

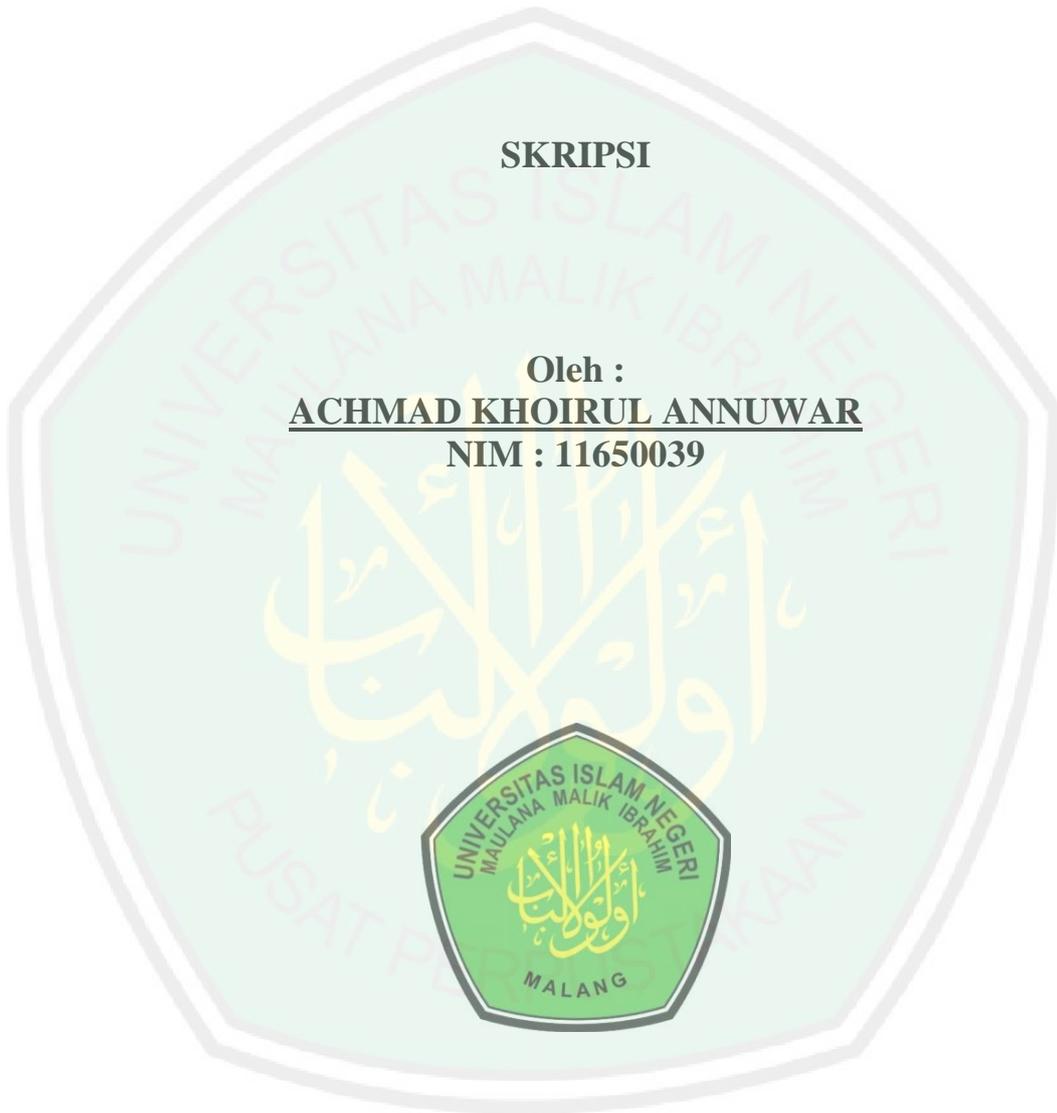
**DESAIN PERILAKU NPC MENGGUNAKAN METODE
ALGORITMA GENETIKA PADA GAME
PENGENALAN HURUF HIJAIYAH**

SKRIPSI

Oleh :

