

**PERGERAKAN NPC (*NON PLAYER CHARACTER*) TERHADAP
PLAYER MENGGUNAKAN ALGORITMA DJIKSTRA DAN
FUZZY LOGIC SUGENO UNTUK MENGATUR BENTUK
BONUS PADA *GAME 3D "MUSLIM KID"***

SKRIPSI

Oleh :

IHYA ISNANI MUHARROMAH

NIM 12650051



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2017**

HALAMAN PENGAJUAN

**PERGERAKAN NPC (*NON PLAYER CHARACTER*) TERHADAP
PLAYER MENGGUNAKAN ALGORITMA DJIKSTRA DAN
FUZZY LOGIC SUGENO UNTUK MENGATUR BENTUK
BONUS PADA *GAME 3D "MUSLIM KID"***

SKRIPSI

**Diajukan Kepada:
Fakultas Sains dan Teknologi
Universitas Islam Negeri
Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh :
IHYA ISNANI MUHARROMAH
NIM 12650051**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2017**

LEMBAR PERSETUJUAN

**PERGERAKAN NPC (NON PLAYER CHARACTER) TERHADAP
PLAYER MENGGUNAKAN ALGORITMA DJIKSTRA DAN
FUZZY LOGIC SUGENO UNTUK MENGATUR BENTUK
BONUS PADA GAME 3D "MUSLIM KID"**

SKRIPSI

Oleh:

Ihva Isnani Muharromah
NIM 12650051

Telah Diperiksa dan Disetujui untuk Diuji :
Tanggal : 10 Maret 2017

Pembimbing I

Pembimbing II



Yunifa Miftachul Arif, M.T
NIP. 19830616 201101 1 004



Roro Inda Melani, M.T., M.Sc
NIP. 19780925 200501 2 008

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Dr. Cahyo Crysdiyan
NIP. 19740424 200901 1 008

HALAMAN PENGESAHAN

**PERGERAKAN NPC (NON PLAYER CHARACTER) TERHADAP
PLAYER MENGGUNAKAN ALGORITMA DJIKSTRA DAN
FUZZY LOGIC SUGENO UNTUK MENGATUR BENTUK
BONUS PADA GAME 3D "MUSLIM KID"**

SKRIPSI

Oleh :

Ihya Isnani Muharromah

NIM 12650051

Telah Dipertahankan di Depan Dewan Penguji Skripsi dan
Dinyatakan Diterima Sebagai Salah Satu Persyaratan Untuk
Memperoleh Gelar Sarjana Komputer (S.Kom)

Tanggal 10 Maret 2017

Susunan Dewan Penguji

1. Penguji Utama : Hani Nurhayati, M.T
NIP. 19780625 200801 2 006
2. Ketua : Dr. Muhammad Faisal, M.T
NIP. 19740510 200501 1 007
3. Sekretaris : Yunifa Miftachul Arif, M.T
NIP. 19830616 201101 1 004
4. Anggota : Roro Inda Melani, M.T., M.Sc
NIP. 19780925 200501 2 008

Tanda Tangan

()
()
()
()

**Mengetahui dan Mengesahkan
Ketua Jurusan Teknik Informatika**



Dr. Chvo Crvsdian

NIP. 19740424 200901 1 008

**HALAMAN PERNYATAAN
ORISINALITAS PENELITIAN**

Nama : Ihya Isnani Muharromah

NIM : 12650051

Jurusan : Teknik Informatika

Fakultas : Sains dan Teknologi

Judul Skripsi : PERGERAKAN NPC (*NON PLAYER CHARACTER*) TERHADAP PLAYER MENGGUNAKAN ALGORITMA DJIKSTRA DAN FUZZY LOGIC SUGENO UNTUK MENGATUR BENTUK BONUS PADA GAME 3D "MUSLIM KID"

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 17 Februari 2017
Yang membuat pernyataan,



Ihya Isnani Muharromah
NIM 12650051

HALAMAN MOTTO

Man Jadda Wa Jadda

Barang siapa yang bersungguh - sungguh akan mendapatkannya.

Man Shabara Zhafira

Siapa yang bersabar pasti beruntung

Man Sara Ala Darsi Washala

Barangsiapa berjalan pada jalannya, sampailah dia



HALAMAN PERSEMBAHAN

Bismillahirrohmanirrohim, kupersembahkan sebuah karya sederhanaku ini untuk orang-orang yang paling kusayangi, kubanggakan, dan selalu memberikan energi semangat untukku..

Seluruh keluarga besarku Bani Kusairi

Khususnya Bapak dan Ibu tercinta

Suyadi dan S.pami

*yang selalu ikhlas mendoakan putra-putrinya
yang selalu mengarahkan kami dalam kebaikan
yang dengan sabar membimbing kami.*

Semoga Allah SWT melindungi dan menjaga mereka dalam naungannya...

Aamiin

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji bagi Allah SWT yang Maha Pengasih lagi Maha Penyayang atas Rahmat dan Hidayah-Nya sehingga penulis dapat menyelesaikan penyusunan skripsi ini. Sholawat serta Salam tetap tercurahkan kepada junjungan kita, kekasih Allah, Nabi Muhammad SAW, sang pemberi syafaat kelak di hari akhir, beserta seluruh keluarga, sahabat, dan para pengikutnya.

Penelitian skripsi yang berjudul **“PERGERAKAN NPC (NON PLAYER CHARACTER) TERHADAP PLAYER MENGGUNAKAN ALGORITMA DJIKSTRA DAN FUZZY LOGIC SUGENO UNTUK MENGATUR BENTUK BONUS PADA GAME 3D “MUSLIM KID”** ini ditulis untuk memnuhi salah satu syarat guna memperoleh gelar Sarja Strata Satu (S1) Fakultas Sains dan Teknologi Universitas Maulana Malik Ibrahim Malang. Karya penelitian skripsi ini tidak akan pernah ada tanpa bantuan baik moral maupun spiritual dari berbagai pihak yang telah terlibat. Untuk itu dengan segala kerendahan hati, penulis mengucapkan rasa terimakasih yang sebesar-besarnya kepada:

1. Prof. Dr. H. Mudjia Rahardjo, M.Si, selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Bapak Yunifa Miftachul Arif, M.T, selaku Dosen Pembimbing I yang telah bersedia meluangkan waktu, tenaga dan pikiran untuk memberikan bimbingan, berbagai pengalaman, arahan, nasihat, motivasi dan pengarahan dalam pembangunan program hingga penyusunan skripsi ini.
3. Ibu Roro Inda Melani, M.T, selaku dosen pembimbing 2 yang selalu memberi masukan, serta pengarahan dalam penyusunan laporan skripsi ini.
4. Bapak Suhartono, M.Kom, selaku dosen wali yang juga selalu memberi pengarahan terkait akademik selama masa study.
5. Dr. Cahyo Crys dian selaku ketua jurusan Teknik Informatika yang mendukung dan mengarahkan skripsi ini.
6. Segenap civitas akademika Fakultas Saintek, Universitas Islam Negeri Maulana Malik Ibrahim Malang terutama seluruh dosen, terimakasih atas segala ilmu dan bimbingannya.

7. Bapak, Ibu dan seluruh keluarga yang selalu memberikan doa, kasih sayang, semangat, dukungan moril, serta motivasi sampai saat ini, terimakasih banyak.
8. Heru Santoso, Nur Halimatus S, Rifqy Rahmatika yang membantu internal maupun eksternal dalam bentuk semangat kepada penulis untuk menyelesaikan skripsi ini.
9. Teman-teman angkatan 2012, yang berjuang bersama-sama untuk meraih mimpi, terimakasih atas kenang-kenangan indah yang dirajut bersama.
10. Keluarga besar IMM UIN Maliki Malang khususnya komisariat Revivalis.
11. Keluarga besar TAP Telkomsel Branch Malang khususnya Batch 2 2016.
12. Keluarga “Familytim” yang dari awal perkuliahan sampe akhir selalu bersama-sama dalam menyelesaikan kewajiban menempuh starata satu.
13. Semua pihak yang tidak dapat penulis sebutkan satu-persatu atas bantuan, masukan, dukungan serta motivasi kepada penulis.

Harapan penulis semoga semua amal kebaikan dan jasa-jasa dari semua pihak yang telah membantu hingga skripsi ini selesai diterima oleh Allah SWT, serta mendapatkan balasan yang lebih baik dan berlipat ganda.

Penulis juga menyadari bahwa skripsi ini masih jauh dari kesempurnaan yang disebabkan keterbatasan Harapan penulis, semoga karya ini bermanfaat dan menambah ilmu pengetahuan bagi kita semua, Aamiin.

Malang, 17 Februari 2017

Penulis

Ihya Isnani Muharromah

DAFTAR ISI

HALAMAN PENGAJUAN.....	ii
LEMBAR PERSETUJUAN	iii
HALAMAN PENGESAHAN.....	iv
HALAMAN PERNYATAAN	v
HALAMAN MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xiv
ABSTRAK	xv
ABSTRACT.....	xvi
ملخص.....	xvii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	3
1.3 Tujuan	4
1.4 Batasan Masalah.....	4
1.5 Manfaat Penelitian	4
BAB 2 TINJAUAN PUSTAKA	5
2.1 <i>Game</i>	5
2.2 Ilmu Sejarah	7
2.3 <i>Non Player Character</i>	7
2.4 <i>Finite State Machine</i>	8
2.5 Logika Fuzzy.....	9
2.6 Fungsi Keanggotaan.....	11

2.7 Fuzzy Sugeno	13
2.8 Algoritma Djikstra	15
2.9 Unity.....	17
2.10 Penelitian Terkait	18
2.11 Metode Penelitian.....	19
BAB 3 DESAIN GAME.....	22
3.1 Skenario dan Perancangan FSM	22
3.1.1 Keterangan Umum <i>Game</i>	22
3.1.2 Deskripsi karakter	23
3.1.3 Story Board <i>Game</i>	24
3.2 <i>Finite State Machine</i>	28
3.3 Perancangan Fuzzy Logic	28
3.4 Algoritma Djikstra	44
BAB 4 IMPLEMENTASI DAN PEMBAHASAN	49
4.1 Implementasi	49
4.1.1 Kebutuhan Perangkat Keras.....	49
4.1.2 Kebutuhan Perangkat Lunak.....	49
4.2 Pengujian Algoritma Fuzzy.....	50
4.2.1 Pengujian Algoritma Fuzzy Sugeno	55
4.3 Pengujian Algoritma Djikstra	62
4.4 Implementasi <i>Game</i>	65
4.5 Integrasi dalam Islam	69
BAB 5 PENUTUP	72
5.1 Kesimpulan	72
5.2 Saran.....	72
DAFTAR PUSTAKA	74

DAFTAR GAMBAR

Gambar 2. 1 Representasi Linier Naik	12
Gambar 2. 2 Representasi Linier Turun	12
Gambar 2. 3 Representasi Kurva Segitiga	12
Gambar 2. 4 Representasi Kurva Trapesium	13
Gambar 2. 5 Tahapan Penelitian	20
Gambar 3. 1 Karakter Utama	23
Gambar 3. 2 Karakter NPC	24
Gambar 3. 3 <i>Item</i> Buku	24
Gambar 3. 4 <i>Item</i> Tas	24
Gambar 3. 5 <i>Scene</i> main menu	25
Gambar 3. 6 <i>Scene</i> play game	26
Gambar 3. 7 <i>Scene</i> next stage	26
Gambar 3. 8 <i>Mission complete</i>	27
Gambar 3. 9 Rancangan FSM	28
Gambar 3. 10 Fuzzyfikasi	29
Gambar 3. 11 Grafik <i>Input</i> Variabel Waktu	30
Gambar 3. 12 Grafik <i>Input</i> Variabel Poin	32
Gambar 3. 13 Grafik <i>Input</i> Variabel Nyawa	33
Gambar 3. 14 Graf Sederhana Sebagai Contoh	44
Gambar 4. 1 Tampilan Simulasi <i>Output</i> pada Matlab	61
Gambar 4. 2 Sumbu Kartesian Untuk Waktu dan Poin	61
Gambar 4. 3 Sumbu Kartesian Untuk Nyawa dan Waktu	61
Gambar 4. 4 Sumbu Kartesian Untuk Poin dan Nyawa	62
Gambar 4. 5 Contoh Posisi Awal NPC	63
Gambar 4. 6 NPC Menuju Target	63

Gambar 4. 7 NPC Bertemu Dengan Target	63
Gambar 4. 8 <i>Splash Screen</i>	65
Gambar 4. 9 <i>Menu Game</i>	66
Gambar 4. 10 <i>Stage Game</i>	66
Gambar 4. 11 <i>Konten Game</i>	66
Gambar 4. 12 <i>Next stage</i>	67
Gambar 4. 13 <i>Mission Complete</i>	67
Gambar 4. 14 <i>Game Over</i>	68
Gambar 4. 15 <i>Player</i>	68
Gambar 4. 16 NPC	68
Gambar 4. 17 <i>Buku</i>	69
Gambar 4. 18 <i>Tas</i>	69

DAFTAR TABEL

Tabel 3. 1 Pencarian Pertama.....	44
Tabel 3. 2 Pencarian Kedua	45
Tabel 3. 3 Pencarian Ketiga	46
Tabel 3. 4 Pencarian Keempat	46
Tabel 3. 5 Pencarian Kelima.....	47
Tabel 3. 6 Pencarian Keenam.....	47
Tabel 3. 7 Pencarian Ketujuh.....	48
Tabel 4. 1 Pengujian Algoritma Fuzzy pada Matlab	55
Tabel 4. 2 Pengujian Algoritma Fuzzy pada <i>Game</i>	58
Tabel 4. 3 Tabel Jarak Tiap <i>Node</i>	64

ABSTRAK

Ihya Isnani Muharromah, **Pergerakan NPC (Non Player Character) Terhadap Player Menggunakan Algoritma Dijkstra Dan Fuzzy Logic Sugeno Untuk Mengatur Bentuk Bonus Pada Game 3D “Muslim Kid”**

Pembimbing I : Yunifa Miftachul Arif, M.T

Pembimbing II : Roro Inda Melani, M.T

Kata kunci : *Game*, kisah Nabi dan Rasul, Algoritma Dijkstra, Fuzzy Sugeno

Game berasal dari kata berbahasa Inggris yang berarti permainan atau pertandingan, atau bisa diartikan sebagai aktifitas terstruktur yang biasanya dilakukan untuk bersenang-senang. *Game* yang dibuat oleh penulis adalah *game adventure* dan terdapat unsur edukasinya yaitu pembelajaran tentang kisah Nabi dan Rasul di dalamnya. Dengan menggunakan metode Algoritma Dijkstra untuk mengatur pergerakan NPC terhadap *player* yaitu menentukan jarak terpendek untuk menghalangi *player* dalam menyelesaikan misi dan Fuzzy Sugeno digunakan untuk mengatur bentuk bonus yang diperoleh *player*. Berdasarkan hasil pengujian kedua metode dapat diterapkan dalam *game “Muslim Kid”* dengan keberhasilan untuk Algoritma Fuzzy Sugeno 85% dan untuk Algoritma Dijkstra 90%.

ABSTRACT

Ihya Isnani Muharomah, **The Movement of NPC (Non Player Character) Against Player Using Fuzzy Logic Algorithms Dijkstra And Sugeno To Set Bonus Game In 3D Shape "Muslim Kid"**

Supervisor I: Yunifa Miftachul Arif, M.T

Supervisor II: Roro Inda Melani, M.T

Keywords: Game, the story of Prophet and Apostle, Dijkstra Algorithm, Fuzzy Sugeno

The game comes from the United Kingdom language which means game or match, or it could be defined as a structured activity usually done for fun. The games created by the author is a series of adventure games and there are elements of hiseducation that is learning about the story of prophets and Apostles in it. Dijkstra's algorithm by using method to regulate the movement of NPC against a player that is the shortest distance to determine the blocking player in completing the mission and Sugeno Fuzzy used to set the form of bonuses earned players. Based on the test results of both methods can be applied in-game "Muslim Kid" with the success of Fuzzy Algorithms for 85% and Sugeno Algorithm Dijkstra to 90%.

