ISSN (Print) : 1693-1173
ISSN (Online) : 2548-4028

IMPLEMENTASI LOGIKA *FUZZY* PADA PEMBUATAN KARAKTER MUSUH UNTUK *GAME SINGLE FIGHTER* BERPLATFORM ANDROID

Fajar Mitasari [1)], Wawan Laksito YS [2)], Sri Siswanti [3)]
[1,2,3)] Program Studi Teknik Informatika, STMIK Sinar Nusantara Surakarta
[1)]poonievam@gmail.com, [2)]wlaksito@yahoo.com, [3)]syswanty@gmail.com

Abstract

This adventure game tells the story of an elephant who struggles to save his son from enemy prisoners. Elephants should be able to beat all three enemies, namely monkeys, wolves, and kingkong to be able to find his son. In the course of fighting the enemy will be no relief in the form of bombs and weapons are power ups in the form of apples to increase the number of lives player. The behavior of the enemy characters in this game are designed with the application of fuzzy logic Sugeno method. Each enemy by artificial intelligence has a health status (lives), ammo (the number of stones to attack the player) and distance (distance to the player during the attack). This game has a rule of implementation Sugeno Fuzzy Logic method for calculating the value of output behavior of the enemy in attacking player. The system design of the game is using the storyboard that contains a description of the story of each scene and any component material that will be used. Then the algorithm implementation using the Unity game development software. For game development, programming language used is C # of Unity. The results of this study are the Application Game "Single Fighter" that has succeeded in publishing in the Google Play Store are made of Unity game engine, side-scrolling genre game, and the android platform. From the application of fuzzy logic in this game also has gone well, where there are 41 types of enemy attackers%, 36% type of archer-distance and 87% of type giant enemy very aggressive in attacking player.

Kata Kunci: Game, Logika Fuzzy, Sugeno, Unity, Android

I. PENDAHULUAN

Game merupakan salah satu sarana hiburan yang banyak diminati oleh masyarakat. Pada mulanya *game* hanya dimainkan di komputer dan *console* tetapi sekarang sudah memasuki era *mobile game*. *Game* yang akan dikembangkan ini berplatform Android yang dapat diunduh secara gratis melalui *Google Play Store*.

Game yang berjudul "*Single Fighter*" pada penelitian ini bercerita tentang perjuangan seekor Gajah yang menyelamatkan anaknya dari tawanan musuh atau *enemy*. Latar kejadian permainan ini berada di sebuah hutan dan terdapat banyak monyet, srigala serta kingkong yang menjadi musuh gajah. Jika sang gajah mampu mengalahkan semua musuhnya maka dia akan bertemu dengan anaknya yang ditawan jauh didalam hutan. Untuk menciptakan suasana hidup dalam *game* ini, aksi para *enemy* harus diberi kecerdasan buatan dengan Logika *Fuzzy* menggunakan metode Sugeno.

Pada karakter musuh akan diberikan kecerdasan buatan berdasarkan variabel yang sudah ditentukan yaitu *health point* musuh, amunisi dan *distance* antara musuh dengan *player*. Penerapan logika *fuzzy* pada permainan ini menghasilkan *output* yaitu musuh memiliki perilaku untuk menyerang *player*.

Pada penelitian lain Logika *Fuzzy* Sugeno juga digunakan untuk menentukan peran robot dalam *game* sepak bola yang menggunakan 2 variabel yaitu sudut dan jarak terhadap bola dan mengatur perilaku musuh dalam *game* bertipe *Action-RPG* yang menggunakan 3 variabel yaitu *life*, jumlah panah, dan *range distance* (Purba, 2013).

Hasil dari kedua penelitian tersebut memiliki keberhasilan 90% dari 10 percobaan yang berhasil mengoptimasi penentuan peran robot dalam *game* sepak bola dan pada *game* "*Song of Ruination 2*" logika Fuzzy Sugeno berjalan dengan baik dimana tiap karakter musuh memiliki prosentase tingkat agresif saat menyerang *player* sebesar 45% untuk tipe penyerang, 49% untuk tipe pemanah, dan 89% untuk tipe boss (Al Hasmy, 2011).

Dari hasil *review* tentang penerapan Logika *Fuzzy* Sugeno terhadap penentu peran robot maupun perilaku karakter musuh dalam *game* diatas, penulis berkeinginan untuk mengembangkan penerapan tersebut dalam *game* adventure "*Single Fighter*" berplatform Android dengan *genre side scrolling game*.

II. TINJAUAN PUSTAKA

2.1 Logika *Fuzzy* Sugeno

Teori himpunan logika samar dikembangkan oleh Prof. Lotfi Zadeh pada tahun 1965. Zadeh berpendapat bahwa logika benar dan salah dalam logika konvensional tidak dapat engatasi masalah gradasi yang berada pada dunia nyata. Untuk mengatasi masalah gradasi yang tidak terhingga tersebut, Zadeh mengembangkan sebuah himpunan *fuzzy*. Tidak seperti logika *boolean*, logika *fuzzy* mempunyai nilai yang *continue*. Kondisi samar dinyatakan dalam derajat dari suatu keanggotaan dan derajat dari kebenaran. Oleh sebab itu sesuatu dapat dikatakan sebagian benar dan sebagian salah pada waktu yang sama.