ACHMAD KHOIRUL ANNUWAR

NIM : 11650039



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2015**

**DESAIN PERILAKU NPC MENGGUNAKAN METODE
ALGORITMA GENETIKA PADA GAME
PENGENALAN HURUF HIJAIYAH**

SKRIPSI

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

Oleh :
ACHMAD KHOIRUL ANNUWAR
NIM : 11650039

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2015**

**DESAIN PERILAKU NPC MENGGUNAKAN METODE
ALGORITMA GENETIKA PADA GAME
PENGENALAN HURUF HIJAIYAH**

SKRIPSI

Oleh :

Nama : Achmad Khoirul Annuwar
NIM : 11650039
Jurusan : Teknik Informatika
Fakultas : Sains dan Teknologi

Telah Disetujui, 6 Juli 2015

Dosen Pembimbing I

Dosen Pembimbing II

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

Fachrul Kurniawan, M.MT
NIP. 19770120 200912 1 001

Mengetahui,

Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdiان
NIP. 19740424 200901 1 008

**DESAIN PERILAKU NPC MENGGUNAKAN METODE
ALGORITMA GENETIKA PADA GAME
PENGENALAN HURUF HIJAIYAH**

SKRIPSI

Oleh :

ACHMAD KHOIRUL ANNUWAR
NIM. 11650039

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

Tanggal, 6 Juli 2015

Susunan Dewan Penguji

Tanda Tangan

- | | | |
|------------------|-------------------------------------|-----|
| 1. Penguji Utama | : <u>Fresy Nugroho, M.T</u> | () |
| | NIP. 19710722 201101 1 001 | |
| 2. Ketua | : <u>Hani Nurhayati, M.T</u> | () |
| | NIP. 19780625 200801 2 006 | |
| 3. Sekretaris | : <u>Yunifa Miftachul Arif, M.T</u> | () |
| | NIP. 19830616 201101 1 004 | |
| 4. Anggota | : <u>Fachrul Kurniawan, M.MT</u> | () |
| | NIP. 19771020 200912 1 001 | |

**Mengetahui dan Mengesahkan,
Ketua Jurusan Teknik Informatika**

Dr. Cahyo Crysdiان
NIP. 19740424 200901 1 008

ORISINALITAS PENELITIAN

Saya yang bertandatangan di bawah ini :

Nama : Achmad Khoirul Annuwar
NIM : 11650039
Fakultas/Jurusan : Sains dan Teknologi / Teknik Informatika
Judul Penelitian : Desain Perilaku NPC Menggunakan Metode
Algoritma Genetika Pada Game Pengenalan Huruf
Hijaiyah

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 di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 6 Juli 2015

Yang Membuat Pernyataan,

Achmad Khoirul A
NIM. 11650039

MOTTO

*“Janganlah kamu bersikap lemah, dan janganlah pula kamu bersedih hati,
padahal kamulah orang-orang yang paling tinggi derajatnya,
jika kamu orang-orang yang beriman”*

(Q.S. Al-Imran: 139)

*“Seorang mukmin yang berbaur dengan manusia dan bersabar atas celaan mereka
adalah lebih besar pahalanya dari pada orang mukmin yang tidak membaur
dengan manusia dan tidak sabar atas celaan mereka”*

(HR Ahmad No 4780)

*“The Mind is everything
What you think
You Become”*

(Buddha)

“Diam adalah lebih baik daripada mengucapkan kata-kata yang tanpa makna”

(Phythagoraz)

PERSEMBAHAN

Dengan rasa syukur karya ini kusembahkan untuk :

Allah SWT yang telah senantiasa menjaga, melindungi dan memberiku rahmat yang tak terhingga.

Orang tua yang telah memberikan doa, cinta, dan kasih sayang sejak aku dilahirkan sampai saat ini.

Semua keluarga dan sahabat yang selalu menemani dan membantuku dalam segala hal.

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji bagi Allah SWT Tuhan seluruh alam yang telah melimpahkan rahmat serta karuniaNya kepada penulis sehingga bisa menyelesaikan skripsi dengan judul “Desain Perilaku NPC Menggunakan Metode Algoritma Genetika Pada Game Pengenalan Huruf Hijaiyah ” dengan baik.