ملخص

إحيا اثنين محرمة، حركة *NPC (Non Player Character)* إلى لاعب باستخدام خوارزمية جيكسترا و أجعد منطق سوغينو لتنظيم شكل علاوة في لعب ثلاث البعد "Muslim Kid"

المشرفة 1: يونيفا مفتاح العارف الماجستير

المشرفة 2: رورو إندا مولاني الماجستير

كلمات أساسية: لعب، قصة الأنبياء والرسول، خوارزمية جيكسترا، أجعد سوغينو

Game مأخوذ من كلمة اللغة الإنجليزية بمعنى اللعب أو المسابقة، أو الأنشطة المنظمة للتسلية. اللعب الذي صنعه الباحثة هو اللعب المغامرة وفيه التربية أي التعليم عن قصة الأنبياء والرسول. يستخدم البحث طريقة خوارزمية جيكسترا لتنظيم حركة *NPC* إلى لاعب وهي تعيين أقصر المسافة لردع اللاعب في إنهاء المهمة وأجعد سوغينو مستخدم لتنظيم شكل العلاوة المحصولة للاعب. اعتمادا على نتيجة التجربة كلتا الطريقتين المطبقة في لعب "Muslim Kid" بنجاح لخوارزمية أجعد سوغينو 85% ولخوارزمية جيكسترا 90%.

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Sejarah dan peradaban Islam merupakan bagian penting yang tidak terpisahkan dari kehidupan umat Islam dari masa ke masa. Dengan memahami sejarah secara baik dan benar umat Islam dapat mengambil pelajaran dan membenahi kekurangan atau kesalahan untuk meraih kemuliaan di dunia maupun di akhirat. Mempelajari pengetahuan tentang 25 Nabi dan Rasul memberikan kita pengetahuan tentang bagaimana Nabi dan Rasul terdahulu dalam berdakwah menyiarkan agama Islam dan menyeru untuk beriman kepada Allah SWT. Dengan mengetahui tentang sejarah umat terdahulu kita bisa menerapkan dalam kehidupan sehari-hari, mencontoh kebajikannya dan membenahi kekurangannya.

Islam juga menjadikan iman kepada Nabi dan Rasul sebagai salah satu rukun Islam. Allah SWT berfirman :

قُولُوا آمَنَّا بِاللَّهِ وَمَا أُنزِلَ إِلَيْنَا وَمَا أُنزِلَ إِلَىٰ إِبْرَاهِيمَ وَإِسْمَاعِيلَ وَإِسْحَاقَ وَيَعْقُوبَ وَالْأَسْبَاطِ وَمَا أُوتِيَ مُوسَىٰ وَعِيسَىٰ وَمَا أُوتِيَ النَّبِيُّونَ مِن رَّبِّهِمْ لَا نُفَرِّقُ بَيْنَ أَحَدٍ مِّنْهُمْ وَنَحْنُ لَهُ مُسْلِمُونَ -

۱۳۶-

Artinya : *“Katakanlah (hai orang-orang mukmin): "Kami beriman kepada Allah dan apa yang diturunkan kepada kami, dan apa yang diturunkan kepada Ibrahim, Ismail, Ishaq, Yaqub dan anak cucunya, dan apa yang diberikan kepada Musa dan Isa serta apa yang diberikan kepada Nabi-Nabi dari Tuhannya. Kami tidak*

membeda-bedakan seorangpun diantara mereka dan kami hanya tunduk patuh kepada-Nya.” (QS. Al- Baqarah:136).

Orang-orang muslim beriman kepada semua Nabi Allah dan menghormati serta memuliakan mereka, maka barangsiapa ingkar kepada salah seorang Nabi dan membenci seorang pun diantara Nabi dan Rasul yang dimuat dalam Al-Qur'an, berarti ia tidak beriman kepada Al-Qur'an.

Menurut Santrock (2006:273) *game* merupakan suatu kegiatan menyenangkan yang dilaksanakan untuk kepentingan kegiatan itu sendiri. Erikson dan Freud berpendapat bahwa bermain merupakan suatu bentuk penyesuaian diri manusia yang sangat berguna menolong anak menguasai kecemasan dan konflik. Begitu juga Piaget memandang bahwa bermain sebagai suatu metode yang meningkatkan perkembangan kognitif anak-anak.

Non-Player Character (NPC) pada *game* biasanya memerlukan pergerakan untuk dapat melawan karakter utama tiap levelnya. Dan biasanya juga seorang pembuat *game* menentukan pergerakannya secara statis di mana rute dari pergerakan NPC tersebut akan selalu sama secara berulang, kelemahannya NPC ini tidak memiliki tujuan dan sangat mudah terjebak. Penanggulangannya adalah dengan menggunakan kecerdasan buatan pada pergerakannya, dimana nantinya karakter yg mempunyai kecerdasan buatan ini akan bergerak dengan cara mencari kondisi *player* utama sesuai perintah yang di masukkan.

Dinamai menurut penemunya, seorang ilmuwan komputer, Edsger Dijkstra, adalah sebuah Algoritma rakus (greedy algorithm) seperti yang diulas pada buku *Algorithmics* (Brassard and Bratley [1988, pp. 87-92]). Algoritma ini dipakai

dalam memecahkan permasalahan jarak terdekat (shortest path problem) untuk sebuah graf berarah (*di-rected graph*) dengan bobot-bobot sisi (edge weights) yang bernilai tak-negatif.

Logika Fuzzy pertama kali diperkenalkan oleh Prof.Lotfi A.Zadeh pada tahun 1965. Dasar logika Fuzzy adalah teori himpunan Fuzzy. Pada teori himpunan Fuzzy, peranan derajat keanggotaan atau derajat keanggotaan atau *membership function* menjadi ciri utama dalam penalaran dengan logika Fuzzy tersebut (Kusuma Dewi, 2003).

Game yang akan dibuat adalah *game* edukasi khususnya untuk pembelajaran pengetahuan tentang 25 Nabi dan Rasul untuk anak SD kelas 4, 5 dan 6 dengan menggunakan Algoritma Dijkstra untuk menentukan jarak terdekat NPC dalam menyerang *player* utama. Fuzzy Logic Sugeno untuk mengatur bonus yang diperoleh *player* dalam menyelesaikan misi sesuai dengan aturan-aturan yang telah ditentukan.

1.2 Identifikasi Masalah

Masalah baru yang harus diselesaikan berdasarkan latar belakang diatas adalah bagaimana menerapkan Algoritma Dijkstra dan Fuzzy Logic metode Sugeno dalam *game* yang di gunakan untuk mengetahui jarak terdekat NPC dalam menyerang *player* dan mengatur bonus yang diperoleh *player*.

1.3 Tujuan

Tujuan dari penelitian ini adalah menerapkan Algoritma Dijkstra dan Fuzzy Logic metode Sugeno pada NPC dan konten bonus untuk *player*.

1.4 Batasan Masalah

- a. *Game* ini dimainkan *single player*.
- b. *Game* ini bergenre *adventure*.
- c. *Game* ditujukan untuk anak SD kelas 4, 5, dan 6.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah *game* yang dibuat memiliki NPC yang dapat mengetahui jarak terdekat untuk menyerang *player* utama sesuai dengan aturan-aturan yang telah ditetapkan.

BAB 2

TINJAUAN PUSTAKA

2.1 *Game*

Game berasal dari kata berbahasa Inggris yang berarti permainan atau pertandingan, atau bisa diartikan sebagai aktifitas terstruktur yang biasanya dilakukan untuk bersenang-senang. *Game* bermakna “permainan”, teori permainan adalah suatu cara belajar yang digunakan dalam menganalisa sejumlah pemain maupun perorangan yang menunjukkan strategi-strategi rasional. Teori permainan pertama kali ditemukan oleh sekelompok ahli matematika pada tahun 1944. Teori tersebut dikemukakan oleh John von Neumann dan Oskar Morgenstren (1994), menurut mereka permainan terdiri atas sekumpulan peraturan yang membangun situasi bersaing dari dua sampai beberapa orang atau kelompok dengan memilih strategi yang dibangun untuk memaksimalkan kemenangan sendiri atau pun untuk meminimalkan kemenangan lawan. Macam-macam *game*, antara lain:

a. Aksi

Game ini membutuhkan kemampuan refleks pemain. Salah satu *subgenre action* yang populer adalah *First Person Shoter* (FPS). Pada *game* FPS diperlukan kecepatan berfikir. *game* ini dibuat seolah-olah pemain yang berada dalam suasana tersebut.

b. Aksi Petualangan

Game ini memadukan *game genre* aksi dan petualangan. Sebagai contoh pemain diajak untuk menelusuri gua bawah tanah sambil mengalahkan musuh dan mencari artefak kuno atau menyebrangi sungai.

c. Strategi

Genre strategi menitikberatkan pada kemampuan pada kemampuan berpikir dan organisasi. *Game* strategi dibedakan menjadi dua, yaitu *Turn Based Strategy* dan *Real Time Strategy*. Jika *real time* strategi mengharuskan pemain membuat keputusan dan secara bersamaan pihak lawan juga beraksi hingga menimbulkan serangkaian kejadian dalam waktu yang sebenarnya, sedangkan *turn based* strategi pemain bergantian menjalankan taktiknya. Saat pemain mengambil langkah, pihak lawan menunggu. Demikian juga sebaliknya.

d. Simulasi, Konstruksi dan Manajemen

Pemain dalam *game* ini diberikan keleluasaan untuk membangun dan suatu proyek tertentu dengan bahan baku yang terbatas dan sudah ditentukan.

e. *Puzzle*

Game puzzle menyajikan teka-teki, menyamakan warna, perhitungan matematika, menyusun balok, atau mengenal huruf dan gambar.

f. Permainan Kata

Game ini sering dirancang untuk menguji kemampuan dengan bahasa atau untuk lebih mengeksplere sifat-sifatnya.

g. Olahraga

Game ini membuat seolah-olah olahraga bisa dimainkan ke dalam sebuah aplikasi. Biasanya dibuat semirip mungkin dengan kondisi olahraga yang sebenarnya.

h. Balapan

Sesuai dengan namanya *game* ini mengusung konsep balapan pada kehidupan nyata. Pemain dapat memilih kendaraan yang akan digunakan dll sesuai dengan balapan pada sebenarnya dengan tujuan untuk mencapai *finish* tercepat.

2.2 Ilmu Sejarah

Secara terminologi sejarah adalah keadaan atau peristiwa penting yang terjadi di masa lampau dan benar-benar terjadi pada individu dan masyarakat sesuai pada kenyataan-kenyataan alam dan manusia yang telah terjadi. Dalam *Game* yang akan dibuat, ilmu sejarah yang akan di angkat adalah sejarah atau kisah 25 Nabi dan Rasul. Oleh karena itu *game* ini bertujuan untuk mengenalkan atau memberi pengetahuan kepada anak-anak SD kelas 4,5 dan 6 pengetahuan tentang 25 Nabi dan Rasul.

2.3 Non Player Character

Non Player Character kemudian di tulis sebagai NPC disebut juga *autonomous character* atau *autonomous agent* yaitu karakter yang terdapat dalam *game* atau *virtual reality* dimana memiliki kemampuan untuk melakukan gerakan secara otonom atau tidak dikontrol secara *real time* oleh pemain. Secara garis besar NPC dapat diartikan sebagai karakter dalam *game* yang dikendalikan oleh komputer dan tidak dapat dikendalikan oleh pemain, untuk itu pengendalian NPC umumnya menggunakan kecerdasan buatan. Dengan kecerdasan buatan yang diberikan maka NPC dapat melakukan pekerjaan seperti yang dapat dilakukan

oleh manusia dan membuat *game* lebih menarik serta variatif. Penelitian tentang kecerdasan buatan pada NPC dalam *game*, hingga saat ini masih terus dikembangkan. Kecerdasan buatan tersebut di kembangkan untuk merancang perilaku NPC. (JimHyuk, 2005)

2.4 *Finite State Machine*

Finite state machine merupakan model komputasi, dan sebagai sebagai teknik permodelan AI, FSM terdiri dari 4 komponen: *State*, merupakan kondisi yang mendefinisikan perilaku sekaligus memungkinkan terjadinya aksi; *State transition*, yang memicu perpindahan *state*; *Rules*, atau kondisi yang harus dipenuhi untuk dapat memicu transisi *state*; *Input events*, yang bisa dipicu secara *internal* ataupun *eksternal* yang memicu tercapainya *rule* dan mengarahkan terjadinya transisi *state*. Jika kemudian dikelompokkan dalam pembagian *set*, maka *FSM* terdiri dari: *state set*, *inputset*, *output set*, dan *fungsi state transition*. (Yunifa dkk : 2012)

Dalam *FSM*, istilah *state* merupakan konsep yang sangat penting karena menyajikan informasi yang berkaitan dengan keadaan sistem saat sebelumnya. Dalam satu periode yang tetap, sistem berada dalam satu *state*, yang tiap *state*-nya mempunyai karakteristik perilaku dan aksi yang spesifik (yang sudah ditentukan). *State-state* dihubungkan melalui transisi antar *state*, selanjutnya masing-masing transisi mengarahkan ke *state* (kondisi) selanjutnya sebagai *target state*. Akan selalu ada *initial state* yang berfungsi sebagai *starting point*, lalu kondisi “saat ini” (*current state*) yang menyimpan informasi *state* sebelumnya. *Input event* (kejadian), yang bisa dipicu secara eksternal maupun internal sistem, berfungsi

sebagai pemicu (*trigger*), yang mengarahkan pada proses evaluasi dari *rule* (aturan). Jika kondisi dan syaratnya terpenuhi, maka terjadi transisi dari *state* saat ini ke *state* selanjutnya sesuai dengan *rule* yang sudah ada. *Transitions* merupakan *class* dari obyek yang mengatur sistem.

Transitions mengontrol aliran dari eksekusi dengan melakukan *setting* pada *state* yang sedang aktif dari *state machine* melalui penggunaan dari kondisi. *Transitions* bisa berlaku *one-to-many*, dengan kata lain, bisa terjadi ada satu *state* terhubung ke sisi *input* dari *transitions*, dan beberapa *state* terhubung ke sisi *output* dari *transitions*, tergantung dari kondisi yang sedang berlangsung dalam *transition* tersebut.

2.5 Logika Fuzzy

Logika Fuzzy adalah sebuah metode untuk menangani masalah ketidakpastian. Yang dimaksud dengan ketidakpastian yaitu suatu masalah yang mengandung keraguan, ketidaktepatan, kurang lengkapnya informasi, dan nilai kebenarannya bersifat sebagian. (Suyanto, 2011). Logika Fuzzy berbeda dengan logika digital biasa, dimana logika digital biasa hanya mengenal dua keadaan yaitu: Ya dan Tidak atau *ON* dan *OFF* atau *High* dan *Low* atau "1" dan "0". Sedangkan Logika Fuzzy meniru cara berpikir manusia dengan menggunakan konsep sifat kesamaran suatu nilai. Ada beberapa alasan digunakanya logika Fuzzy (Kusuma Dewi, 2003), diantaranya :

- a. Konsep logika Fuzzy mudah dimengerti. Konsep matematis yang mendasari penalaran Fuzzy sangat sederhana dan mudah dimengerti.
- b. Logika Fuzzy sangat fleksibel.

- c. Logika Fuzzy memiliki toleransi terhadap data-data yang tidak tepat.
- d. Logika Fuzzy mampu memodelkan fungsi-fungsi *non linear* yang sangat kompleks.
- e. Logika Fuzzy dapat bekerjasama dengan teknik-teknik kendali secara konvensional.
- f. Logika Fuzzy didasarkan pada bahasa alami.