Sebelum menjadi sebuah keputusan, maka ada beberapa tahap proses *fuzzy*, yaitu :

- *Fuzzyfikasi* : Mencari derajat keanggotaan dari input *fuzzy* yang memenuhi *fuzzy* rule
- Operasi Logika *Fuzzy* : menggunakan fungsi AND, yaitu mengambil nilai MAX dari derajat keanggotaan setiap input *fuzzy*.
- Implikasi : Mendapatkan keluaran dari *fuzzy rule* menggunakan fungsi MIN.
- Agregasi : Mengkombinasikan keluaran semua *fuzzy rule* menjadi *fuzzy* set tunggal dengan menggunakan fungsi Max.
- *Defuzzyfikasi* : Masukkan *fuzzy* set (dalam hal ini *fuzzy* set hasil agregasi) dan keluaran sebuah bilangan tunggal.

Model Sugeno merupakan *fuzzy* yang menggambarkan dengan "IF-THEN" yaitu sebuah aturan yang mewakili hubungan input atau output lokal dari sistem nonlinear. Dalam metode Sugeno, output sistem berupa konstanta atau persamaan linier. Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada 1985. Berikut adalah bentuk umum model *fuzzy* Sugeno (Kusumadewi, 2010).

if(x1 is A1)…..(xn is An then z=f(x,y)) (1)

2.2 Unity 3D

Dalam pembuatan aplikasi ini menggunakan *software* Unity 3D karena Unity 3D mendukung bahasa pemrograman C#. Bagian *projects* meliputi semua elemen dalam *game* yang dibuat, seperti *models, scripts, levels*, dan *menu*. Unity lebih tepat dijelaskan sebagai software yang mengembangkan video *game* atau disebut juga *game engine* (Watkins, 2011).

2.3 Android

Sistem operasi berbasis android dipilih karena bersifat *open source*, jadi sangat memungkinkan penggunanya untuk membuat *software* sendiri. Pengertian sistem operasi android merupakan sebuah sistem operasi untuk perangkat mobile berbasis Linux yang dikeluarkan oleh Google Inc pada bulan November 2007 (Wahana Komputer, 2013).

III. METODE PENELITIAN

Untuk memperoleh hasil penelitian berupa *game* bergenre *side scrolling game* dari implementasi logika *fuzzy*, maka digunakan beberapa metode penelitian sebagai berikut :

3.1 *Pre Production*

Tahap *pre-production* adalah tahap dalam pembuatan *game* yang berfokus pada ide dan konsep pengembangan *game*. Tujuan dari proses ini adalah untuk memperoleh gambaran jelas tentang *game* yang akan dibuat serta membantu dalam menentukan target serta *deadline* dalam proses pembuatan *game*.

Pada tahap ini terdiri dari :

- Penulisan *Story*
- Pembuatan *Storyboard*
- Melakukan Desain *Game*
- Mempersiapkan Audio

3.2 *Production*

Pada proses ini terdapat tahapan untuk merubah konsep menjadi realita, dari ide yang sudah dipikirkan menjadi *gameplay* pada layer. Tahap pengumpulan segala kebutuhan produksi yang telah dipersiapkan dari tahap *Pre Production* akan diolah lebih lanjut menggunakan *software* Unity 3D. Semua *tool* dan bahan yang telah disediakan harus kompatibel, supaya proses ini dapat diterapkan menggunakan Logika *Fuzzy* Sugeno.

3.3 *Post Production*

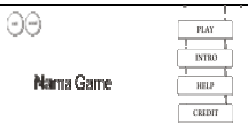
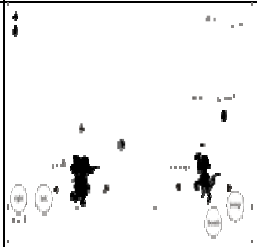
Setelah *game* selesai dibuat, dilakukan beberapa pengujian atau *testing*. Untuk menguji hasil perhitungan Logika *Fuzzy* dalam membangun karakter musuh akan dilakukan perhitungan manual yang dicocokkan dengan hasil output otomatis pada *field* dalam *software Unity*. Untuk menguji fungsionalitas dari *game* yang telah dibangun menggunakan *Blackbox,* sedangkan untuk menguji *game* tersebut *support* diberbagai jenis *smartphone* maka dilakukan uji *device* ke perangkat seperti *tablet* dan beberapa *smartphone* berbeda merk, kemudian untuk menguji kelayakan dari *game* tersebut dilakukan penyebaran Kuesioner.

