunction("Cepat", 8, 9, 12, 12));

                LinguisticVariable aspeed = new
LinguisticVariable("Aspeed");
                aspeed.MembershipFunctionCollection.Add(new
MembershipFunction("Lambat", 8, 8, 10, 13));
                aspeed.MembershipFunctionCollection.Add(new
MembershipFunction("Normal", 10, 14, 14, 18));
                aspeed.MembershipFunctionCollection.Add(new
MembershipFunction("Cepat", 15, 18, 20, 20));

                LinguisticVariable healthPoint = new
LinguisticVariable("HealthPoint");
                healthPoint.MembershipFunctionCollection.Add(new
MembershipFunction("Sedikit", 1, 1, 90, 200));
                healthPoint.MembershipFunctionCollection.Add(new
MembershipFunction("Sedang", 100, 250, 250, 400));
                healthPoint.MembershipFunctionCollection.Add(new
MembershipFunction("Banyak", 300, 400, 500, 500));

                LinguisticVariable damage = new
LinguisticVariable("Damage");
                damage.MembershipFunctionCollection.Add(new
MembershipFunction("Lemah", 10, 10, 13, 18));
                damage.MembershipFunctionCollection.Add(new
MembershipFunction("Normal", 15, 24, 24, 33));
                damage.MembershipFunctionCollection.Add(new
MembershipFunction("Kuat", 30, 35, 40, 40));
        }
    }
}
```

```

//===== OUTPUT =====//

LinguisticVariable priorityTarget = new
LinguisticVariable("PriorityTarget");

    priorityTarget.MembershipFunctionCollection.Add(new
MembershipFunction("SangatRendah", 0, 0, 20, 40));
    priorityTarget.MembershipFunctionCollection.Add(new
MembershipFunction("Rendah", 20, 40, 40, 60));
    priorityTarget.MembershipFunctionCollection.Add(new
MembershipFunction("Normal", 40, 60, 60, 80));
    priorityTarget.MembershipFunctionCollection.Add(new
MembershipFunction("Tinggi", 60, 80, 80, 100));
    priorityTarget.MembershipFunctionCollection.Add(new
MembershipFunction("SangatTinggi", 80, 100, 120, 120));

//===== PROCESS =====//
FuzzyEngine = new FuzzyEngine();

    fuzzyEngine.LinguisticVariableCollection.Add(speed);
fuzzyEngine.LinguisticVariableCollection.Add(aseed);
fuzzyEngine.LinguisticVariableCollection.Add(healthPoint);
fuzzyEngine.LinguisticVariableCollection.Add(damage);
fuzzyEngine.LinguisticVariableCollection.Add(priorityTarget);

    fuzzyEngine.Consequent = "PriorityTarget";

    ////////////////////////////////////////
    //////////////////////////////////////// SATU ////////////////////////////////////////
    ////////////////////////////////////////
    //speed cepat
    fuzzyEngine.FuzzyRuleCollection.Add(new
FuzzyRule("IF (Speed IS Cepat) AND (Aspeed IS Cepat) THEN
PriorityTarget IS SangatTinggi"));
    fuzzyEngine.FuzzyRuleCollection.Add(new
FuzzyRule("IF (Speed IS Cepat) AND (Aspeed IS Normal) THEN
PriorityTarget IS Tinggi"));
    fuzzyEngine.FuzzyRuleCollection.Add(new
FuzzyRule("IF (Speed IS Cepat) AND (Aspeed IS Lambat) THEN
PriorityTarget IS Normal"));
    //speed normal
    fuzzyEngine.FuzzyRuleCollection.Add(new
FuzzyRule("IF (Speed IS Normal) AND (Aspeed IS Cepat) THEN
PriorityTarget IS Tinggi"));
    fuzzyEngine.FuzzyRuleCollection.Add(new
FuzzyRule("IF (Speed IS Normal) AND (Aspeed IS Normal) THEN
PriorityTarget IS Normal"));
    fuzzyEngine.FuzzyRuleCollection.Add(new
FuzzyRule("IF (Speed IS Normal) AND (Aspeed IS Lambat) THEN
PriorityTarget IS Rendah"));
    //speed lambat
    fuzzyEngine.FuzzyRuleCollection.Add(new
FuzzyRule("IF (Speed IS Lambat) AND (Aspeed IS Cepat) THEN
PriorityTarget IS Normal"));

```

```

        fuzzyEngine.FuzzyRuleCollection.Add(new
FuzzyRule("IF (Speed IS Lambat) AND (Aspeed IS Normal) THEN
PriorityTarget IS Rendah"));
        fuzzyEngine.FuzzyRuleCollection.Add(new
FuzzyRule("IF (Speed IS Lambat) AND (Aspeed IS Lambat) THEN
PriorityTarget IS SangatRendah"));