Ada beberapa hal yang perlu diketahui dalam memahami logika Fuzzy (Kusuma Dewi, 2003) yaitu :

a. Variabel Fuzzy

Variabel Fuzzy merupakan variabel yang akan dibahas dalam suatu sistem Fuzzy. Variabel tidak harus mewakili angka saja tetapi juga dapat mewakili benda atau tempat. Contoh: umur, temperatur, permintaan, dsb.

b. Himpunan Fuzzy

Himpunan Fuzzy merupakan suatu kelompok yang mewakili kondisi atau keadaan tertentu dalam suatu variabel Fuzzy. Contoh :

- Variabel jarak, terbagi menjadi 3 himpunan Fuzzy, yaitu : dekat, sedang, dan jauh

Himpunan Fuzzy terdiri dari 2 atribut , yaitu :

1. *Linguistik*, yaitu nama suatu kelompok yang mewakili suatu keadaan tertentu dengan menggunakan bahasa alami, sebagai contoh : Muda, Parobaya, Tua
2. *Numeris*, yaitu suatu nilai yang menunjukkan ukuran dari suatu variable, misalnya 5, 15, 30 dan sebagainya.

c. Semesta pembicaraan

Semesta pembicaraan merupakan nilai yang diperbolehkan dioperasikan dalam suatu variabel Fuzzy. Sebagai contoh :

- Semesta pembicaraan temperatur : $[0 \ 40]$

d. Domain

Domain himpunan Fuzzy merupakan keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan Fuzzy. Misalnya domain himpunan Fuzzy : Muda = $[0 \ 40]$, Parobaya = $[35 \ 50]$, Tua = $[40 \ +\infty]$ dst.

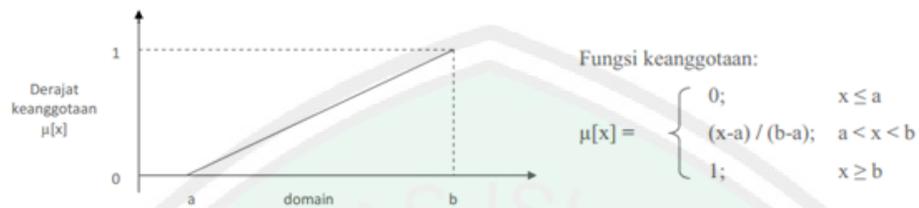
2.6 Fungsi Keanggotaan

Fungsi Keanggotaan adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data ke dalam nilai keanggotaannya (derajat keanggotaan) yang memiliki *interval* antara 0 sampai 1. (Kusuma Dewi, 2013). Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Ada beberapa fungsi yang bisa digunakan diantaranya :

a) Representasi Linear

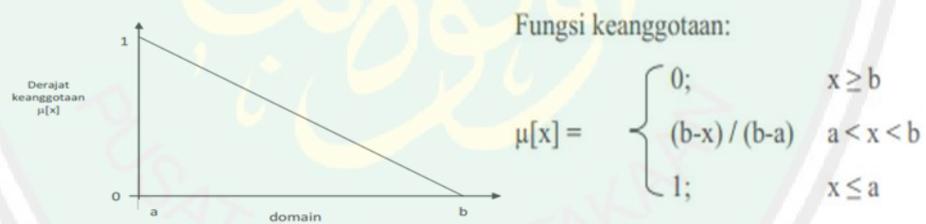
Pada representasi linear, pemetaan *input* ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang sedikit untuk membahayai suatu konsep yang kurang jelas. Ada 2 grafik keanggotaan linier. Pertama, grafik keanggotaan kurva linier naik, yaitu 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. (T.Sutojo dkk, 2011).



Gambar 2. 1 Representasi Linier Naik

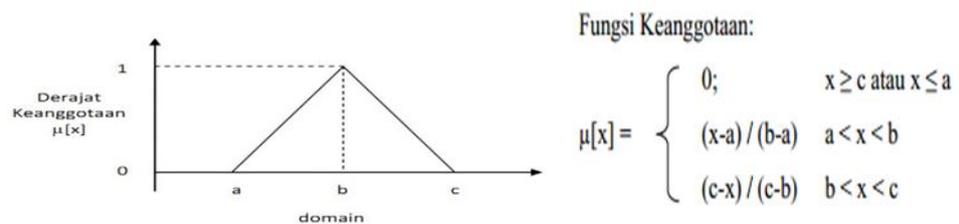
Kedua, merupakan kebalikan dari yang pertama. Garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah.



Gambar 2. 2 Representasi Linier Turun

b) Representasi Kurva Segitiga

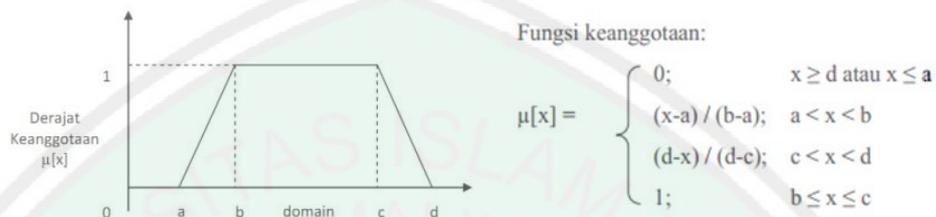
Kurva segitiga pada dasarnya merupakan gabungan antara 2 garis (linear) :



Gambar 2. 3 Representasi Kurva Segitiga

c) Representasi Kurva Trapesium

Kurva trapesium pada dasarnya seperti bentuk segitiga, hanya saja ada titik yang memiliki nilai keanggotaan.



Gambar 2. 4 Representasi Kurva Trapesium

2.7 Fuzzy Sugeno

Fuzzy metode Sugeno merupakan metode inferensi Fuzzy untuk aturan yang direpresentasikan dalam bentuk *IF – THEN*, dimana *output* (konsekuen) sistem tidak berupa himpunan Fuzzy, melainkan berupa konstanta atau persamaan linear. Model Sugeno menggunakan fungsi keanggotaan Singleton yaitu fungsi keanggotaan yang memiliki derajat keanggotaan 1 pada suatu nilai *crisp* tunggal dan 0 pada nilai *crisp* yang lain (Kusuma Dewi, 2003). Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985.

Proses Fuzzy inference dapat dibagi dalam lima bagian, yaitu :

- Fuzzyfikasi *Input* : FIS mengambil masukan masukan dan menentukan derajat keanggotaannya dalam semua Fuzzy *set*.
- Operasi logika Fuzzy : Hasil akhir dari operasi ini adalah derajat kebenaran *antecedent* yang berupa bilangan tunggal.
- Implikasi : Merupakan proses mendapatkan *consequent* atau keluaran sebuah *IF-THEN rule* berdasarkan derajat kebenaran *antecedent*. Proses ini menggunakan mengambil nilai MIN/terkecil

dari dua bilangan : Hasil operasi *Fuzzy logic OR* dan *Fuzzy set* banyak.

- Agregasi :Yaitu proses mengkombinasikan keluaran semua *IF-THEN rule* menjadi sebuah *Fuzzy set* tunggal. Pada dasarnya agregasi adalah operasi *Fuzzy logic OR* dengan masukannya adalah semua *Fuzzy set*.
- DeFuzzyfikasi : Keluaran dari defuzzyfikasi adalah sebuah bilangan tunggal, cara mendapatkannya ada beberapa versi, yaitu *centroid*, *bisector*, *middle of maximum*, *largest of maximum* dan *smallest of maximum*.

Penalaran dengan metode SUGENO hampir sama dengan penalaran MAMDANI,hanya saja *output* (konsekuen) sistem tidak berupa himpunan Fuzzy, melainkan berupa konstanta atau persamaan linear.

a. Module Fuzzy Sugeno Orde-Nol

$$IF (x_1 \text{ is } A_1) * (x_2 \text{ is } A_2) * \dots * (x_n \text{ is } A_n) THEN z = k$$

Dengan A_i adalah himpunan Fuzzy ke-i sebagai antesenden, dan k adalah suatu konstanta (tegas) sebagai konsekuen.

b. Model Fuzzy Sugeno Orde-Satu

$$IF (x_1 \text{ is } A_1) * (x_2 \text{ is } A_2) * \dots * (x_n \text{ is } A_n) THEN z = p_1 * z_1 + \dots + p_n * x_n + q$$

Dengan A_i adalah konstanta Fuzzy ke-i sebagai antesenden, dan p_i adalah suatu konstanta (tegas) ke-i dan q juga merupakan konstanta dalam konsekuen.

2.8 Algoritma Dijkstra

Algoritma Dijkstra diterapkan untuk mencari lintasan terpendek pada graf berarah. Namun, Algoritma ini juga benar untuk graf tak berarah. Algoritma Dijkstra mencari lintasan terpendek dalam sejumlah langkah. Algoritma ini menggunakan prinsip *greedy*. Prinsip *greedy* pada Algoritma Dijkstra menyatakan bahwa pada setiap langkah kita memilih sisi yang berbobot minimum dan memasukkannya dalam himpunan solusi. (Rinaldi Munir, 2005)

Contoh penerapan Algoritma Dijkstra adalah lintasan terpendek yang menghubungkan antara dua kota berlainan tertentu (*Single-source Single-destination Shortest Path Problems*). Cara kerja Algoritma Dijkstra memakai strategi *greedy*, dimana pada setiap langkah dipilih sisi dengan bobot terkecil yang menghubungkan sebuah simpul yang sudah terpilih dengan simpul lain yang belum terpilih. Algoritma Dijkstra membutuhkan parameter tempat asal, dan tempat tujuan. Hasil akhir dari Algoritma ini adalah jarak terpendek dari tempat asal ke tempat tujuan beserta rutenya.

Misalkan sebuah graf berbobot dengan n buah simpul dinyatakan dengan matriks ketetanggaan $M = [m_{ij}]$ yang dalam hal ini,

$$m_{ij} = \text{bobot sisi } (i,j) \quad (\text{pada graf tak-berarah } m_{ij} = m_{ji})$$

$$m_{ij} = 0$$

$$m_{ij} = \infty, \text{ jika tidak ada sisi dari simpul } i \text{ ke simpul } j$$

Selain matriks M , kita juga menggunakan tabel $S = [s_i]$ yang dalam hal ini,

$$s_i = 1, \text{ jika simpul } i \text{ termasuk ke dalam lintasan terpendek}$$

$$s_i = 0, \text{ jika simpul } i \text{ tidak termasuk ke dalam lintasan terpendek}$$

dan tabel $D = [d_i]$ yang dalam hal ini,

$d_i = \text{panjang lintasan dari simpul awal } a \text{ ke simpul } i$

Menurut Siang (2009), Algoritma Dijkstra menyelesaikan masalah pencarian jalur terpendek (sebuah lintasan yang mempunyai panjang minimum) dari vertek a ke vertek z dalam graf berbobot, bobot tersebut adalah bilangan positif jadi tidak dapat dilalui oleh *node* negatif. Misalkan G adalah graf berarah berlabel dengan titik-titik $V(G) = \{v_1, v_2, \dots, v_n\}$ dan path terpendek yang dicari adalah dari v_1 ke v_n . Algoritma Dijkstra dimulai dari titik v_1 . Dalam iterasinya, Algoritma akan mencari satu titik yang jumlah bobotnya dari titik 1 terkecil. Titik-titik yang terpilih dipisahkan, dan titik-titik tersebut tidak diperhatikan lagi dalam iterasi berikutnya.

Langkah-langkah dalam menentukan lintasan terpendek pada Algoritma Dijkstra yaitu :

1. Memilih *node* sumber sebagai *node* awal, diinisiasikan dengan '1'.
2. Bentuk tabel yang terdiri dari *node-node*, status, bobot, dan *predocessor*.
Kemudian melengkapi kolom bobot yang diperoleh dari jarak *node* sumber ke semua *node* yang langsung terhubung dengan *node* sumber tersebut.
3. Jika *node* sumber ditemukan maka tetapkan sebagai *node* terpilih.
4. Tetapkan *node* terpilih dengan label permanen dan perbaharui *node* yang langsung terhubung.
5. Tentukan *node* sementara yang terhubung pada *node* yang sudah terpilih sebelumnya dan merupakan bobot terkecil dilihat dari tabel dan tentukan sebagai *node* terpilih berikutnya.

6. Jika *node* yang terpilih merupakan *node* tujuan maka kumpulan *node* terpilih atau *predecessor* merupakan rangkaian yang menunjukkan lintasan terpendek.

2.9 Unity

Game engine adalah sebuah *software* atau perangkat lunak yang dirancang untuk membuat sebuah *game*. Sebuah *game engine* biasanya dibangun dengan mengenkapsulasi beberapa fungsi standar yang umum digunakan dalam pembuatan sebuah *game*. Misalnya fungsi *rendering*, pemanggilan suara, *network*, atau pembuatan partikel untuk *special effect*. Sebagian besar *game engine* umumnya berupa library atau sekumpulan fungsi-fungsi yang penggunaannya dipadukan dengan bahasa pemrograman. Salah satu *game engine* yang cukup populer adalah Unity.

Unity *Technologies* dibangun di tahun 2004 oleh David Helgason, Nicholas Francis dan Joachim Ante. *Game engine* ini dibangun atas dasar kepedulian mereka terhadap indie *developer* yang tidak bisa membeli *game engine* karena terlalu mahal. Fokus perusahaan ini adalah membuat sebuah perangkat lunak yang bisa digunakan oleh semua orang, khususnya untuk membangun sebuah *game*. Di tahun 2009, Unity diluncurkan secara gratis dan april 2012, Unity mencapai popularitas tertinggi dengan lebih dari 1 juta *developer* terdaftar di seluruh dunia. Versi terakhir dari *game engine* unity ini adalah Unity 5.4.

Selain bisa didapatkan secara gratis, Unity adalah sebuah *game engine* yang memungkinkan *developer* untuk membuat sebuah *game* 3D dengan mudah dan cepat. Secara *default*, Unity telah di atur untuk pembuatan *game* bergenre *First*

Person Shooting (FPS Unity3D.), namun juga bisa di gunakan untuk membuat *game* bergenre *Role Playing Game* (RPG) dan *Real Time Strategi* (RTS). Selain itu, Unity merupakan sebuah *engine multiplatform* yang memungkinkan *game* yang dibangun diperkenalkan untuk berbagai *platform* seperti Windows, Mac, Android, IOS, PS3 dan juga Wii (Rickman Roedavan).

2.10 Penelitian Terkait

Beberapa penelitian yang terkait dengan penelitian yang sedang dikerjakan, diantaranya :

1) Pergantian Senjata NPC pada *Game FPS Menggunakan Fuzzy Sugeno*

Penelitian yang telah dilakukan oleh Yunifa Miftachul Arif, Fachrul Kurniawan Jurusan Teknik Informatika Fakultas Saintek UIN Maulana Malik Ibrahim Malang dan Ady Wicaksono Jurusan Multimedia SMKN 3 Batu. Penelitian ini membahas tentang bagaimana sistem perubahan senjata secara otomatis pada NPC berdasarkan perubahan kondisi lingkungan yang dihadapi. Metode yang digunakan untuk menentukan jenis senjata yang digunakan pada penelitian ini adalah Fuzzy Sugeno. Untuk menghasilkan *output* Fuzzy yang bervariasi, maka digunakan variabel jarak musuh dan jumlah teman. Untuk selanjutnya diujicobakan dalam *game First Person Shooter* menggunakan *Torque Game Engine*. Dalam hasil uji coba terjadi respon pergantian senjata pada masing-masing NPC terhadap kondisi yang dihadapi sesuai dengan *rule* Fuzzy yang telah didesain sebelumnya.