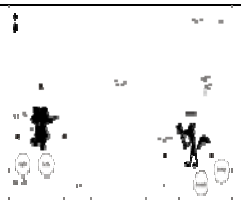
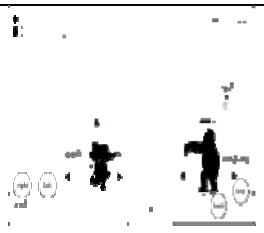
IV. HASIL DAN PEMBAHASAN

4.1 *Pre-Production*

a. Storyboard

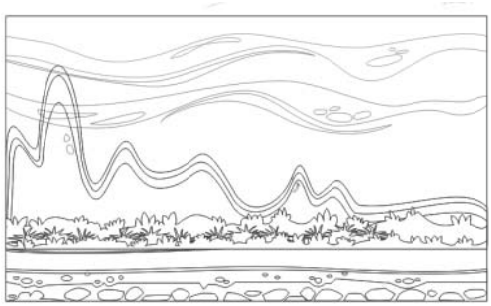
Tabel 1. *Storyboard Game*

No	Scene	Keterangan
1		***Splash Screen*** *Scene* pertama berisi *splash screen* yang menampilkan judul *game*, *copyright*, dan logo almamater pembuat.
2		**Main Menu** Rancangan tampilan main menu berisi 4 menu utama dan 2 tombol untuk setting sound dan tombol keluar aplikasi *game* ini. Sound yang digunakan pada *scene* ini adalah *Jungle Bit Jukebox.*
3		***Credit Scene*** Tidak di isi dengan audio Berisi : - Nama pembuat - *Software* pembuatan *game* - *Sound* yang digunakan - Ucapanterimakasih
4		***Intro Scene*** Berisi : - Judul *game story* - Urutan alur cerita
5		***Help Scene*** Tidak di isi dengan audio Berisi *icon* yang terdapat pada aplikasi *game* beserta keterangannya.
6		***Level* 1** Pada semua *level* menggunakan *Sound Jungle Indian Soundtrack* Latar *level* 1 bertemakan hutan di pagi hari Berisi : - Indikator jumlah nyawa *player* - Indikator jumlah bom *player* - Tampilan skor yang terkumpul - Tombol *pause* - Tombol *left* - Tombol *right* - Tombol *jump* - Tombol bom untuk menyerang musuh

No	Scene	Keterangan
		- Terdapat koin sebagai dasar perhitungan skor - Terdapat gambar bom di area sekitar gajah berlari sebagai bantuan untuk menambah jumlah persediaan bom Pada *level 1* pemain akan dihadapkan dengan musuh monyet yang melempari *player* dengan batu. *Player* dapat menyerang musuh menggunakan bom. Jika sudah sampai pada skor 200, *player* akan masuk ke *level* 2.
7		***Level 2*** Latar di sore hari. Pada *level* 2 *player* akan dihadapkan pada srigala yang menyerang menggunkan pisau besar dan tajam. Jika jumlah nyawa *player* berada dibawah 50% maka secara otomatis akan ada bantuan power up berupa gambar apel. Jika sudah sampai pada skor 500, *player* akan masuk ke *level* 3.
8		***Level* 3** Latar di malam hari. Pada *level* 3 *player* dihadapkan dengan musuh *giant* yang bernama kingkong. Musuh ini sangat kejam, dia dapat memukul, menendang, dan melakukan serangan spesial yang berakibat pada berkurangnya nyawa *player* begitu banyak. Jika berhasil melewati seluruh kingkong dan skor yang didapatkan diatas 700 maka *player* dinyatakan menang.
9		***Game Over Scene*** Berisi skor baru dan skor lama sebagai perbandingan nilai. Terdapat *action (tap)* jika ingin mengulang *game.* *Scene* ini akan muncul jika *player* mati karena tercebur kedalam air atau terbunuh oleh serangan musuh.
10		***Win Game Scene*** Berisi skor baru dan skor lama. Terdapat *action (tap)* jika ingin mengulang *game.* *Scene* ini muncul jika *player* berhasil mengalahkan seluruh musuh

b. Desain *Game*

Desain *Game* diawali dengan sketsa *background* sebagai *setting* latar permainan yang dirancang menggunakan Coreldraw X4, seperti gambar dibawah ini:

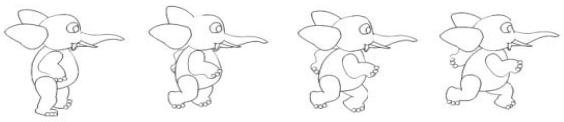


Gambar 1. Desain *Background*

Selain itu desain *game* akan ditambahkan dengan elemen pendukung atau yang sering disebut dengan *enviorment*.

c. Desain Karakter

Pembuatan karakter menggunakan CorelDraw X4 dengan model *animation sprite*, yang digambar tiap gerakan *frame by frame* yang tampak seperti dibawah ini:



Gambar 2. Desain Karakter Utama

Gambar 3. Desain Karakter Musuh

4.2 ***Production***

a. Jenis Musuh

Dalam permainan yang akan dirancang pada penelitian ini, *player* berhadapan dengan musuh sebagai berikut:

Tabel 2. Rincian Tipe Musuh, Nama dan Tampilannya.