Shalawat serta salam semoga tercurah kepada Nabi Agung Muhammad SAW atas segala bentuk kemapanan dan kejayaan yang beliau hadirkan bagi seluruh umat Islam di dunia, serta kepada semua keluarga, sahabat, dan para pengikut yang senantiasa meneruskan perjuangan saat ini hingga akhir zaman.

Penulis menyadari keterbatasan pengetahuan yang penulis miliki, karena itu tanpa keterlibatan dan sumbangsih dari berbagai pihak, sulit bagi penulis untuk menyelesaikan skripsi ini. Karena itu, penulis ingin mengucapkan terima kasih yang sebesar-besarnya dan penghargaan yang tak terhingga nilainya atas bantuan banyak orang, terutama kepada:

1. Bapak, Ibu dan Saudara tercinta yang selalu memberi motivasi serta doa yang senantiasa mengiringi setiap langkah penulis. Perjuangan dan keikhlasan kalian membuat penulis malu untuk tidak berkarya dan berprestasi

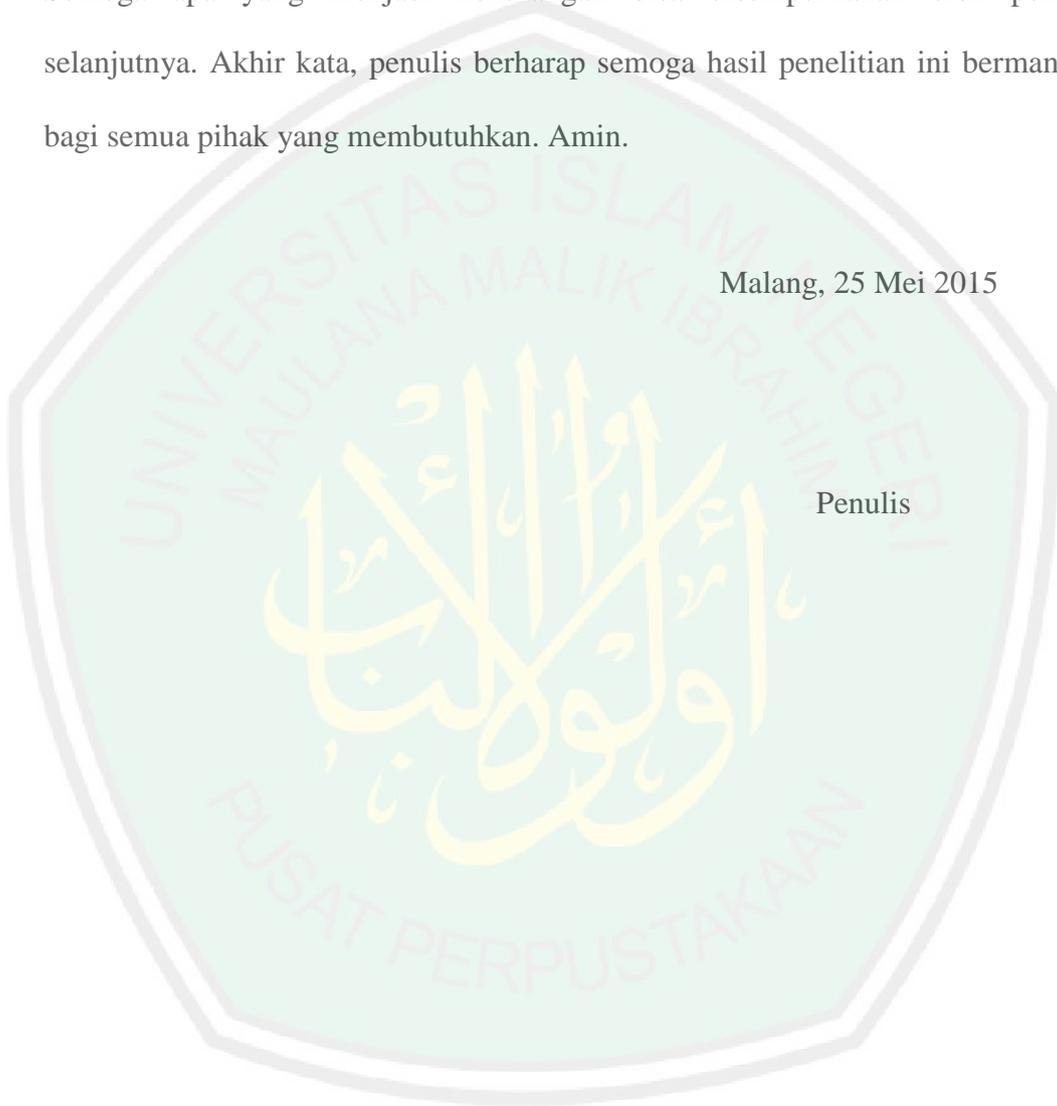
2. Bapak Yunifa Miftachul Arif, M.T, selaku dosen pembimbing I yang telah meluangkan waktu untuk membimbing, memotivasi, mengarahkan dan memberi masukan dalam pengerjaan skripsi ini.
3. Bapak Fachrul Kurniawan, M.MT selaku dosen pembimbing II yang juga senantiasa memberi masukan dan nasihat serta petunjuk dalam penyusunan skripsi ini.
4. Bapak Muhammad Ainul Yaqin, M.Kom, selaku dosen wali yang juga selalu memberi nasihat akademik kepada penulis selama masa *study*.
5. Segenap Dosen Teknik Informatika yang telah memberikan bimbingan keilmuan kepada penulis selama masa *study*.
6. Teman-teman Teknik Informatika yang telah memberikam motivasi, informasi dan masukan kepada penulis.
7. Seluruh keluarga besar Teater Komedi Kontemporer yang telah mengajarkan penulis tentang arti kehidupan dan cara bersikap.
8. Seluruh keluarga besar RTD Malang yang telah memberikan bimbingan, motivasi dan dukungan penuh kepada penulis.
9. Peneliti terdahulu tentang algoritma genetika dan game, atas segala yang telah diberikan kepada penulis dan dapat menjadi pelajaran.
10. Seluruh pihak yang secara langsung maupun tidak langsung yang telah ikut memberikan bantuan dan motivasi, yang tidak bisa disebutkan satu per satu