2) Penentuan Jarak Terpendek dan Jarak Terpendek Alternatif Menggunakan Algoritma Dijkstra Serta Estimasi Waktu Tempuh

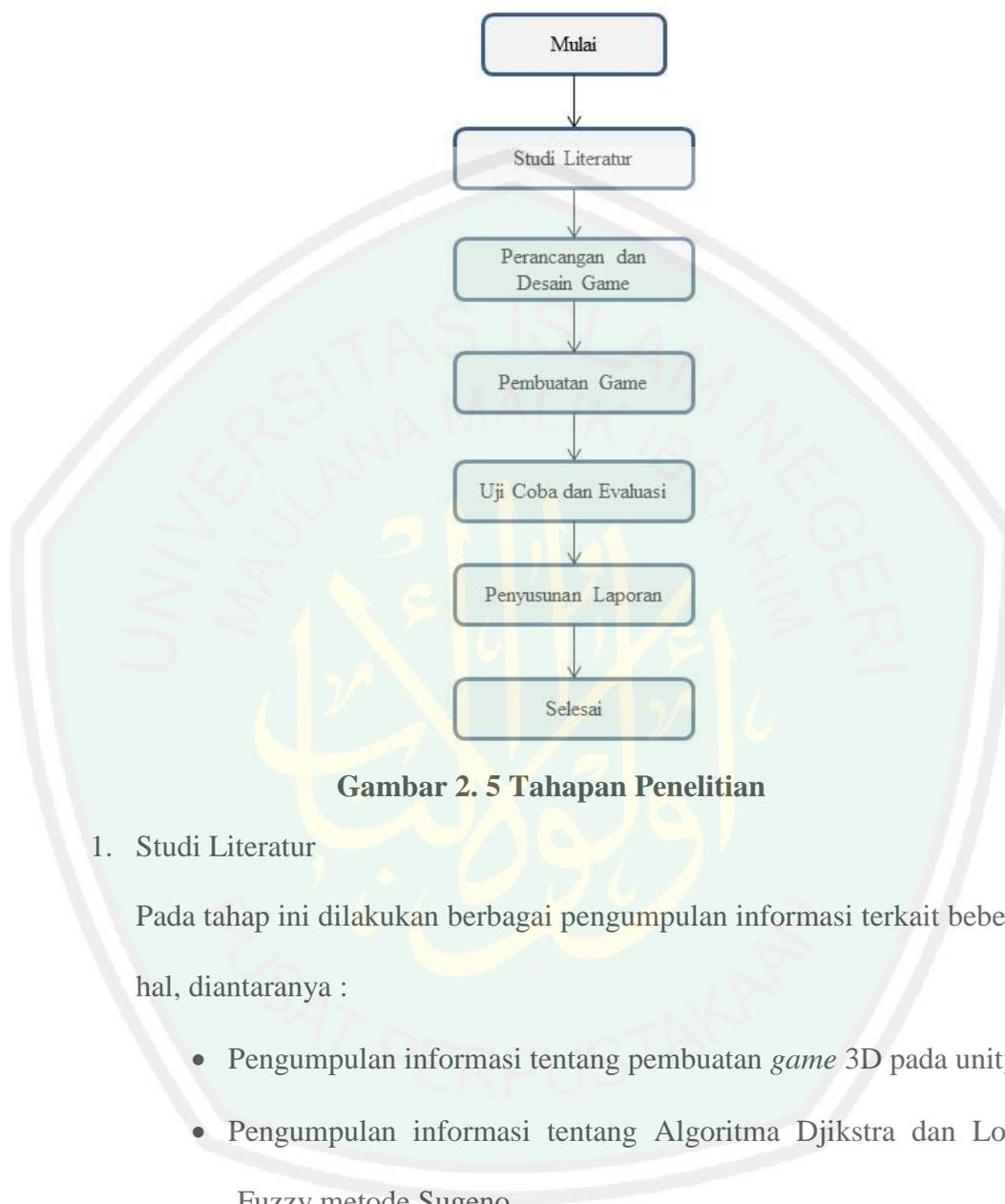
Penelitian yang dilakukan oleh Asti Ratnasari , Farida Ardiani dan Feny Nurvita Teknik Informatika Universitas Islam Indonesia Yogyakarta membahas tentang Implementasi sistem yang dibuat digunakan untuk mencari jarak terpendek, jarak terpendek alternatif serta estimasi waktu tempuh dalam sebuah *graf* dengan menggunakan Algoritma Dijkstra. Hasil yang didapatkan dari implementasi sistem ini adalah mampu menemukan jarak terpendek dan jarak terpendek alternatif ketika terjadi hambatan (pemblokiran jalan) pada jalur terpendek utama dan juga dapat mengetahui estimasi waktu tempuhnya.

3) Implementasi Fuzzy Sugeno Untuk Mengatur Bentuk Bonus Dari Konten Islami (ILMU TAJWID HUKUM BACAAN NUN SUKUN ATAU TANWIN) Pada *Game 3D Battle Jet*

Penelitian yang dilakukan oleh Moh.Ainur Rahman Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang membahas tentang penggunaan logika Fuzzy Sugeno untuk mengatur bentuk bonus yaitu dengan mempertimbangkan beberapa kondisi yaitu nyawa *player*, skor *player* dan level yang dimainkan. Hasil dari penelitian yang telah dilakukan menunjukkan bahwa Fuzzy Sugeno bisa diterapkan untuk mengatur bentuk bonus dari konten Islami secara dinamis sesuai dengan nilai nyawa, skor dan level.

2.11 Metode Penelitian

Penelitian yang dilakukan terdiri dari beberapa tahapan , diantaranya :



Gambar 2. 5 Tahapan Penelitian

1. Studi Literatur

Pada tahap ini dilakukan berbagai pengumpulan informasi terkait beberapa hal, diantaranya :

- Pengumpulan informasi tentang pembuatan *game* 3D pada unity
- Pengumpulan informasi tentang Algoritma Dijkstra dan Logika Fuzzy metode Sugeno
- Pengumpulan informasi tentang penelitian – penelitian terkait
- Pengumpulan informasi tentang kisah atau sejarah 25 Nabi dan Rasul

2. Perancangan dan desain *game*

Perancangan *game* terdiri dari beberapa perancangan yaitu :

- Perancangan *story board game*
- Perancangan *Finite State Machine (FSM) game*
- Perancangan desain lingkungan *game* pada Unity 3D
- Perancangan Fuzzy Sugeno dan Algoritma Dijkstra

3. Pembuatan *game*

Setelah mengumpulkan informasi – informasi yang terkait dengan penelitian ini , dan merancang *game* , tahap selanjutnya adalah pembuatan *game* dengan menggunakan Unity 3D dengan menggunakan bahasa java *script*.

4. Uji coba dan evaluasi

Uji coba dan evaluasi dilakukan untuk mengetahui apakah tahapan – tahapan aplikasi dan pergerakan NPC sesuai setelah dimplementasikan dengan Algoritma Dijkstra dan Fuzzy *logic*.

5. Penyusunan laporan

Penyusunan laporan akhir merupakan dokumentasi dari keseluruhan pelaksanaan penelitian dan diharapkan bermanfaat bagi penelitian lebih lanjut.

BAB 3

DESAIN GAME

3.1 Skenario dan Perancangan FSM

Game ini dirancang untuk membantu pengetahuan tentang 25 Nabi dan Rasul untuk anak SD kelas 4, 5 dan 6 dengan media pembelajaran *game* yang bersifat edukasi. *Game* diharapkan dapat membantu mengenalkan pengetahuan tentang Nabi dan Rasul secara umum. *Game* dirancang dengan sistem pemain tunggal atau *single player* yang digambarkan dengan karakter seorang anak laki-laki muslim yang bernama Ahmad , dan terdapat NPC berupa karakter kelinci yang dapat menghalangi Ahmad untuk menyelesaikan misi. Ahmad harus bisa menyelesaikan misi untuk memenangkan *game*. *Game* ini nantinya dapat dimainkan di PC dengan *grafis* 3 dimensi.

3.1.1 Keterangan Umum Game.

Game ini dirancang dengan model *game* petualangan, *player* diminta untuk menyelesaikan tiap misi yang ada , pada setiap misi terdapat pertanyaan sehingga *player* harus bisa menjawab seluruh pertanyaan pada tiap misi untuk menyelesaikan *game*. Pada tiap misi terdapat pernyataan tentang kisah atau sejarah 25 Nabi dan Rasul, jadi *player* dapat membacanya untuk mempunyai pengetahuan tentang 25 Nabi dan Rasul sebagai inti dari pembelajaran yang ada didalam *game* ini.

Dalam *game* *player* tidak membunuh NPC melainkan hanya menghindar dari kejaran NPC. Karakter Ahmad tidak membunuh karakter kelinci hanya menghindar apabila Ahmad dikejar kelinci.

Unsur edukasi yang diberikan dalam *game* ini adalah pembelajaran pengetahuan tentang kisah 25 Nabi dan Rasul untuk anak-anak SD kelas 4, 5 dan 6 dengan tantangan dan pengetahuan baru yang diberikan kepada pemain *game* ini, diharapkan menjadi pengetahuan untuk menjadikan kisah atau sejarah Nabi dan Rasul sebagai suri tauladan dalam kehidupan sehari-hari. Unsur pembelajaran yang dimasukkan dalam *game* ini diantaranya sebagai berikut :

1. Nama-nama 25 Nabi dan Rasul
2. Nama-nama keluarga 25 Nabi dan Rasul
3. Kisah 25 Nabi dan Rasul mulai dari lahir hingga wafat

3.1.2 Deskripsi karakter

A. Karakter Utama

Karakter utama adalah seorang anak muslim laki-laki yang bernama Ahmad.



Gambar 3. 1 Karakter Utama

B. NPC

Karakter musuh digambarkan dengan hewan kelinci. Sebagai pembangkit *Artificial Intelligence*, Algoritma Dijkstra diimplementasikan pada NPC kelinci.



Gambar 3. 2 Karakter NPC

C. Item Buku



Gambar 3. 3 Item Buku

Merupakan *item* yang harus di ambil oleh *player*, *item* buku berisi pernyataan dan pertanyaan. Ketika *player* dihadapkan dengan pertanyaan maka harus menjawab pertanyaan untuk mendapatkan poin.

D. Item Tas

Item bonus pada *game* ini di simbolkan dengan objek tas. Apabila *player* mengambil *item* tas , maka dia akan memperoleh bonus pada *game*.



Gambar 3. 4 Item Tas

3.1.3 Story Board Game

Pada *game* yang akan dibuat, terdapat beberapa *scene* yaitu *scene Splash screen*, *scene main menu*, *scene play*, seperti di jelaskan pada gambar di bawah ini

:



Gambar 3.5 Scene main menu

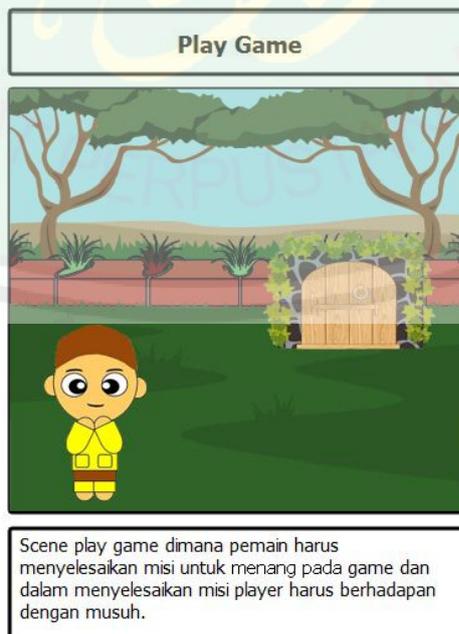
Gambar 3.5, merupakan *scene* menu yang terdapat 2 pilihan yaitu menu *play*, menu *quit*. Jika *player* memilih menu *play* maka akan mulai bermain, jika *player* memilih menu *quit* maka akan keluar dari *game*.

Gambar 3.6, merupakan *scene game play* dimana karakter Ahmad harus menyelesaikan misi yaitu mengambil *item* buku yang berisi pertanyaan tentang 25 Nabi dan Rasul, dalam perjalanan menyelesaikan misi Ahmad akan bertemu dengan kelinci yang akan menghalangi Ahmad karakter utama dalam *game* tersebut. Apabila Ahmad tidak dapat menghindar maka nyawa nya akan berkurang. Terdapat *item* tas yang merupakan konten bonus apabila Ahmad mengambil *item* tas ini akan memperoleh bonus dapat berupa poin atau pun waktu. Saat *player* berada dalam jarak pandang NPC maka NPC akan mencari jalur terpendek untuk menyerang *player*.



Gambar 3. 6 Scene play game

Gambar 3.7 , apabila Ahmad dapat menyelesaikan misi pada *stage* pertama maka dia dapat membuka pintu untuk masuk ke *stage* selanjutnya, jika semua *item* buka belum berhasil diambil maka pintu tidak akan terbuka pintu akan terbuka apabila semua *item* buku telah berhasil diambil.



Gambar 3. 7 Scene next stage



Gambar 3. 8 Mission complete

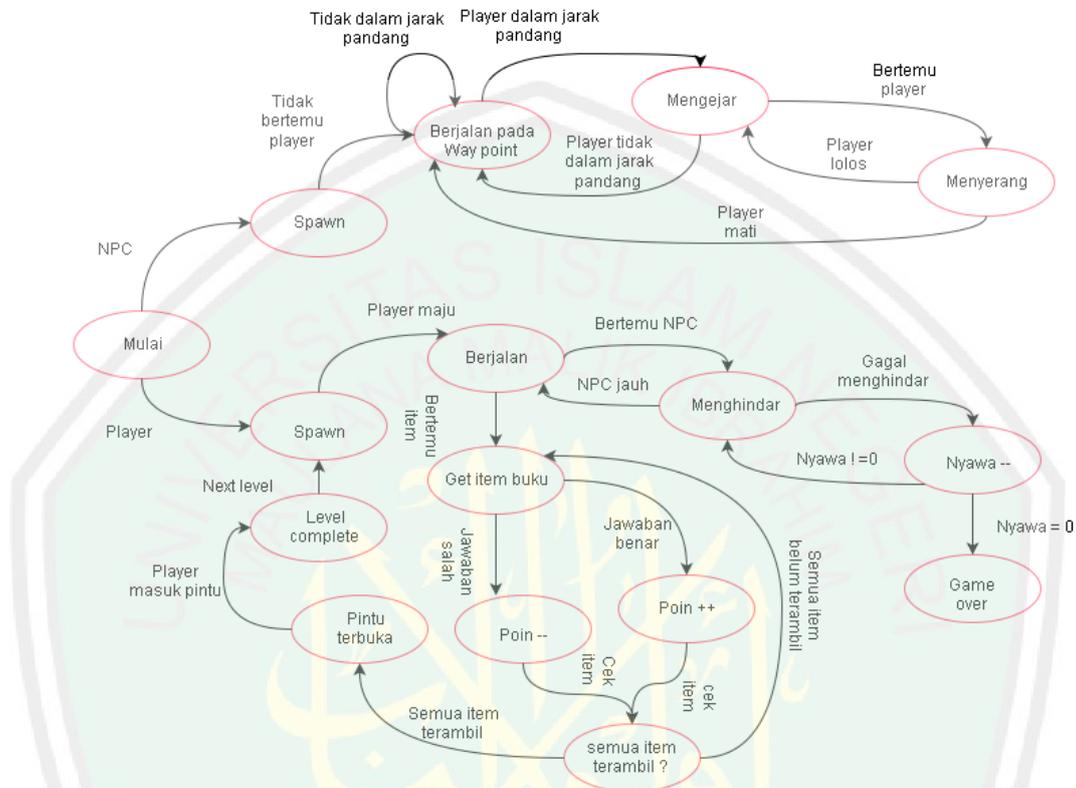
Pada gambar 3.8 menjelaskan jika Ahmad berhasil menyelesaikan semua misi maka dia akan memperoleh piala pada tahap akhir *game*, setelah itu *game* selesai.

Bisa disimpulkan *workflow* dari *game Moslem Kid* yang dibuat yaitu :

- *Splash screen*
- *Main menu* :
 - *Play* : *Player* bermain *game*
 - *Quit* : *Player* keluar dari *game*
- *Game finish*

3.2 Finite State Machine

Berikut adalah gambaran *Finite State Machine* game yang dibuat :



Gambar 3. 9 Rancangan FSM

3.3 Perancangan Fuzzy Logic

Pada *game* ini Fuzzy Sugeno digunakan untuk menentukan “bentuk bonus” dari *item* yang berbentuk tas pada *game Muslim Kid*. Dengan adanya Fuzzy Sugeno, masing-masing *item* tas dapat menjadi bonus berupa skor, nyawa dan yang lainnya sesuai dengan variabel masukan. Tahap – tahapan perancangan Fuzzy Sugeno sebagai berikut :

a. Variabel Fuzzy

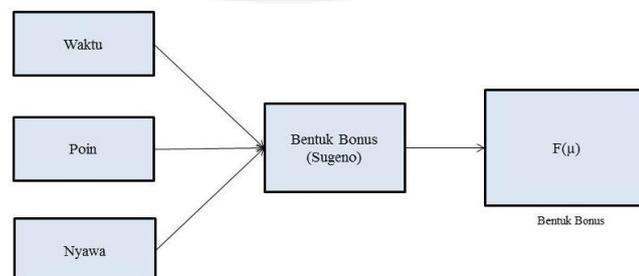
Terdapat 4 variabel dalam fungsi Fuzzy yang digunakan pada *game* ini, yaitu variabel waktu, variabel poin, dan variabel nyawa sebagai *input*.