        //////////////////////////////////////
        ////////////////////////////////////// DUA //////////////////////////////////////
        //////////////////////////////////////
        //speed cepat
        fuzzyEngine.FuzzyRuleCollection.Add(new
FuzzyRule("IF (Speed IS Cepat) AND (HealthPoint IS Banyak) THEN
PriricityTarget IS SangatTinggi"));
        fuzzyEngine.FuzzyRuleCollection.Add(new
fuzzyRule("IF (Speed IS Cepat) AND (HealthPoint IS Sedang) THEN
PriorityTarget IS Tinggi"));
        fuzzyEngine.FuzzyRuleCollection.Add(new
FuzzyRule("IF (Speed IS Cepat) AND (HealthPoint IS Sedikit) THEN
PriorityTarget IS Normal"));
        //speed normal
        fuzzyEngine.FuzzyRuleCollection.Add(new
FuzzyRule("IF (Speed IS Normal) AND (HealthPoint IS Banyak) THEN
PriorityTarget IS Tinggi"));
        fuzzyEngine.FuzzyRuleCollection.Add(new
FuzzyRule("IF (Speed IS Normal) AND (HealthPoint IS Sedang) THEN
PriorityTarget IS Normal"));
        fuzzyEngine.FuzzyRuleCollection.Add(new
FuzzyRule("IF (Speed IS Normal) AND (HealthPoint IS Sedikit)
THEN PriorityTarget IS Rendah"));
        //speed lambat
        fuzzyEngine.FuzzyRuleCollection.Add(new
FuzzyRule("IF (Speed IS Lambat) AND (HealthPoint IS Banyak) THEN
PriorityTarget IS Normal"));
        fuzzyEngine.FuzzyRuleCollection.Add(new
FuzzyRule("IF (Speed IS Lambat) AND (HealthPoint IS Sedang) THEN
PriorityTarget IS Rendah"));
        fuzzyEngine.FuzzyRuleCollection.Add(new
FuzzyRule("IF (Speed IS Lambat) AND (HealthPoint IS Sedikit)
THEN PriorityTarget IS SangatRendah"));

        //////////////////////////////////////
        ////////////////////////////////////// TIGA //////////////////////////////////////
        //////////////////////////////////////
        //speed cepat
        fuzzyEngine.FuzzyRuleCollection.Add(new
FuzzyRule("IF (Speed IS Cepat) AND (Damage IS Kuat) THEN
PriorityTarget IS SangatTinggi"));
        fuzzyEngine.FuzzyRuleCollection.Add(new
FuzzyRule("IF (Speed IS Cepat) AND (Damage IS Normal) THEN
PriorityTarget IS Tinggi"));
        fuzzyEngine.FuzzyRuleCollection.Add(new
FuzzyRule("IF (Speed IS Cepat) AND (Damage IS Lemah) THEN
PriorityTarget IS Normal"));
        //speed normal
        fuzzyEngine.FuzzyRuleCollection.Add(new
FuzzyRule("IF (Speed IS Normal) AND (Damage IS Kuat) THEN
PriricityTarget IS Tinggi"));

```

```

        fuzzyEngine.FuzzyRuleCollection.Add(new
FuzzyRule("IF (Speed IS Normal) AND (Damage IS Normal) THEN
PriorityTarget IS Normal"));
        fuzzyEngine.FuzzyRuleCollection.Add(new
FuzzyRule("IF (Speed IS Normal) AND (Damage IS Lemah) THEN
PriorityTarget IS Rendah"));
        //speed lambat
        fuzzyEngine.FuzzyRuleCollection.Add(new
FuzzyRule("IF (Speed IS Lambat) AND (Damage IS Kuat) THEN
PriorityTarget IS Normal"));
        fuzzyEngine.FuzzyRuleCollection.Add(new
FuzzyRule("IF (Speed IS Lambat) AND (Damage IS Normal) THEN
PriorityTarget IS Rendah"));
        fuzzyEngine.FuzzyRuleCollection.Add(new
FuzzyRule("IF (Speed IS Lambat) AND (Damage IS Lemah) THEN
PriorityTarget IS SangatRendah"));

        //input
        speed.InputValue = _speed;
        aspeed.InputValue = _aspeed;
        healthPoint.InputValue = _healthPoint;
        damage.InputValue = _damage;

    }

    //mengambil nilai crisp
    public double getCrisp()
    {
        return fuzzyEngine.Defuzzify();
    }

    public string getName(int index){
        return fuzzyEngine.getName(index);
    }

    public double getValueKeabuan(int index)
    {
        return fuzzyEngine.getValue(index);
    }

    public double getCentroid(int index)
    {
        return fuzzyEngine.getCentroid(index);
    }

    public double getMiu(int index)
    {
        return fuzzyEngine.getMiu(index);
    }
}
}

```

## Source Code CollectionsCreep.cs

```
using System.Collections.Generic;
using UnityEngine;
using StatusCharacter.Creep;
using StatusCharacter.Tower;
using DotFuzzy;
namespace List
{
    public class CollectionsCreep : MonoBehaviour
    {
        public List<CreepDefuzzy> enemies = new
List<CreepDefuzzy>();
        public List<string> nameCreep = new List<string>();
        public List<double> targetPriority = new List<double>();
        private Defuzzyfi creep;
        public bool isFuzzy = true;
        public bool isAnyDie;
        public bool isAnyHurt;
        public bool isAnyEnter;
        void OnTriggerStay2D(Collider2D other)
        {
            if (isAnyDie || isAnyHurt || isAnyEnter) //triger
dari CreepStatus, saat ada creep yang mati atau ada creep yang
terluka
            {
                // jika menggunakan isFuzzy true maka tower akan
mengurutkan hasil fuzzy
                //ramun jika isFuzzy false maka tower tidak
mengurutkan hasil fuzzy (non fuzzy - mengurutkan berdasarkan
pertama kali masuk area tembak)
                if (isFuzzy)
                {
                    enemies.Sort((a, b) =>
b.TargetPriority.CompareTo(a.TargetPriority)); //desc sort
berdasarkan target priority
                }
                nameCreep.Clear();
                targetPriority.Clear();
                foreach (CreepDefuzzy musuh in enemies)
                {
                    if (!nameCreep.Contains(musuh.Name)) //jika
belum ada
                    {
                        nameCreep.Add(musuh.Name);
                    }
                    targetPriority.Add(musuh.TargetPriority);
                }
                isAnyDie = false;
                isAnyHurt = false;
                isAnyEnter = false;
            }
            creep = other.GetComponent<Defuzzyfi>();
        }
    }
}
```