Tipe	**Nama**	**Gambar**
Penyerang	Joger	
Penembak	Khero	
Giant	King Kong	

Terdapat 3 tipe musuh dalam permainan ini yaitu :

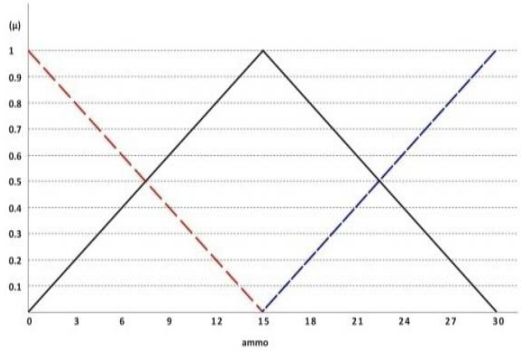
- Penyerang
  Memiliki sifat agresif. Mereka akan maju sebagai *front-liner* musuh sebagai penyerang jarak dekat.
- Penembak
  Memiliki sifat tidak agresif, menembakan peluru dari kejauhan.
- Giant
  Memiliki sifat agresif, musuh tipe giant memiliki karakteristik serangan kuat, nyawa banyak, namun geraknya lambat.

b. Variabel Penentu Perilaku Musuh

Dalam logika *fuzzy* yang dirancang ini, terdapat 3 variabel linguistik sebagai penentu perilaku musuh yaitu *ammo*, *health* dan *distance*. Pada bagian ini juga akan dirinci variable mana saja yang akan dipakai oleh setiap tipe musuh, misalnya tipe penyerang hanya menggunakan 2 variabel *input* yaitu *distance* dan *health*.

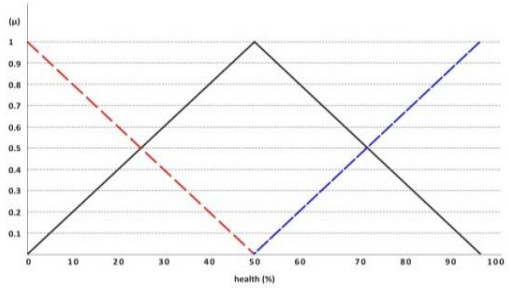
Berikut ini 3 variabel penentu tersebut, dan juga fungsi keanggotaanya :

- Variabel Ammo
  Ammo atau istilah untuk amunisi dari batu yang akan dilemparkan monyet untuk menyerang *player*. Jumlah *ammo* merupakan angka bulat dengan nilai antara 0-10.



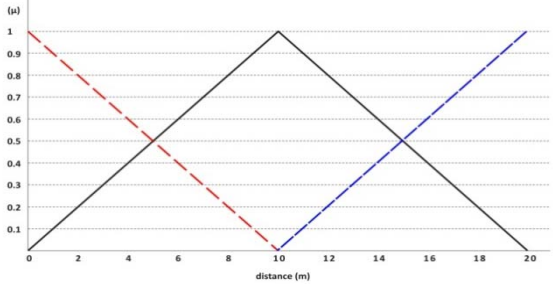
Gambar 4. Fungsi Keanggotaan Ammo

- Variabel *Health*
  Jumlah *health* merupakan angka bulat. Variable *health_max* merupakan *health* (nyawa) maksimal yang dimiliki oleh jenis musuh tertentu.



Gambar 5. Fungsi Keanggotaan Health

- Variabel *Distance*
  Distance merupakan jarak pemain terhadap musuh, dalam meter. Berikut ini adalah fungsi keanggotaan untuk *distance*.

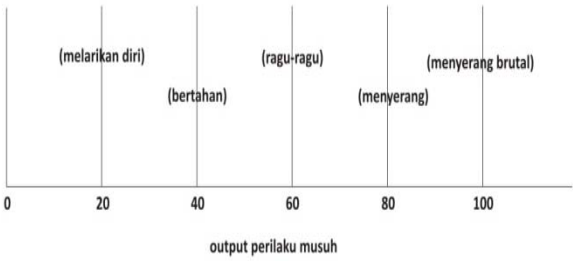


Gambar 6. Fungsi Keanggotaan Distance

Input variable penentu perilaku tersebut nantinya akan menentukan musuh akan bertindak bagaimana. Perilaku musuh terbagi dalam 5 kategori yaitu :

- Melarikan Diri/Lari (range 0-20)
- Bertahan (*range* 21-40)
- Ragu-ragu (*range* 41-60)
- Menyerang (*range* 61-80)
- Menyerang brutal (*range* 81-100)

Diagram output untuk perilaku musuh terlihat pada gambar berikut ini:



Gambar 7. Diagram Perilaku Musuh

Rule pada setiap tipe musuh ditentukan masing-masing dengan tingkat kesulitan yang berbeda, berikut adalah *rule* dari ketiga musuh :

- *Fuzzy rule* untuk musuh penyerang :
  - `IF distance_near AND health_few THEN lari`
  - `IF distance_near AND health_med THEN bertahan`
  - `IF distance_near AND health_much THEN menyerang`
  - `IF distance_med AND health_few THEN lari`
  - `IF distance_med AND health_med THEN bertahan`
  - `IF distance_med AND health_much THEN menyerang`
  - `IF distance_far AND health_few THEN bertahan`
  - `IF distance_far AND health_med THEN menyerang`
  - `IF distance_far AND health_much THEN brutal`
- *Fuzzy rule* untuk musuh penembak :
  - `IF distance_near AND ammo_little THEN lari`
  - `IF distance_near AND ammo_med THEN lari`