Sedangkan variabel *outputnya* adalah variabel bentuk bonus dari konten Islam. Dari tiga variabel yang digunakan, maka nilai linguistiknya sebagai berikut:

1. Variable waktu, dibagi menjadi 3 himpunan keanggotaan yaitu : banyak, sedang, sedikit.
2. Variabel poin, dibagi menjadi 3 himpunan keanggotaan yaitu : banyak, sedang, sedikit.
3. Variable nyawa, dibagi menjadi 3 himpunan keanggotaan : bahaya, sedang, aman.
4. Variabel bentuk bonus, dibagi menjadi 2 himpunan keanggotaan : nyawa, poin, waktu

b. Fuzzyfikasi

Fuzzyfikasi adalah proses memetakan nilai *crisp* (numerik) ke dalam himpunan Fuzzy dan menentukan derajat keanggotaannya. Secara garis besar pemetaan nilai *crisp* ke dalam himpunan Fuzzy dijelaskan dengan gambar berikut ini :

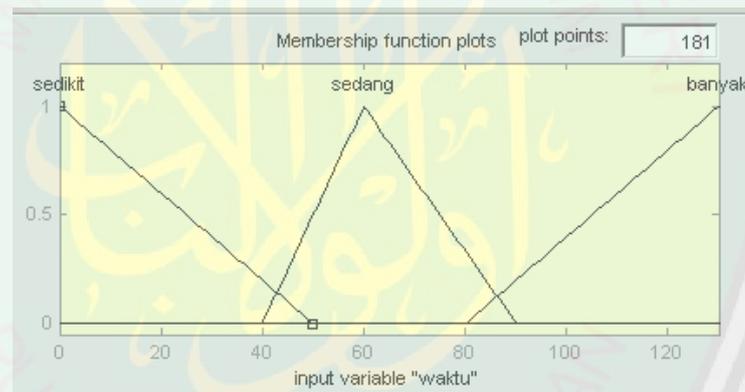


Gambar 3. 10 Fuzzyfikasi

Berdasarkan Fuzzy *Interface* sistem diatas maka pemetaan himpunan Fuzzy adalah sebagai berikut:

1. Variabel waktu dibagi menjadi tiga keanggotaan yaitu : banyak, sedang, dan sedikit. *Range* nilai untuk variabel waktu antara 0- 180 akan dijelaskan sebagai berikut :

- a. Sedikit = 0-50
- b. Sedang = 40-90
- c. Banyak = 80-130



Gambar 3. 11 Grafik *Input* Variabel Waktu

Pada gambar 3.17 menunjukkan sebuah grafik waktu yang mempunyai *range* nilai dari 0 – 180 setiap nilai linguistik dari variabel waktu seperti banyak, sedang dan sedikit mempunyai fuzzifikasi yang berbeda-beda. Perhitungan nilai fuzzifikasi didapatkan dari beberapa fungsi, fungsi yang digunakan pada variabel waktu ada 3 yaitu fungsi linier turun, segitiga dan Linear naik. Berikut perhitungan manual dari ketiga fungsi tersebut :

$$\mu[\text{Sedikit}] = \begin{cases} 0; x \geq 50 \\ 1; x \leq 0 \\ \frac{50 - x}{50 - 0}; 0 < x < 50 \end{cases}$$

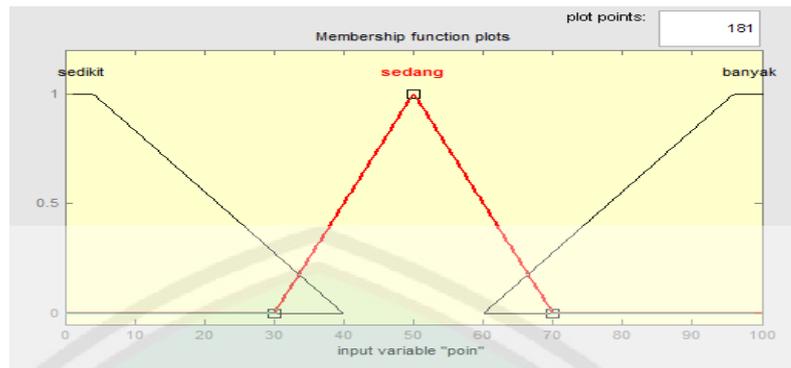
$$\mu[\text{sedang}] = \begin{cases} 0; x \leq 40 \text{ atau } x \geq 90 \\ \frac{x - 40}{60 - 40}; 40 < x < 60 \\ \frac{90 - x}{90 - 60}; 60 \leq x < 90 \end{cases}$$

$$\mu[\text{Banyak}] = \begin{cases} 0; x \leq 80 \\ \frac{x - 130}{130 - 80}; 80 < x < 130 \\ 1; x \geq 130 \end{cases}$$

2. Variabel poin dibagi menjadi tiga keanggotaan yaitu : banyak, sedang dan sedikit. *Range* nilai untuk variabel poin antara 0-100 akan dijelaskan sebagai berikut :

- a. Sedikit = 0-40
- b. Sedang = 30-70
- c. Banyak = 60-100

Pada gambar Menunjukkan sebuah grafik poin yang mempunyai *range* nilai dari 0 – 100, setiap nilai linguistik dari variabel Jumlah Poin seperti Banyak, Sedang dan sedikit mempunyai nilai Fuzzyfikasi yang berbeda-beda. Perhitungan nilai Fuzzyfikasi didapatkan dari beberapa fungsi, fungsi yang digunakan pada variabel Waktu ada 3 yaitu fungsi Linear turun, segitiga dan Linear Naik. Berikut perhitungan manual dari ketiga fungsi tersebut :



Gambar 3. 12 Grafik *Input Variabel Poin*

$$\mu[\text{sedikit}] = \begin{cases} 0; x \geq 40 \\ 1; x \leq 0 \\ \frac{40 - x}{40 - 0}; 0 < x < 40 \end{cases}$$

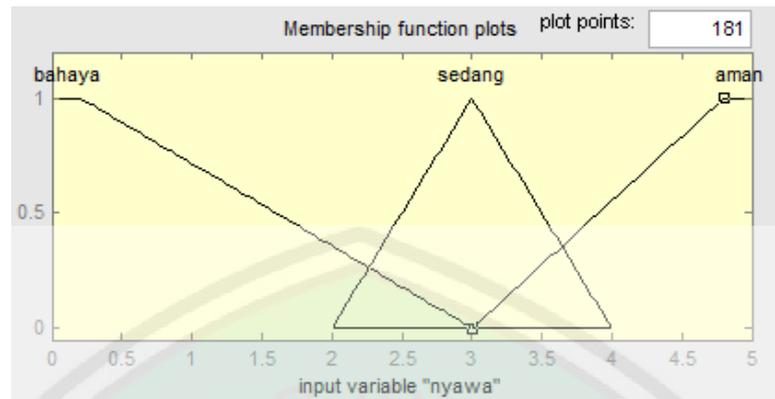
$$\mu[\text{sedang}] = \begin{cases} 0; x \leq 30 \text{ atau } x \geq 70 \\ \frac{x - 30}{50 - 30}; 30 < x < 50 \\ \frac{70 - x}{70 - 50}; 50 < x < 70 \end{cases}$$

$$\mu[\text{banyak}] = \begin{cases} 0; x \leq 60 \\ \frac{x - 60}{100 - 60}; 60 < x < 100 \\ 1; x \geq 100 \end{cases}$$

3. Variabel Nyawa dibagi menjadi tiga keanggotaan yaitu : bahaya, sedang

dan aman dengan *range* antara 0-5 sebagai berikut :

- a. Bahaya = 0-3
- b. Sedang = 2-4
- c. Aman = 3-5



Gambar 3. 13 Grafik *Input* Variabel Nyawa

$$\mu[Bahaya] = \begin{cases} 0; x \geq 3 \\ 1; x \leq 0 \\ \frac{3-x}{3-0}; 0 < x < 3 \end{cases}$$

$$\mu[Sedang] = \begin{cases} 0; x \leq 2 \text{ atau } x \geq 4 \\ \frac{x-2}{2,5-2}; 2 < x < 3 \\ \frac{4-x}{4-3}; 3 < x < 4 \end{cases}$$

$$\mu[Aman] = \begin{cases} 0; x \leq 3 \\ \frac{x-3}{5-3}; 3 < x < 5 \\ 1; x \geq 5 \end{cases}$$

c. Fuzzy rules

1. *If* (Waktu *is* sedikit) *and* (Poin *is* sedikit) *and* (Nyawa *is* bahaya) *then*
(BentukBonus *is* nyawa)
2. *If* (Waktu *is* sedikit) *and* (Poin *is* sedikit) *and* (Nyawa *is* sedang) *then*
(BentukBonus *is* waktu)
3. *If* (Waktu *is* sedikit) *and* (Poin *is* sedikit) *and* (Nyawa *is* aman) *then*
(BentukBonus *is* poin)

4. *If (Waktu is sedikit) and (Poin is sedang) and (Nyawa is bahaya) then (BentukBonus is nyawa)*
5. *If (Waktu is sedikit) and (Poin is sedang) and (Nyawa is sedang) then (BentukBonus is waktu)*
6. *If (Waktu is sedikit) and (Poin is sedang) and (Nyawa is aman) then (BentukBonus is waktu)*
7. *If (Waktu is sedikit) and (Poin is banyak) and (Nyawa is bahaya) then (BentukBonus is nyawa)*
8. *If (Waktu is sedikit) and (Poin is banyak) and (Nyawa is sedang) then (BentukBonus is waktu)*
9. *If (Waktu is sedikit) and (Poin is banyak) and (Nyawa is aman) then (BentukBonus is waktu)*
10. *If (Waktu is sedang) and (Poin is sedikit) and (Nyawa is bahaya) then (BentukBonus is nyawa)*
11. *If (Waktu is sedang) and (Poin is sedikit) and (Nyawa is sedang) then (BentukBonus is poin)*
12. *If (Waktu is sedang) and (Poin is sedikit) and (Nyawa is aman) then (BentukBonus is poin)*
13. *If (Waktu is sedang) and (Poin is sedang) and (Nyawa is bahaya) then (BentukBonus is nyawa)*

14. *If (Waktu is sedang) and (Poin is sedang) and (Nyawa is sedang) then (BentukBonus is waktu)*
15. *If (Waktu is sedang) and (Poin is sedang) and (Nyawa is aman) then (BentukBonus is waktu)*
16. *If (Waktu is sedang) and (Poin is banyak) and (Nyawa is bahaya) then (BentukBonus is nyawa)*
17. *If (Waktu is sedang) and (Poin is banyak) and (Nyawa is sedang) then (BentukBonus is waktu)*
18. *If (Waktu is sedang) and (Poin is banyak) and (Nyawa is aman) then (BentukBonus is poin)*
19. *If (Waktu is banyak) and (Poin is sedikit) and (Nyawa is bahaya) then (BentukBonus is nyawa)*
20. *If (Waktu is banyak) and (Poin is sedikit) and (Nyawa is sedang) then (BentukBonus is poin)*
21. *If (Waktu is banyak) and (Poin is sedikit) and (Nyawa is aman) then (BentukBonus is poin)*
22. *If (Waktu is banyak) and (Poin is sedang) and (Nyawa is bahaya) then (BentukBonus is nyawa)*
23. *If (Waktu is banyak) and (Poin is sedang) and (Nyawa is sedang) then (BentukBonus is nyawa)*

24. *If (Waktu is banyak) and (Poin is sedang) and (Nyawa is aman) then (BentukBonus is poin)*

25. *If (Waktu is banyak) and (Poin is banyak) and (Nyawa is bahaya) then (BentukBonus is nyawa)*

26. *If (Waktu is banyak) and (Poin is banyak) and (Nyawa is sedang) then (BentukBonus is nyawa)*

27. *If (Waktu is banyak) and (Poin is banyak) and (Nyawa is aman) then (BentukBonus is waktu)*

d. Contoh perhitungan

Apabila waktu memiliki nilai 90 , poin memiliki nilai 80 dan nyawa sebesar 1 , maka tahapan untuk mendapatkan hasil keputusan adalah sebagai berikut :

a. Fuzzifikasi

Yaitu memetakan nilai crisp dari waktu, poin dan nyawa kedalam himpunan Fuzzy dan menentukan derajat keanggotaanya. Berikut

Perhitungan Fuzzyfikasi untuk variabel waktu dengan nilai 90 :

$$\mu_{\text{Waktu Sedikit}}[90] = 0; \text{Waktu} \geq 50$$

$$\mu_{\text{waktu Sedang}}[90] = \frac{90 - 60}{30} = 1; 60 < \text{Jarak} \leq 90$$

$$\mu_{\text{waktu Banyak}}[90] = 0; x \leq 130$$

a. Derajat keanggotaan sedikit [90]= 0.

b. Derajat keanggotaan sedang [90]= 1.

c. Derajat keanggotaan banyak [90]= 0.

Perhitungan Fuzzyfikasi variabel poin dengan nilai 80 :

$$\mu \text{ Poin Sedikit}[80] = 0; \text{Poin} \geq 40$$

$$\mu \text{ Poin Sedang}[80] = 0; \text{Poin} \leq 41 \text{ atau } \text{Poin} \geq 60$$

$$\mu \text{ Poin Banyak}[80] = \frac{\text{Poin} - 60}{100 - 60} = 0,5; 60 < \text{Poin} < 100$$

a. Derajat keanggotaan sedikit [80]=0.

b. Derajat keanggotaan sedang [80]=0.

c. Derajat keanggotaan banyak[80]=0,5.

Perhitungan fuzzifikasi variabel nyawa dengan nilai 1:

$$\mu \text{ nyawa bahaya}[1] = \frac{3 - \text{nyawa}}{3 - 0} = 0,6; 0 < \text{nyawa} < 2$$

$$\mu \text{ nyawa sedang}[1] = 0; \text{nyawa} \leq 2 \text{ atau } \text{nyawa} \geq 3$$

$$\mu \text{ nyawa aman}[1] = 0; \text{nyawa} \leq 3$$

a. Derajat keanggotaan bahaya [1]=0,6.

b. Derajat keanggotaan sedang [1]=0.

c. Derajat keanggotaan aman[1]=0.

e. Implikasi

Pada tahap ini akan di bandingkan tiap variabel sesuai dengan *rule* Fuzzy yang sudah di buat , untuk Fuzzy Sugeno digunakan fungsi minimum.