Penulis menyadari dalam skripsi ini masih banyak kekurangan dalam penyusunan skripsi ini. Oleh karena itu saran dan kritik yang membangun

diperlukan untuk menyempurnakan lebih lanjut, setidaknya sebagai pengingat bahwa skripsi ini hanya hasil karya manusia yang jauh dari kata sempurna. Semoga apa yang menjadi kekurangan bisa disempurnakan oleh peneliti selanjutnya. Akhir kata, penulis berharap semoga hasil penelitian ini bermanfaat bagi semua pihak yang membutuhkan. Amin.

Malang, 25 Mei 2015

Penulis



DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN.....	iv
HALAMAN PERNYATAAN	v
HALAMAN MOTTO.....	vi
HALAMAN PERSEMBAHAN.....	vii
KATA PENGANTAR	vii
DAFTAR ISI	xi
DAFTAR GAMBAR	xiv
DAFTAR TABEL	xvi
DAFTAR LAMPIRAN.....	xvii
ABSTRAK	xviii
BAB I PENDAHULUAN	
1.1. Latar Belakang	1
1.2. Identifikasi Masalah	4
1.3. Tujuan Penelitian	5
1.4. Manfaat Penelitian	5
1.5. Batasan Masalah.....	5
1.6. Sistematika Penulisan.....	6
BAB II TINJAUAN PUSTAKA	
2.1. Penelitian Terkait.....	8
2.2. Studi Literatur.....	11
2.2.1 Algoritma Genetika.....	11
2.2.2 Game Edukasi.....	15
2.2.3 Non-Player Character.....	16