- `IF distance_near AND ammo_many THEN lari`
- `IF distance_med AND ammo_little THEN bertahan`
- `IF distance_med AND ammo_med THEN ragu`
- `IF distance_med AND ammo_many THEN ragu`
- `IF distance_far AND ammo_little THEN menyerang`
- `IF distance_far AND ammo_med THEN menyerang`
- `IF distance_far AND ammo_many THEN menyerang`

- *Fuzzy* rule untuk musuh giant :
  - `IF distance_near AND health_few THEN bertahan`
  - `IF distance_near AND health_med THEN menyerang`
  - `IF distance_near AND health_much THEN brutal`
  - `IF distance_med AND health_few THEN bertahan`
  - `IF distance_med AND health_med THEN brutal`
  - `IF distance_med AND health_much THEN brutal`
  - `IF distance_far AND health_few THEN menyerang`
  - `IF distance_far AND health_med THEN brutal`
  - `IF distance_far AND health_much THEN brutal`

Keseluruhan proses logika *fuzzy* dapat diterangkan sebagai berikut:

- *Fuzzyfikasi Input* : Mencari derajat keanggotaan dari input *fuzzy* yang memenuhi *fuzzy rule*.
- Operasi *Fuzzy* : Menggunakan fungsi AND yaitu mengambil nilai MAX dari derajat keanggotaan setiap input *fuzzy*.
- Implikasi : Mendapatkan keluaran dari *fuzzy* rule menggunakan fungsi MIN.
- Agregasi : Mengkombinasikan keluaran semua *fuzzy* rule menjadi *fuzzy* set tunggal dengan menggunakan fungsi MAX.
- Hasil *Fuzzy* : Menghasilkan nilai perilaku musuh, antara skala 0-100 dengan metode *weighted average*. Nilai 0-100 ini nantinya menentukan peluang musuh menyerang pemain.

c. Proses *Fuzzy*

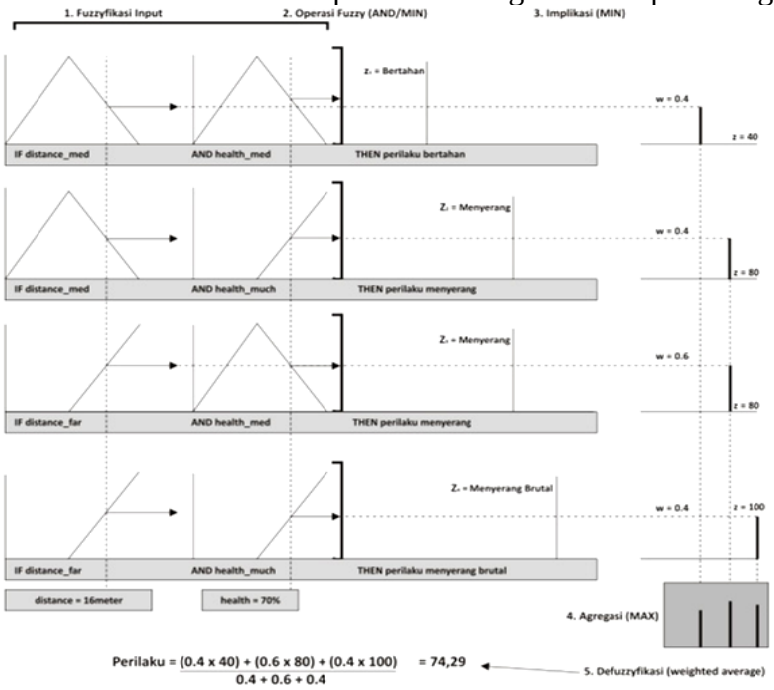
Operasi Logika *fuzzy* yang digunakan pada penelitian ini adalah menggunakan fungsi AND (fungsi MIN). Proses *fuzzyfikasi* yang digunakan adalah FIS tipe Sugeno berorder nol, karena *fuzzy* Sugeno menghasilkan konstanta tegas, sehingga dapat mewakili nilai perilaku yang telah di rancang. Proses implikasi menggunakan fungsi MIN, dan agregasi menggunakan fungsi MAX. kemudian hasil *fuzzy* dihitung dengan weighted average, yaitu dengan rumus:

$$output = \frac{\sum_{i=1}^{N} WiZi}{\sum_{i=1}^{N} Wi} \qquad (2)$$

Jika mengambil contoh musuh tipe penyerang, dengan keadaan saat ini miliki health 70% dan distance 16 meter dari pemain, maka rule yang memenuhi adalah

- `IF distance_med AND health_med THEN bertahan`
- `IF distance_med AND health_much THEN menyerang`
- `IF distance_far AND health_med THEN menyerang`
- `IF distance_far AND health_much THEN brutal`

Berdasarkan contoh tersebut dapat dilihat diagram hasil perhitungan yaitu:



Gambar 8. Diagram Proses *Fuzzy*

Proses perhitungan *fuzzy* dengan dijabarkan maupun dengan diagram diatas terlihat bahwa hasilnya 74, berarti musuh penyerang yang memiliki distance (jarak ke *player*) 16 meter dan health (nyawa) 70% memiliki peluang 74% untuk maju menyerang pemain, dan 26% untuk mundur.

d. Hasil Pengujian *Fuzzy*

- Penyerang (perhitungan menggunakan variabel *health* dan *distance*)

Tabel 3. Hasil Pengujian perilaku Penyerang.