Rule 1:

If (Waktu is sedikit) and (Poin is sedikit) and (Nyawa is bahaya) then

(BentukBonus is nyawa)

$$\rightarrow \text{Min}(0,0,0.6) = 0$$

Rule 2:

*If (Waktu is sedikit) and (Poin is sedikit) and (Nyawa is sedang) then
(BentukBonus is waktu)*

$$\rightarrow \text{Min}(0,0,0) = 0$$

Rule 3:

*If (Waktu is sedikit) and (Poin is sedikit) and (Nyawa is aman) then
(BentukBonus is poin)*

$$\rightarrow \text{Min}(0,0,0) = 0$$

Rule 4:

*If (Waktu is sedikit) and (Poin is sedang) and (Nyawa is bahaya) then
(BentukBonus is nyawa)*

$$\rightarrow \text{Min}(0,0,0.6) = 0$$

Rule 5:

*If (Waktu is sedikit) and (Poin is sedang) and (Nyawa is sedang) then
(BentukBonus is waktu)*

$$\rightarrow \text{Min}(0,0,0) = 0$$

Rule 6:

*If (Waktu is sedikit) and (Poin is sedang) and (Nyawa is aman) then
(BentukBonus is waktu)*

$$\rightarrow \text{Min}(0,0,0) = 0$$

Rule 7:

*If (Waktu is sedikit) and (Poin is banyak) and (Nyawa is bahaya) then
(BentukBonus is nyawa)*

$$\rightarrow \text{Min}(0,0.5,0.6) = 0$$

Rule 8:

*If (Waktu is sedikit) and (Poin is banyak) and (Nyawa is sedang) then
(BentukBonus is waktu)*

$$\rightarrow \text{Min}(0,0.5,0) = 0$$

Rule 9:

*If (Waktu is sedikit) and (Poin is banyak) and (Nyawa is aman) then
(BentukBonus is waktu)*

$$\rightarrow \text{Min}(0,0.5,0) = 0$$

Rule 10:

*If (Waktu is sedang) and (Poin is sedikit) and (Nyawa is bahaya) then
(BentukBonus is nyawa)*

$$\rightarrow \text{Min}(1,0,0.6) = 0$$

Rule 11:

*If (Waktu is sedang) and (Poin is sedikit) and (Nyawa is sedang) then
(BentukBonus is poin)*

$$\rightarrow \text{Min}(1,0,0) = 0$$

Rule 12:

*If (Waktu is sedang) and (Poin is sedikit) and (Nyawa is aman) then
(BentukBonus is poin)*

$$\rightarrow \text{Min}(1,0,0) = 0$$

Rule 13:

*If (Waktu is sedang) and (Poin is sedang) and (Nyawa is bahaya) then
(BentukBonus is nyawa)*

$$\rightarrow \text{Min}(1,0,0.6) = 0$$

Rule 14:

*If (Waktu is sedang) and (Poin is sedang) and (Nyawa is sedang) then
(BentukBonus is waktu)*

$$\rightarrow \text{Min}(1,0,0) = 0$$

Rule 15:

*If (Waktu is sedang) and (Poin is sedang) and (Nyawa is aman) then
(BentukBonus is waktu)*

$$\rightarrow \text{Min}(1,0,0) = 0$$

Rule 16:

*If (Waktu is sedang) and (Poin is banyak) and (Nyawa is bahaya) then
(BentukBonus is nyawa)*

$$\rightarrow \text{Min}(1,0.5,0.6) = 0.3$$

Rule 17:

*If (Waktu is sedang) and (Poin is banyak) and (Nyawa is sedang) then
(BentukBonus is waktu)*

$$\rightarrow \text{Min}(1,0.5,0) = 0$$

Rule 18:

*If (Waktu is sedang) and (Poin is banyak) and (Nyawa is aman) then
(BentukBonus is poin)*

$$\rightarrow \text{Min}(1,0.5,0) = 0$$

Rule 19:

*If (Waktu is banyak) and (Poin is sedikit) and (Nyawa is bahaya) then
(BentukBonus is nyawa)*

$$\rightarrow \text{Min}(0,0,0.6) = 0$$

Rule 20:

*If (Waktu is banyak) and (Poin is sedikit) and (Nyawa is sedang) then
(BentukBonus is poin)*

$$\rightarrow \text{Min}(0,0,0) = 0$$

Rule 21:

*If (Waktu is banyak) and (Poin is sedikit) and (Nyawa is aman) then
(BentukBonus is poin)*

$$\rightarrow \text{Min}(0,0,0) = 0$$

Rule 22:

*If (Waktu is banyak) and (Poin is sedang) and (Nyawa is bahaya) then
(BentukBonus is nyawa)*

$$\rightarrow \text{Min}(0,0,0.6) = 0$$

Rule 23:

*If (Waktu is banyak) and (Poin is sedang) and (Nyawa is sedang) then
(BentukBonus is nyawa)*

$$\rightarrow \text{Min}(0,0,0) = 0$$

Rule 24:

*If (Waktu is banyak) and (Poin is sedang) and (Nyawa is aman) then
(BentukBonus is poin)*

$$\rightarrow \text{Min}(0,0,0) = 0$$

Rule 25:

*If (Waktu is banyak) and (Poin is banyak) and (Nyawa is bahaya) then
(BentukBonus is nyawa)*

$$\rightarrow \text{Min}(0,0.5,0.6) = 0$$

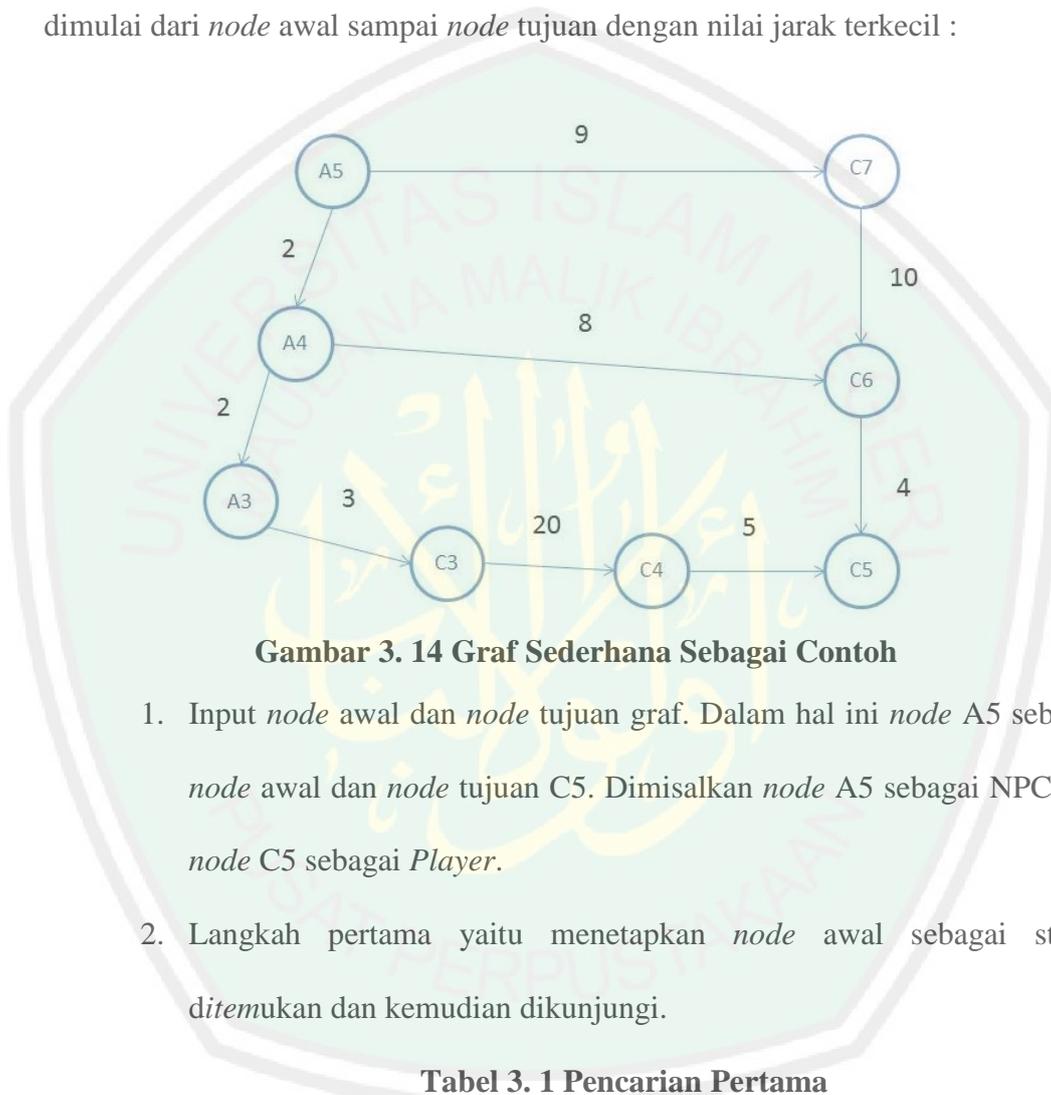
Rule 26:

*If (Waktu is banyak) and (Poin is banyak) and (Nyawa is sedang) then
(BentukBonus is nyawa)*

$$\rightarrow \text{Min}(0,0.5,0) = 0$$

3.4 Algoritma Dijkstra

Berdasarkan penjelasan tentang teori Dijkstra pada bab sebelumnya, maka berikut ini penjelasan langkah per langkah pencarian jalur terpendek secara rinci dimulai dari *node* awal sampai *node* tujuan dengan nilai jarak terkecil :



Gambar 3. 14 Graf Sederhana Sebagai Contoh

1. Input *node* awal dan *node* tujuan graf. Dalam hal ini *node* A5 sebagai *node* awal dan *node* tujuan C5. Dimisalkan *node* A5 sebagai NPC dan *node* C5 sebagai *Player*.
2. Langkah pertama yaitu menetapkan *node* awal sebagai status *ditemukan* dan kemudian dikunjungi.

Tabel 3. 1 Pencarian Pertama

<i>Node</i>	Status	Bobot	Rute
A5(asal)	Dikunjungi	0	A5
A3	Belum <i>ditemukan</i>	~	-
A4	Belum <i>ditemukan</i>	~	-
C3	Belum <i>ditemukan</i>	~	-
C4	Belum <i>ditemukan</i>	~	-
C6	Belum <i>ditemukan</i>	~	-

C7	Belum ditemukan	~	-
C5(tujuan)	Belum ditemukan	~	-

3. Langkah kedua yaitu dilakukan pencarian terhadap setiap *node* yang dapat dicapai secara langsung dari *node* yang sedang dikunjungi.

Karena yang berdekatan dengan A5 adalah *node* A4 = 2 dan C7 = 9

4. Langkah ketiga yaitu :
- Apabila *node* yang didapatkan pada langkah kedua belum pernah ditemukan, maka rubah statusnya menjadi *ditemukan*.
 - Apabila *node* yang didapatkan sudah pernah *ditemukan* maka lakukan update pada bobotnya, ambil bobot yang lebih kecil.

Tabel 3. 2 Pencarian Kedua

<i>Node</i>	Status	Bobot	Rute
A5(asal)	Dikunjungi	0	A5
A3	Belum ditemukan	~	-
A4	Ditemukan	2	A5 => A4
C3	Belum ditemukan	~	-
C4	Belum ditemukan	~	-
C6	Belum ditemukan	~	-
C7	Ditemukan	9	A5 => C7
C5(tujuan)	Belum ditemukan	~	-

5. Langkah keempat yaitu dilakukan pencarian terhadap *node* yang memiliki bobot paling kecil dari semua *node* yang berada pada status *ditemukan* kemudian mengunjunginya. Maka dihasilkan *node* A4 sebagai *node* bobot paling kecil. Kemudian sistem mengunjungi *node* A4. Statusnya berubah menjadi A4 dikunjungi.

6. *Node* yang berdekatan dengan *node* A4 adalah *node* A3= 4 dan C6=10.

Tabel 3. 3 Pencarian Ketiga

<i>Node</i>	Status	Bobot	Rute
A5(asal)	Dikunjungi	0	A5
A3	Ditemukan	4	A5 => A4 => A3
A4	Dikunjungi	2	A5 => A4
C3	Belum ditemukan	~	-
C4	Belum ditemukan	~	-
C6	Ditemukan	10	A5 => A4 => C6
C7	Ditemukan	9	A5 => C7
C5(tujuan)	Belum ditemukan	~	-

7. Antara *node* yang ditemukan dihasilkan *node* A3 dengan bobot terkecil sistem akan mengunjungi A3.
8. *Node* yang berdekatan dengan A3 yaitu C3 = 7 .

Tabel 3. 4 Pencarian Keempat

<i>Node</i>	Status	Bobot	Rute
A5(asal)	Dikunjungi	0	A5
A3	Ditemukan	4	A5 => A4 => A3
A4	Dikunjungi	2	A5 => A4
C3	Ditemukan	7	A5 => A4 => A3 => C3
C4	Belum ditemukan	~	-
C6	Ditemukan	10	A5 => A4 => C6
C7	Ditemukan	9	A5 => C7
C5(tujuan)	Belum ditemukan	~	-

9. *Node* C3 merupakan *node* dengan bobot terkecil diantara *node-node* yang ditemukan , maka sistem akan mengunjungi C3.
10. *Node* yang berdekatan dengan C3 adalah C4 = 27 , status C4 berubah menjadi ditemukan.

Tabel 3. 5 Pencarian Kelima

<i>Node</i>	Status	Bobot	Rute
A5(asal)	Dikunjungi	0	A5
A3	Ditemukan	4	A5 => A4 => A3
A4	Dikunjungi	2	A5 => A4
C3	Ditemukan	7	A5 => A4 => A3 => C3
C4	Ditemukan	27	A5 => A4 => A3 => C3 => C4
C6	Ditemukan	10	A5 => A4 => C6
C7	Ditemukan	9	A5 => C7
C5(tujuan)	Belum ditemukan	~	-

11. *Node* C7 ditemukan paling kecil bobotnya, maka sistem mengunjungi C7.

12. *Node* yang berdekatan dengan C7 adalah C6 maka akan ditemukan akumulasi jalur sebesar C6 = 19. Dibandingkan dengan akumulasi nilai bobot C6 dari jalur A4 yang nilainya C6 = 10 maka sistem akan mengambil C6 dengan jalur dari A4.

Tabel 3. 6 Pencarian Keenam

<i>Node</i>	Status	Bobot	Rute
A5(asal)	Dikunjungi	0	A5
A3	Dikunjungi	4	A5 => A4 => A3
A4	Dikunjungi	2	A5 => A4
C3	Dikunjungi	7	A5 => A4 => A3 => C3
C4	Ditemukan	27	A5 => A4 => A3 => C3 => C4
C6	Ditemukan	10	A5 => A4 => C6
C7	Dikunjungi	9	A5 => C7

C5(tujuan)	Belum ditemukan	~	-
------------	-----------------	---	---

13. Dari *node* yang ditemukan yaitu *node* C4= 27 dan C6= 10 , maka sistem akan mengunjungi C6.
14. *Node* yang berdekatan dengan C6 merupakan *node* tujuan yaitu C5 dengan bobot 14 sedangkan C4= 27
15. Maka dihasilkan C5 dengan bobot akumulasi terkecil karena C5 merupakan *node* tujuan maka pencarian dihentikan. Dengan demikian rute dengan bobot terkecil yang dapat dicapai C5 yaitu A5 => A4 => C6 => C5 dengan total 14.

Tabel 3. 7 Pencarian Ketujuh

<i>Node</i>	Status	Bobot	Rute
A5(asal)	Dikunjungi	0	A5
A3	Dikunjungi	4	A5 => A4 => A3
A4	Dikunjungi	2	A5 => A4
C3	Dikunjungi	7	A5 => A4 => A3 => C3
C4	Ditemukan	27	A5 => A4 => A3 => C3 => C4
C6	Ditemukan	10	A5 => A4 => C6
C7	Dikunjungi	9	A5 => C7
C5(tujuan)	Dikunjungi	14	A5 => A4 => C6 => C5

Dari pencarian di atas dapat diambil kesimpulan yaitu NPC dapat menyerang *player* dengan jalur terpendek A5 => A4 => C6 => C5.

BAB 4

IMPLEMENTASI DAN PEMBAHASAN

4.1 Implementasi

Bab ini menjelaskan tentang hasil implemtasi dari perancangan desain *game*, perancangan Fuzzy dan Djikstra serta pengujian dalam *game* yang telah dibuat. Tujuan dilakukan pengujian ini adalah untuk mengetahui apakah metode yang dipilih oleh penulis bisa mengatur perilaku NPC musuh sesuai variabel yang dimiliki.

Sebelum diimplementasikan, terlebih dahulu dipaparkan spesifikasi dan software yang akan digunakan :

4.1.1 Kebutuhan Perangkat Keras

Perangkat keras yang digunakan untuk pembuatan aplikasi dan pengujian metode tersebut adalah sebagai berikut :

- a. Laptop Lenovo Z470 : Processor intel(R) Core(TM) i3 Geforce NVidia
RAM 2.00GB

4.1.2 Kebutuhan Perangkat Lunak

Software yang digunakan dalam pembuatan *game* dan pengujian metode adalah:

- Sistem Operasi : Windows 7 64 Bit
- *Game Engine* : Unity3D 5.3.5f1
- Konsep desain 2D : Corel Draw X6 dan Adobe Photoshop CS6
- *Script Writer* : Monodevelop
- Pengujian metode : Matlab 2013

4.2 Pengujian Algoritma Fuzzy

Pada implementasi Fuzzy Sugeno pada Unity ini, terdapat 3 variabel *input* yaitu Waktu, Poin dan Nyawa dengan Bentuk bonus sebagai *outputnya*. Dalam Fuzzy terdapat 3 proses yakni fuzzifikasi, Fuzzy inference, dan defuzzikasi. Penggunaan Fuzzy Sugeno sudah sesuai pada *game*, digunakan Fuzzy Sugeno karena Fuzzy ini yang nilai *outputnya* berupa bilangan sedangkan untuk Fuzzy yang lain hanya berupa konstanta atau persamaan linier.

a. Fuzzifikasi Waktu

Pada proses ini memetakan nilai waktu kedalam himpunan Fuzzy dan menentukan derajat keanggotaanya. Dalam penelitian ini menggunakan fungsi keanggotaan linier naik, linier turun, dan linier segitiga. Terdapat 3 himpunan Fuzzy yaitu sedikit, sedang, dan banyak. Masing-masing himpunan Fuzzy memiliki nilai keanggotaan yang berbeda tergantung nilai *input* yang dimasukkan. Variabel waktu memiliki *range* nilai untuk variabel waktu antara 0- 180, waktu sedikit memiliki range = 0-50, himpunan waktu sedang = 40-90, dan himpunan banyak = 80-130.

```
Float Waktu;
static float WaktuSedikit;
static float WaktuSedang;
static float WaktuBanyak;

//Waktu Sedikit
if(Waktu>=50){
    WaktuSedikit=0;
} else if(Waktu<=0){
    WaktuSedikit=1;
```

```

    } else if((Waktu>0)&&(Waktu<50)){
        WaktuSedikit=(50-Waktu)/(50-0);
    }
//Waktu Sedang
if((Waktu<=40)|| (Waktu>=90)){
    WaktuSedang=0;
} else if((Waktu>40)&&(Waktu<60)){
    WaktuSedang=(Waktu-40)/(60-40);
} else if((Waktu>=60)&&(Waktu<90)){
    WaktuSedang=(90-Waktu)/(90-60);
}
//Waktu Banyak
if(Waktu<=80){
    WaktuBanyak=0;
} else if((Waktu>80)&&(Waktu<130)){
    WaktuBanyak=(Waktu-130)/(130-80);
} else if(Waktu>=130){
    WaktuBanyak=1;
}

```

b. Fuzzifikasi Poin

Pada proses ini memetakan nilai poin kedalam himpunan Fuzzy dan menentukan derajat keanggotaanya. Dalam penelitian ini menggunakan fungsi keanggotaan linier naik, linier turun, dan linier segitiga. Terdapat 3 himpunan Fuzzy yaitu sedikit, sedang, dan banyak. Masing-masing himpunan Fuzzy memiliki nilai keanggotaan yang berbeda tergantung nilai *input* yang dimasukkan. Variabel waktu memiliki *range* nilai untuk

variabel poin antara 0- 100, poin sedikit memiliki *range* = 0-40, himpunan poin sedang = 30-70, dan himpunan banyak = 60-100.

```

Float Poin;

static float PoinSedikit;

static float PoinSedang;

static float PoinBanyak;

    //Poin Sedikit
    if(Poin >= 40){
        PoinSedikit=0;
    } else if(Poin<=0){
        PoinSedikit=1;
    } else if((Poin>0)&&(Poin<40)){
        PoinSedikit = (40 - Poin)/(40-0);
    }

    //Poin Banyak
    if(Poin<=60){
        PoinBanyak=0;
    }else if((Poin>60)&&(Poin<100)){
        PoinBanyak=(Poin-60)/(100-60);
    }else if(Poin>=100){
        PoinBanyak=1;
    }

```

c. Fuzzifikasi Nyawa

Pada proses ini memetakan nilai nyawa kedalam himpunan Fuzzy dan menentukan derajat keanggotaanya. Dalam penelitian ini menggunakan fungsi keanggotaan linier naik, linier turun, dan linier segitiga. Terdapat

3 himpunan Fuzzy yaitu bahaya, sedang, dan aman. Masing-masing himpunan Fuzzy memiliki nilai keanggotaan yang berbeda tergantung nilai *input* yang dimasukkan. Variabel nyawa memiliki *range* nilai untuk variabel nyawa antara 0- 5, nyawa bahaya memiliki *range* = 0-3, himpunan nyawa sedang = 2-4, dan himpunan aman = 3-5.

```

Float Nyawa;
static float NyawaBahaya;
static float NyawaSedang;
static float NyawaAman;

//Nyawa Bahaya
    if (Nyawa >= 3) {
        NyawaBahaya = 0;
    } else if ((Nyawa > 0) && (Nyawa < 3)) {
        NyawaBahaya = (3 - Nyawa) / (3 - 0);
    } else if (Nyawa <= 0) {
        NyawaBahaya = 1;
    }

// Nyawa Sedang
    if((Nyawa<=2)|| (Nyawa>=4)){
        NyawaSedang=0;
    } else if((Nyawa>2)&&(Nyawa<=3)){
        NyawaSedang = (Nyawa-2)/(3-2);
    } else if((Nyawa>3)&&(Nyawa<4)){
        NyawaSedang=(4-Nyawa)/(4-3);
    }

//Nyawa Aman
    if(Nyawa<=3){

```

```

        NyawaAman=0;

        }else if((Nyawa>3)&&(Nyawa<5)){

            NyawaAman=(Nyawa-3)/(5-3);

        }else if(Nyawa>=5){

            NyawaAman=1;

        }

```

d. Inference

Proses untuk mengubah *input* Fuzzy menjadi *output* Fuzzy dengan cara mengikuti aturan-aturan (*IF-THEN Rules*) yang telah ditetapkan pada basis pengetahuan Fuzzy.

```

float NyawaBahaya;
float NyawaSedang;
float NyawaAman;

float WaktuSedikit;

float WaktuSedang;

float WaktuBanyak;

float PoinSedikit;

float PoinSedang;

float PoinBanyak;

float z;

z == 1 (Poin bertambah)

z == 2 (Nyawa bertambah)

z == 3 (Waktu bertambah)

float aa = Mathf.Min(PoinSedikit,NyawaBahaya,WaktuSedikit);

float zzz = Mathf.Min(PoinBanyak,NyawaAman,WaktuBanyak);

float z27 = 3;

```

e. Defuzzifikasi

Metode defuzzifikasi menggunakan rumus rata-rata, yaitu :

$$z = \frac{\sum a_i z_i}{\sum a_i}$$

$$z = ((aa * z1) + (bb * z2) + (cc * z3) + (dd * z4) + (ee * z5) + (ff * z6) + (gg * z7) + (hh * z8) + (ii * z9) + (jj * z10) + (kk * z11) + (ll * z12) + (mm * z13) + (nn * z14) + (oo * z15) + (pp * z16) + (qq * z17) + (rr * z18) + (ss * z19) + (tt * z20) + (uu * z21) + (vv * z22) + (ww * z23) + (xx * z24) + (yy * z25) + (zz * z26) + (zzz * z27)) / (aa + bb + cc + dd + ee + ff + gg + hh + ii + jj + kk + ll + mm + nn + oo + pp + qq + rr + ss + tt + uu + vv + xx + yy + zz + zzz);$$

4.2.1 Pengujian Algoritma Fuzzy Sugeno

Pada bagian ini akan menjelaskan tentang hasil pengujian sistem dengan menguji hasil Fuzzy Sugeno terhadap tingkat penyerangan NPC pada *game Muslim Kid*. Berikut akan dijelaskan tentang hasil pengujian dari Algoritma Fuzzy Sugeno dalam bentuk tabel :

Tabel 4. 1 Pengujian Algoritma Fuzzy pada Matlab

No	INPUT			OUTPUT		KESES UAIAN
	WAKTU	POIN	NYAWA	HASIL	KEPUTUSAN	
1	10	10	1	3	Waktu	Sesuai
2	30	30	2	3	Waktu	Sesuai
3	50	10	2	2	Nyawa	Sesuai

4	70	10	3	3	Waktu	Sesuai
5	70	10	4	1	Poin	Sesuai
6	50	50	3	1	Poin	Sesuai
7	40	80	3	3	Waktu	Sesuai
8	80	80	1	1	Poin	Sesuai
9	80	70	4	3	Waktu	Sesuai
10	10	50	4	3	Waktu	Sesuai
11	40	100	2	2	Nyawa	Sesuai
12	20	60	3	3	Waktu	Sesuai
13	120	30	2	2	Nyawa	Sesuai
14	100	15	4	1	Poin	Sesuai
15	130	0	5	1	Poin	Sesuai
16	95	50	3	2	Nyawa	Sesuai
17	15	50	4	3	Waktu	Sesuai
18	121	10	4	1	Poin	Sesuai
19	121	50	4	1	Poin	Sesuai
20	121	35	1	2	Nyawa	Sesuai
21	65	50	2	2	Nyawa	Sesuai
22	65	10	3	1	Poin	Sesuai
23	65	10	1	2	Nyawa	Sesuai
24	22	15	3	3	Waktu	Sesuai
25	22	30	3	3	Waktu	Sesuai
26	22	60	5	3	Waktu	Sesuai

27	22	60	1	2	Nyawa	Sesuai
28	22	60	2	2	Nyawa	Sesuai
29	95	10	5	1	Poin	Sesuai
30	95	15	3	1	Poin	Sesuai
31	95	25	1	2	Nyawa	Sesuai
32	95	45	4	1	Poin	Sesuai
33	95	10	2	2	Nyawa	Sesuai
34	110	10	4	1	Poin	Sesuai
35	110	15	4	1	Poin	Sesuai
36	110	5	4	1	Poin	Sesuai
37	10	55	3	3	Waktu	Sesuai
38	10	40	3	3	Waktu	Sesuai
39	10	40	1	2	Nyawa	Sesuai
40	10	65	2	2	Nyawa	Sesuai
41	10	60	4	3	Waktu	Sesuai
42	125	60	4	1	Poin	Sesuai
43	125	60	2	2	Nyawa	Sesuai
44	125	50	5	1	Poin	Sesuai
45	25	80	5	3	Waktu	Sesuai
46	25	80	3	3	Waktu	Sesuai
47	25	80	2	2	Nyawa	Sesuai
48	15	50	4	3	Waktu	Sesuai
49	15	40	3	3	Waktu	Sesuai

50	15	40	1	2	Nyawa	Sesuai
----	----	----	---	---	-------	--------

Tabel 4. 2 Pengujian Algoritma Fuzzy pada Game

No	INPUT			OUTPUT		KESES
	WAKTU	POIN	NYAWA	HASIL	KEPUTUSAN	UAIAN
1	10	10	1	3	Waktu	Sesuai
2	30	30	2	3	Waktu	Sesuai
3	50	10	2	2	Nyawa	Sesuai
4	70	10	3	3	Waktu	Sesuai
5	70	10	4	1	Poin	Sesuai
6	50	50	3	1	Poin	Sesuai
7	40	80	3	3	Waktu	Sesuai
8	80	80	1	1	Poin	Sesuai
9	80	70	4	3	Waktu	Sesuai
10	10	50	4	3	Waktu	Sesuai
11	40	100	2	2	Nyawa	Sesuai
12	20	60	3	3	Waktu	Sesuai
13	120	30	2	2	Nyawa	Sesuai
14	100	15	4	1	Poin	Sesuai
15	130	0	5	1	Poin	Sesuai
16	95	50	3	2	Nyawa	Sesuai
17	15	50	4	3	Waktu	Sesuai
18	121	10	4	1	Poin	Sesuai
19	121	50	4	1	Poin	Sesuai

20	121	35	1	2	Nyawa	Sesuai
21	65	50	2	2	Nyawa	Sesuai
22	65	10	3	1	Poin	Sesuai
23	65	10	1	2	Nyawa	Sesuai
24	22	15	3	3	Waktu	Sesuai
25	22	30	3	3	Waktu	Sesuai
26	22	60	5	3	Waktu	Sesuai
27	22	60	1	2	Nyawa	Sesuai
28	22	60	2	2	Nyawa	Sesuai
29	95	10	5	1	Poin	Sesuai
30	95	15	3	1	Poin	Sesuai
31	95	25	1	2	Nyawa	Sesuai
32	95	45	4	1	Poin	Sesuai
33	95	10	2	2	Nyawa	Sesuai
34	110	10	4	1	Poin	Sesuai
35	110	15	4	1	Poin	Sesuai
36	110	5	4	1	Poin	Sesuai
37	10	55	3	3	Waktu	Sesuai
38	10	40	3	3	Waktu	Sesuai
39	10	40	1	2	Nyawa	Sesuai
40	10	65	2	2	Nyawa	Sesuai
41	10	60	4	3	Waktu	Sesuai
42	125	60	4	1	Poin	Sesuai

43	125	60	2	2	Nyawa	Sesuai
44	125	50	5	1	Poin	Sesuai
45	25	80	5	3	Waktu	Sesuai
46	25	80	3	3	Waktu	Sesuai
47	25	80	2	2	Nyawa	Sesuai
48	15	50	4	3	Waktu	Sesuai
49	15	40	3	3	Waktu	Sesuai
50	15	40	1	2	Nyawa	Sesuai

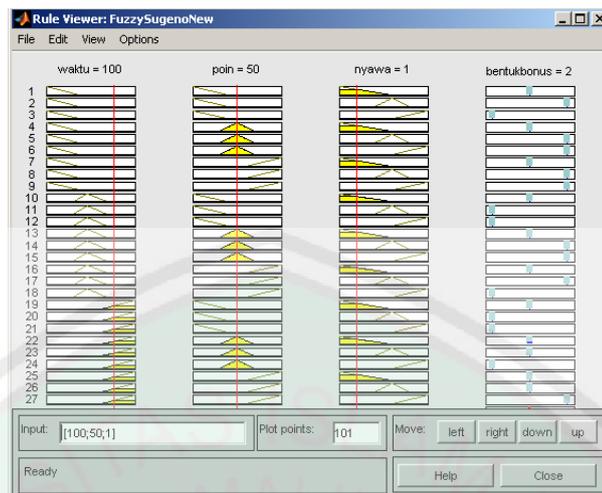
Dari tabel tersebut dapat di lihat bahwa semua *output* sudah sesuai *rule* yang telah ditentukan. Bentuk bonus yang dihasilkan dari *output* tersebut adalah :

$$\text{Poin} = (16/50) * 100 = 32\%$$

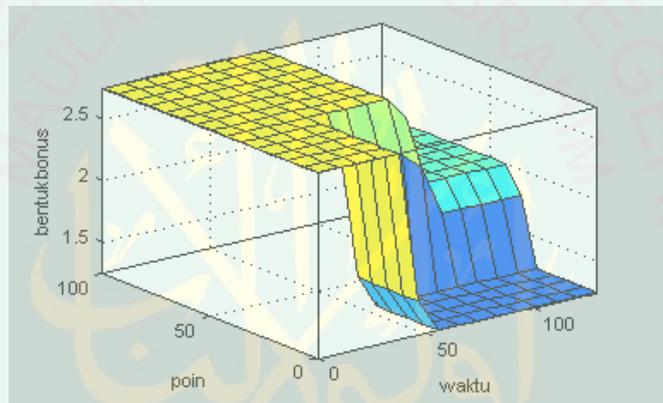
$$\text{Nyawa} = (16/50) * 100 = 32\%$$

$$\text{Waktu} = (18/50) * 100 = 36\%$$

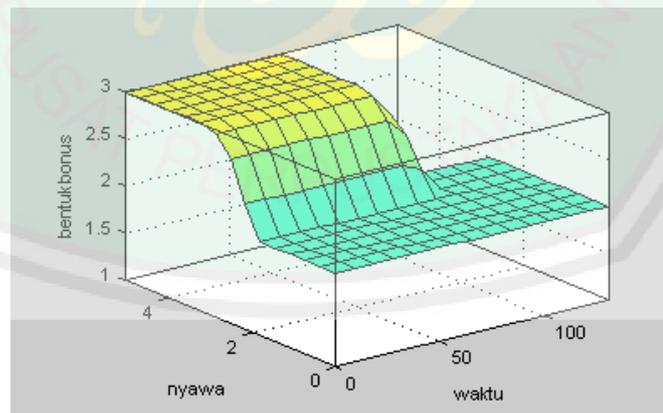
Pengujian Algoritma Fuzzy Sugeno pada *output* bentuk bonus dengan contoh input waktu = 100, poin = 50, nyawa = 1 disimulasikan dalam aplikasi Matlab. Berikut hasil simulasi :



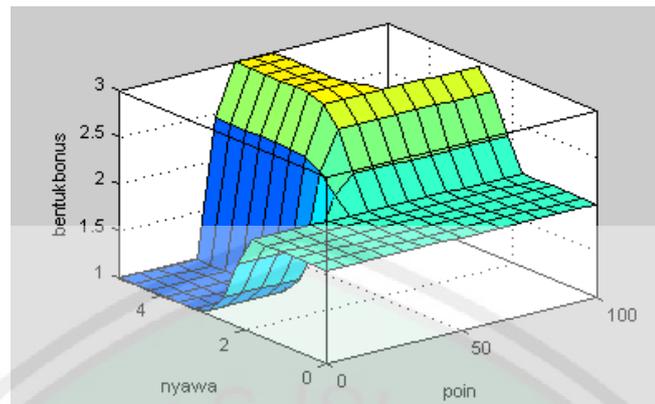
Gambar 4. 1 Tampilan Simulasi *Output* pada Matlab



Gambar 4. 2 Sumbu Kartesian Untuk Waktu dan Poin



Gambar 4. 3 Sumbu Kartesian Untuk Nyawa dan Waktu

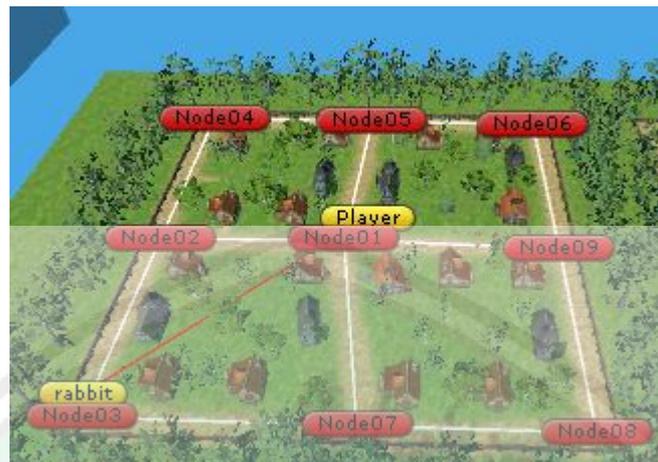


Gambar 4. 4 Sumbu Kartesian Untuk Poin dan Nyawa

Pada gambar 4.2 , 4.3 dan 4.4 merupakan sumbu kartesian pada matlab untuk *inputan* variabel fuzzy Poin, Waktu, dan Nyawa.