HASIL PENGUJIAN PERILAKU PENYERANG											
health (%) \ *distance* (m)	0	2	4	6	8	10	12	14	16	18	20
0	20	20	20	20	20	20	24	28	32	36	40
10	24	24	25	25	24	24	33	37	40	43	48
20	28	28	28	28	28	28	37	43	46	55	56
30	32	32	32	32	32	32	45	46	51	57	64
40	36	36	35	34	36	36	49	50	57	63	72
50	40	40	40	40	40	40	48	56	64	72	80
60	48	48	50	50	48	48	65	63	70	77	84
70	56	56	56	56	56	56	73	85	74	80	88
80	64	64	64	64	64	64	70	74	77	83	92
90	72	72	70	68	72	72	77	80	83	87	96
100	80	80	80	80	80	80	84	88	92	96	100

Dari tabel diatas, terlihat bahwa perilaku penyerang adalah :

- Melarikan diri : 5% (6 dari 121 data)
- Bertahan : 32% (39 dari 121 data)
- Ragu-ragu : 22% (27 dari 121 data)
- Menyerang : 30% (36 dari 121 data)

- Menyerang Brutal : 11% (13 dari 121 data)

- Penembak (perhitungan menggunakan variabel *distance* dan *ammo*)

Tabel 4. Hasil Pengujian perilaku Penembak.

HASIL PENGUJIAN PERILAKU PELEMPAR											
***distance* (m)** / **ammo**	**0**	**2**	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**
0	20	24	28	32	36	40	48	56	64	77	80
1	20	30	33	37	38	44	50	57	63	73	80
2	20	32	37	40	43	48	53	57	63	68	80
3	20	32	37	43	47	52	57	80	80	68	80
4	20	30	37	43	50	56	60	65	67	73	74
5	20	28	36	44	52	60	60	68	72	76	80
6	20	28	36	44	52	60	64	69	72	76	80
7	20	30	36	44	51	60	64	68	72	72	80
8	20	30	36	44	60	60	66	68	72	72	80
9	20	32	39	44	52	60	62	68	73	76	80
10	20	28	36	44	52	60	60	68	72	76	80

Dari tabel diatas, terlihat bahwa perilaku penembak adalah :
- Melarikan diri : 9% (11 dari 121 data)
- Bertahan : 23% (28 dari 121 data)
- Ragu-ragu : 31% (38 dari 121 data)
- Menyerang : 36% (44 dari 121 data)
- Menyerang Brutal : 0% (0 dari 121 data)

Giant (perhitungan menggunakan variable *distance* dan *health)*

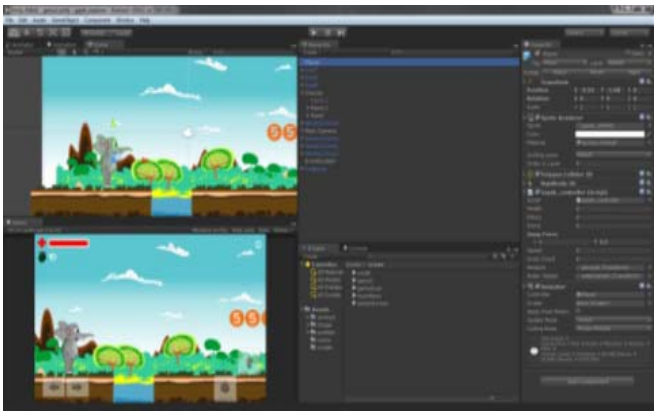
Tabel 5. Hasil Pengujian perilaku *Giant*.

distance (m) / *health* (%)	0	2	4	6	8	10	12	14	16	18	20
0	40	40	40	40	40	40	48	56	64	72	80
10	48	57	60	60	57	52	57	63	70	77	84
20	56	63	69	69	67	64	67	69	74	80	88
30	64	70	74	77	77	76	77	77	77	83	92
40	72	77	80	83	87	88	87	84	84	87	96
50	80	84	88	92	96	100	100	100	100	100	100
60	84	84	88	92	96	100	100	100	100	100	100
70	88	88	88	92	95	100	100	100	100	100	100
80	92	92	92	92	95	100	100	100	100	100	100
90	96	96	95	95	96	100	100	100	100	100	100
100	100	100	100	100	100	100	100	100	100	100	100

Dari tabel diatas, terlihat bahwa perilaku giant adalah :
- Melarikan diri : 0% (0 dari 121 data)
- Bertahan : 5% (6 dari 121 data)
- Ragu-ragu : 8% (10 dari 121 data)
- Menyerang : 23% (28 dari 121 data)
- Menyerang Brutal : 64% (77 dari 121 data)

e. **Implementasi Sistem**
- Pembuatan *Scene Game*

Langkah awal pembuatan scene menggunaka Unity 3D ini adalah dengan menyusun objek-objek yang sudah dibuat sesuai dengan konsep seperti *sprite player*, *background* dan objek-objek pendukung lainnya seperti *power up*, bom maupun *enviorment* tanah dan air.