4.3 Pengujian Algoritma Dijkstra

Pengujian ini dilakukan untuk melihat bagaimana proses menuju target yang dilakukan oleh Algoritma Dijkstra. Penggunaan Algoritma Dijkstra sudah sesuai pada *game*, digunakan Dijkstra ini karena Algoritma ini untuk menentukan jarak yang terpendek dan sesuai dengan arena *game* yang tidak luas, misal untuk arena *game* yang luas Algoritma yang sesuai adalah A*. Pengujian pertama dilakukan dengan posisi NPC pada *node* 3 dan *player* pada *node* 1 (lihat Gambar 4.5). Proses yang pertama NPC bergerak menuju *node* 2 karena *node* yang berhubungan dengan *node* 3 adalah *node* 7 dan *node* 2, sesuai dengan Algoritma Dijkstra NPC memilih *node* 2 (47,3) (lihat Gambar 4.6) karena jaraknya lebih pendek daripada *node* 7 (52,7) (lihat Tabel 4.3). Proses yang kedua NPC langsung menuju ke *node* 1 (lihat Gambar 4.7) karena *node* yang berhubungan dengan *node* 2 adalah *node* 1 dan *node* 4, *player* terletak pada *node* 1 maka NPC langsung menuju ke *node* 1. Berikut gambar proses NPC menuju target :



Gambar 4. 5 Contoh Posisi Awal NPC



Gambar 4. 6 NPC Menuju Target



Gambar 4. 7 NPC Bertemu Dengan Target

Dengan Algoritma Dijkstra diharapkan NPC dapat bergerak menuju *player* dengan memilih jarak yang terdekat terhadap *player*. Pengujian ini dilakukan untuk mengecek apakah algoritma Dijkstra telah berfungsi sehingga dapat membuat NPC bergerak menuju *player* seperti yang diharapkan. Dan terbukti

Algoritma Dijkstra dapat diterapkan untuk menentukan jarak terdekat terhadap *player*. Berikut merupakan tabel jarak tiap *node* – *node* :

Tabel 4. 3 Tabel Jarak Tiap *Node*

NO	NODE	NODE	JARAK
1	1	2	45,2
2	1	5	50,3
3	1	7	49,1
4	1	9	52,4
5	2	1	45,2
6	2	3	47,3
7	2	4	51,6
8	3	2	47,3
9	3	7	52,7
10	4	2	51,6
11	4	5	48,5
12	5	1	50,3
13	5	4	48,5
14	5	6	48,8
15	6	5	48,8
16	6	9	51,8
17	7	1	49,1
18	7	3	52,7
19	7	8	44,9

20	8	7	44,9
21	8	9	47,1
22	9	1	52,4
23	9	6	51,8
24	9	8	47,1

4.4 Implementasi Game

Bagian ini merupakan hasil akhir setelah dilakukan uji coba dan implementasi Fuzzy dan Dijkstra. Bagian ini juga terdiri dari gambar *splash screen*, gambar tampilan menu *game*, gambar *stage game*, gambar karakter *player*, gambar karakter musuh, gambar tampilan ketika muncul misi/soal, gambar tampilan saat *game* berlangsung, gambar tampilan, gambar tampilan *game over*.

a. *Splash Screen*



Gambar 4. 8 *Splash Screen*

Splash screen ini merupakan *opening* dalam *game*, menampilkan tampilan awal pada *game*.

b. Menu *game*

Menu *game* merupakan tampilan yang muncul setelah *splash screen*.



Gambar 4. 9 Menu Game

c. Stage game



Gambar 4. 10 Stage Game

Stage game adalah halaman permainan pada *game*, yang berisi *player*, NPC dan *terrain* permainan yang akan dimainkan dalam *game* yang dibuat. Dalam *stage* ini pemain di tuntut untuk mengambil *item-item* yang tersebar dan melewati rintangan dan NPC yang siap menghalangi *player* dalam mengambil *item*.

d. Konten game



Gambar 4. 11 Konten Game

Konten *game* berisi pernyataan dan pertanyaan yang akan muncul ketika *player* mengambil *item*. Di dalamnya terdapat informasi dan pertanyaan seputar Nabi dan Rasul.

e. *Next Stage*



Gambar 4. 12 *Next stage*

Berisi *item box* yang berfungsi untuk pindah ke *stage* selanjutnya.

f. *Mission complete*



Gambar 4. 13 *Mission Complete*

Ketika *player* telah menyelesaikan misi pada *game* maka akan mendapatkan piala.

g. *Game Over*



Gambar 4. 14 *Game Over*

Apabila *player* dalam kondisi nyawa 0 maka akan *game over*.

h. *Player*



Gambar 4. 15 *Player*

Player Ahmad adalah karakter utama dalam *game* ini, bertugas menyelesaikan *game* dengan melewati rintangan dan musuh, dan mengambil *item* tas dan buku.

i. NPC



Gambar 4. 16 NPC

j. *Item* Buku

Gambar 4. 17 Buku

Merupakan *item* yang harus diambil oleh *player* yang berisi pernyataan dan pertanyaan.

k. *Item* Tas

Gambar 4. 18 Tas

Item tas merupakan *item* yang berisi bonus yang jika diambil oleh *player* maka akan memperoleh bonus.

4.5 Integrasi dalam Islam

Salah satu kisah yang terdapat dalam Al-Qur'an adalah kisah Nabi Adam AS. Terdapat juga kisah-kisah Nabi, Rasul dan kisah yang menceritakan tokoh dan kehidupan masa lampau. Tujuan utama penuturan kisah masa lalu dalam Al-Qur'an bukanlah untuk memenuhi hasrat keingin-tahuan manusia akan cerita sejarah, tetapi lebih merupakan upaya untuk menjadikan sejarah sebagai pelajaran. Karena itu, Al-Qur'an hanya mengambil bagian-bagian terpenting dari suatu sejarah yang dapat dijadikan pelajaran bagi kehidupan umat manusia.

Ayat yang memerintahkan untuk mempelajari sejarah dan peradaban Islam terdapat dalam QS. Yusuf : 111 berikut :

لَقَدْ كَانَ فِي قَصَصِهِمْ عِبْرَةٌ لِأُولِي الْأَلْبَابِ مَا كَانَ حَدِيثًا يُفْتَرَى وَلَكِن تَصْدِيقَ الَّذِي بَيْنَ يَدَيْهِ

وَتَفْصِيلَ كُلِّ شَيْءٍ وَهُدًى وَرَحْمَةً لِّقَوْمٍ يُؤْمِنُونَ - ١١١

Artinya : *“Sesungguhnya pada kisah-kisah mereka itu terdapat pengajaran bagi orang-orang yang mempunyai akal. Al Quran itu bukanlah cerita yang dibuat-buat, akan tetapi membenarkan (kitab-kitab) yang sebelumnya dan menjelaskan segala sesuatu, dan sebagai petunjuk dan rahmat bagi kaum yang beriman.”*
(QS.Yusuf:111)

Menurut Tafsir Ibnu Katsir Allah SWT berfirman bahwa sesungguhnya, dalam kisah para Rasul dan kaum mereka serta bagaimana Allah SWT telah menyelamatkan orang-orang yang beriman dan menghancurkan orang-orang yang kafir : *“Terdapat pengajaran bagi orang-orang yang mempunyai akal. Al Quran itu bukanlah cerita yang dibuat-buat.”* Maksudnya, Al-Qur’an tidak seharusnya didustakan dan dibuat-buat dari selain Allah.

Ayat diatas menunjukkan bahwa Allah SWT memerintahkan untuk mempelajari kisah Nabi dan Rasul dan umat sebelumnya yang terdapat dalam Al-Qur’an karena didalamnya terdapat pelajaran yang sangat berharga yang dapat di terapkan dalam kehiduapn sehari-hari. Manfaat mempelajari Sejarah Kebudayaan Islam diantaranya adalah:

1. Merasa bangga dan mencintai kebudayaan Islam yang merupakan buah karya kaum Muslimin masa lalu.

2. Berpartisipasi memelihara peninggalan-peninggalan masa lalu dengan cara mempelajari dan mengambil manfaat dari peninggalan-peninggalan tersebut.
3. Meneladani perilaku yang baik dari tokoh-tokoh terdahulu.
4. Mengambil pelajaran dari berbagai keberhasilan dan kegagalan masa lalu.



BAB 5

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dari implementasi dan pengujian yang dilakukan peneliti, maka dapat ditarik kesimpulan sebagai berikut :

1. Penggunaan Fuzzy Logic Sugeno berhasil diterapkan untuk mengetahui bentuk bonus yang di peroleh *player* dalam *game Muslim Kid*, yaitu terdapat bentuk bonus waktu, poin, dan nyawa dengan keberhasilan 85% di terapkan pada *game*. Sedangkan Algoritma Dijkstra digunakan untuk mengetahui jarak terpendek NPC menuju ke *player*, yang bertujuan menghalangi *player* dalam menyelesaikan misi yaitu mengambil *item-item* yang terdapat pada *game*. Dan menunjukkan keberhasilan 90%.

5.2 Saran

Dalam pembuatan *game* ini peneliti sadar bahwa masih banyak kekurangan yang nantinya sangat perlu untuk dilakukan pengembangan dan sumbangsih terhadap ilmu pengetahuan, diantaranya :

1. Menambah jumlah *stage* permainan sehingga permainan menjadi lebih menantang dan menarik.
2. Menambah ragam NPC dan perilaku yang bervariasi dan NPC disertai dengan senjata agar permainan lebih seri dan menantang. Serta *player* yang dilengkapi juga dengan perilaku dan senjata sehingga bisa menyerang NPC.

3. Permainan ini tidak hanya disajikan dalam *platform desktop* saja, namun juga bisa dikembangkan pada *platform smartphone* agar pemahaman terkait sejarah atau kisah Nabi dan Rasul semakin kian diminati.



DAFTAR PUSTAKA

- Abbas, Ngatmin dkk. 2011. Pendidikan Agama Islam untuk Siswa Sekolah Dasar Kelas V. Jakarta : Pusat Kurikulum dan Perbukuan, Kementrian Pendidikan Nasional.
- Al-Qur'an dan Terjemahannya. 2003. Semarang: Toha Putera.
- Andayani, Sri & Perwitasari, Endah. 2014. *Penentuan Rute Terpendek Pengambilan Sampah di Kota Merauke Menggunakan Algoritma Dijkstra*. Jurusan teknik Informatika Sekolah Tinggi Teknik Musi Palembang.
- Alexander, Michael & Karyono. 2013. Pengukuran Beban Komputasi Algoritma Dijkstra, A*, dan Floyd-Warshall pada Perangkat Android. Program Studi Sistem Komputer Universitas Multimedia Nusantara Tangerang.
- Ibnu Katsir, Al- Imam Abu Fida Isma'il. 2004. Terjemahan Tafsir Ibn Katsir Juz 12. Jakarta: Sinar Baru AL- Gensindo
- Jek Siang, Jong. 2002. *Matematika Diskrit dan Aplikasinya pada Ilmu Komputer*. Yogyakarta: Andi.
- JimHyuk, Hong.2005.Evolving Reactive NPCs for the Real-Time SimulationGame. CIG.
- Kusumadewi, Sri. 2003. Artificial intelegence (Teknik dan Aplikasinya). Yogyakarta: Graha Ilmu.
- Kusumadewi, Sri. 2013.Aplikasi Logika Fuzzy untuk Pendukung Keputusan. Graha Ilmu : Yogyakarta. ISBN : 978-979-756-632-6
- Munir, Rinaldi. 2005. Matematika Diskrit. Bandung: Informatika.
- Neumann, J. Von dan O. Morgenstern. 1944. Theory of *Games* and Economic Behavior. Princeton New Jersey: Princeton University Press.

Nurochman. 2011. . Pendidikan Agama Islam untuk Siswa Sekolah Dasar Kelas

VI. Jakarta : Pusat Kurikulum dan Perbukuan, Kementrian Pendidikan Nasional.

Prasetya, Diyo Gilang & Pradesan, Iis. 2014. Penerapan Algoritma Dijkstra Dalam Animasi Lintasan Terpendek Dengan Visualisasi 3D Menggunakan Maya. Jurnal Teknik Informatika STMIK GI MDP.

Rahman, Ainur. 2015. Implementasi Fuzzy Sugeno Untuk Mengatur Bentuk Bonus Dari Konten Islami (ILMU TAJWID HUKUM BACAAN NUN SUKUN ATAU TANWIN) Pada *Game* 3D Battle Jet. Teknik Informatika Universitas Islam Negri Maulana Malik Ibrahim Malang.

Ratnasari, Asti dkk. Penentuan Jarak Terpendek dan Jarak Terpendek Alternatif Menggunakan Algoritma Dijkstra Serta Estimasi Waktu Tempuh. Magister Teknik Informatika Universitas Islam Indonesia Yogyakarta.

Santrock. 2006. Life Span Development: Perkembangan Masa Hidup. Jakarta : PT Erlangga.

Suyanto.2007.Artificial Intelligence.Informatika : Bandung. ISBN : 978-602-8758-20-8.

Zoharudin, Uay dkk. 2011. Pendidikan Agama Islam untuk Siswa Sekolah Dasar Kelas IV. Jakarta : Pusat Kurikulum dan Perbukuan, Kementrian Pendidikan Nasional.

Wati, Dwi Ana Ratna. 2011. Sistem Kendali Cerdas. Yogyakarta: Garaha Ilmu.

Yunifa M.A. dkk. Pergantian Senjata NPC pada *Game* FPS Menggunakan Fuzzy sugeno. Malang.

<http://www.storyboardthat.com/> Diakses 10 Mei 2016

<http://tf3dm.com/3d-model/> Diakses 10 Mei 2016

<http://www.tafsirqu.com/2015/06/pdf-ebook-tafsir-ibnu-katsir-tafsir.html>. Diakses

02 Februari 2017