Gambar 9. Pembuatan Arena *Game*

- Proses *build game*
  Game yang sudah jadi akan di build ke file berekstensi '.apk' agar dapat diinstall ke android.

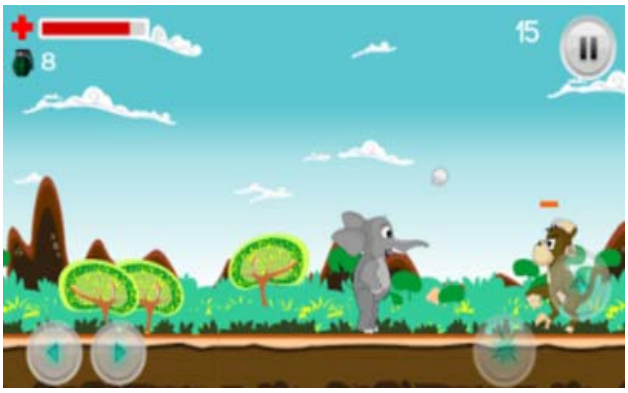
4.3 ***Post-Production***

a. *Publish* ke *Play Store*

APK yang sudah dibuat kemudian dapat diupload ke *Play Store.* Jika berhasil maka akan keluar jendela konfigurasi produksi atau kelengkapan informasi tentang APK tersebut dan aplikasi sudah berhasil terpasang.

Proses instalasi aplikasi menggunakan smarthphone android cukup mudah dengan download *via Play Store* kemudian *install*. Berikut adalah tampilan *game* di *smarthphone* yang akan dimainkan

Tampilan *level game* seperti pada Gambar 10, Gambar 11 dan Gambar 12, perbedaan masing-masing *level* adalah rintangan yang dihadapi.



Gambar 10. Tampilan *Level* 1

Gambar 11. Tampilan *Level* 2

Gambar 12. Tampilan *Level* 3

b. Pengujian dengan *Balckbox*

- Scene
  - *Splash Screen*: *Splash Screen* muncul,saat awal *game* dijalankan
  - *Main Menu*: *Main menu* muncul, setelah *splash screen*
  - *Credit Screen*: Muncul, jika tombol *credit screen* ditekan pada *main menu*
  - *Intro Game*: Muncul, jika tombol *Intro* ditekan pada mainmenu
  - *Game Over* : Muncul, jika *player* tercebur atau diserang musuh hingga nyawanya habis
  - *Game Win*: Muncul, jika *player* berhasil mengalahkan musuh sampai kingkong tuntas
- *Player* Animasi
  - Berlari: Berjalan, sejak awal *game* dimainkan
  - Melompat: Berjalan, jika mendapatkan inputan tap
- Animasi Musuh
  - Monyet: Berjalan, sejak awal *game* dimainkan
  - Serigala: Bergerak setelah *player* berhasil mengalah-kan monyet
  - Kingkong: Bergerak setelah *player* berhasil mengalah-kan monyet dan srigala
- *Sound*
  - *Back sound*: Berbunyi, sejak awal *game* dimainkan
  - Duarr: Berbunyi, ketika *player* melempar bom ke musuh
  - Suara Percikan Air: Berbunyi, ketika *player* atau musuh tercebur ke dalam air

- *Main Music*: Berbunyi, ketika *main menu* muncul

- *Button*
  - *Play* : Berfungsi, dan untuk masuk ke arena *game*
  - *Credit* : Berfungsi, dan untuk masuk ke tampilan credit screen
  - *Quit* : Berfungsi, dan untuk keluar *game*
  - *Pause* : Berfungsi, dan untuk menghenti-kan *game* sementara
  - *Play* : Berfungsi, dan untuk memainkan *game* kembali
- Grafis
  - Tampilan *Game*: Bagus, seluruh desain tampil dan memenuhi layar
  - Resolusi Layar: Bagus, *game* dapat dijalankan pada beberapa jenis resolusi layar
- *Game Play*
  - Inputan *Player*: Berfungsi, jika *player Tap* layar akan menggerak-kan karakter *player*
  - Gerakan Musuh: Berfungsi, seluruh musuh bergerak sesuai *script* yang diberikan.
  - Gerakan Objek: Berfungsi, seluruh objek bergerak sesuai *script* yang diberikan
  - Kecepatan Musuh: Berfungsi, kecepatan musuh sesuai *script* yang diberikan dimana jika tingkatan bertambah maka kecepatannya pun bertambah
  - Indikator Menang: Berfungsi, sesuai *script* yang diberikan dimana jika *player* menemukan anaknya maka *game win* akan muncul
  - Indikator *Game Over*: Berfungsi, sesuai *script* yang diberikan dimana *player* tercebur atau nyawanya habis

c. Pengujian *Device*

Permainan ini dirancang dan diujicoba ke *device* dengan spesifikasi :

Tabel 6. Uji Device Menggunakan Tabel t

TABELT EVERCOSS W7B Android OS, v4.4.4 (KitKat) Prosesor Quad Core 1,3GHz Resolusi 480x800 *pixels*, 7 *inches* 512MB RAM	Animasi	Seluruh animasi berjalan lancar
	Audio	Seluruh *audio* berjalan dengan baik
	Tombol	Seluruh tombol fungsi berfungsi dengan baik
	Grafis	resolusi gambar terlihat jelas pada resolusi 480x800 px, tanpa ada sisi kosong
	Screen play	*screenplay* berjalan baik
	Game play	tidak ditemukan *bug* secara kesuluruhan *game* berjalan lancar
	Smooth ness	berjalan kurang begitu lancar /terkadang terjadi *lag* namun tidak menganggu
	Memory	Ketika menginstall *game* maka memakan *memory* sebesar 17.87MB

d. Pengujian dengan Kuesioner

Tabel 7. Hasil Pengujian Tiap Aspek

No	Aspek Penguji *Game*	Hasil (%)
1	Tampilan *game* dan animasi menarik	87.9
2	Audio sesuai dengan aksi atau scene yang sedang berjalan	95.0
3	Cerita atau konsep dalam *game* ini sesuai dengan *game* yang sedang dimainkan	79.3
4	Bahasa dalam aplikasi *game* ini mudah dipahami	92.9
5	Penggunaan warna dan desain background sudah sesuai	90.7
6	Ketetapan ukuran, warna, dan pemilihan jenis tulisan sesuai	81.4
7	Ukuran, warna, dan bentuk tombol pada *game* sudah sesuai	79.3
8	Ketetapan fungsi tombol dengan tujuan yang di inginkan sesuai	87.1
9	Kesesuaian ukuran dan warna karakter	92.9
10	Kemudahan pengoperasian *game*	89.3

Dari hasil pengujian diketahui bahwa 28 responden memberikan penilaian terhadap *game "Single Fighter"* memiliki keunggulan dalam bentuk karakter, desain *background* dari animasi yang sesuai dengan tema dengan bahasa yang mudah dipahami serta efek *sound* berjalan dengan baik. Kelemahan dari *game* ini adalah terletak pada ceritanya yang tidak banyak menampilkan kisah sebelumnya dalam bentuk animasi serta penampilan tombol yang samar.

V. KESIMPULAN

5.1 Kesimpulan

Dari hasil pembuatan dan pengujian *game* "*Single Fighter*", dapat ditarik kesimpulan sebagai berikut :

1. Logika *Fuzzy* dapat diterapkan untuk mengatur perilaku musuh. Perilaku ini ditentukan dari jarak musuh ke *player (distance),* amunisi *(ammo)* dan nyawa musuh *(health).*
2. Penerapan *fuzzy* dalam permainan ini telah berjalan dengan baik, dimana musuh tipe penyerang menjadi cukup agresif (30% perilaku menyerang, 11% menyerang brutal), tipe penembak agresif jika berada pada jarak jauh (36% perilaku menyerang), dan musuh giant sangat agresif (23% perilaku menyerang, 64% perilaku menyerang brutal).
3. Aplikasi *game* "*Single Fighter*" berhasil di *publish* di *Google Play Store* yang dibuat dari *game engine* Unity, ber-*genre side scrolling game*, dan berplatform android.
4. Aplikasi *game* "*Single Fighter*", dapat dijalankan pada berbagai macam *smartphone* berbasis Android dengan spesifikasi minimum RAM 512MB, CPU *single core* 1Ghz, resolusi 320 x 480 px.

5.2 Saran

Berdasarkan kesimpulan diatas, maka saran yang dapat penulis berikan agar aplikasi dapat dikembangkan menjadi lebih baik dengan cara :

1. Penambahan *variable* input untuk *fuzzy*, misalnya musuh yang menyerang tidak hanya satu, atau jika rekan musuh nyawanya melemah, musuh yang lain langsung datang membantu menyerang.
2. Ditambahkan objek-objek animasi yang lebih menarik

3. Pengembangan cerita yang lebih kompleks dan variatif sehingga *game* akan memiliki umur yang cukup panjang.

DAFTAR PUSTAKA

Al Hasmy, R. (2011). Penentuan Peran Dalam Robot Sepak Bola Dengan Metode Fuzzy Sugeno . *Jurnal ITS Surabaya* .

Kusumadewi, S. (2010). *Aplikasi Logika Fuzzy Untuk Pendukung Keputusan Edisi 2.* Yogyakarta: Graha Ilmu.

Purba, R. (2013). Implementasi Logika Fuzzy Untuk Mengatur Perilaku Musuh Dlaam Game Bertipe Action-RPG . *Jurnal EECCISVol.7, No.1* .

Wahana Komputer. (2013). *Step by Step Menjadi Programmer Android.* Semarang: Andi Publisher.

Watkins, A. (2011). *Creating Games with Unity and Maya* . Burlington: Elsevier Inc.