2.2.4	Pengenalan Huruf Hijaiyah.....	17
2.2.5	Unity 3D (Free Version)	19
2.2.6	Bahasa Pemrograman C#.....	19
BAB III DESAIN DAN PERANCANGAN		
3.1.	Perencanaan dan Desain Aplikasi	21
3.1.1.	Perancangan Sistem.....	21
3.1.1.1.	Flowchart Algoritma Genetika.....	21
3.1.1.2.	Flowchart Seleksi Roulette Wheel.....	22
3.1.1.3.	Flowchart Cross-over.....	24
3.1.1.4.	Flowchart Mutasi.....	25
3.1.1.5.	FSM Game.....	26
3.1.1.6.	FSM Gameplay.....	27
3.1.1.7.	Analisa Kebutuhan Sistem.....	30
3.1.2.	Perancangan Game.....	31
3.1.2.1	Informasi Umum.....	31
3.1.2.2	Objektif Game.....	31
3.1.2.3	Target Game.....	32
3.1.2.4	Storyline.....	32
3.1.2.5	Konsep.....	33
3.1.2.6	Storyboard.....	33
3.1.2.7	Desain Environment.....	37
3.1.2.8	Desain Karakter.....	38
BAB IV HASIL DAN PEMBAHASAN		
4.1.	Implementasi Algoritma Genetika pada NPC	40
4.2.	Implementasi Permainan	54
4.2.1.	Model dan Karakter dalam Permainan.....	54
4.2.1.1.	Environment Game.....	54
4.2.1.2.	Model Karakter.....	54
4.2.1.3.	Model Huruf Hijaiyah.....	56

4.2.2.	Antarmuka dalam Permainan.....	57
4.2.2.1.	Splash Scene.....	57
4.2.2.2.	Menu Scene.....	57
4.2.2.3.	Petunjuk Permainan Scene.....	58
4.2.2.4.	Gameplay Scene.....	59
4.2.2.5.	Menang Scene.....	59
4.2.2.6.	Kalah Scene.....	60
4.2.3.	Gameplay Permainan.....	60
4.2.4.	Hasil Uji Coba Algoritma Genetika.....	62
4.2.5.	Integrasi Game Fight For Hijaiyah dengan Islam.....	66
 BAB V KESIMPULAN		
5.1.	Kesimpulan	68
5.2.	Saran.....	68

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar 3.1.	Flowchart Algoritma Genetika.....	21
Gambar 3.2.	Flowchart Seleksi	23
Gambar 3.3.	Flowchart Cross Over.....	24
Gambar 3.4.	Flowchart Mutasi	25
Gambar 3.5.	FSM Game	26
Gambar 3.6.	FSM Gameplay	28
Gambar 3.7.	Desain Splash Scene Game	34
Gambar 3.8.	Desain Menu Utama Game	34
Gambar 3.9.	Halaman Petunjuk Permainan	35
Gambar 3.10.	Halaman Gameplay	35
Gambar 3.11.	Halaman Pemain Menang	36
Gambar 3.12.	Halaman Pemain Kalah	36
Gambar 3.13.	Desain Environment	37
Gambar 3.14.	Sketsa Karakter Player	38
Gambar 3.15.	Sketsa Karakter NPC Musuh	39
Gambar 4.1.	Rentang Roulette Wheel.....	47
Gambar 4.2.	Environment Game	54
Gambar 4.3.	Model Karakter Pemain	55
Gambar 4.4.	Model Karakter NPC Musuh.....	55
Gambar 4.5.	Model Huruf Hijaiyah 3D	56
Gambar 4.6.	Splash Scene Game	57
Gambar 4.7.	Menu Scene Game	57
Gambar 4.8.	Petunjuk Permainan	58
Gambar 4.9.	Gameplay Scene	59
Gambar 4.10.	Scene Game Menang.....	59
Gambar 4.11.	Scene Game Kalah.....	60

Gambar 4.12. Gameplay Pemain Menghadapi MiniBoss.....	60
Gambar 4.13. Tampilan Mahraj dan Keterangan Huruf.....	61



DAFTAR TABEL

Tabel 2.1.	Huruf Hijaiyah dan Bacaan.....	18
Tabel 4.1.	Kromosom.....	41
Tabel 4.2.	Perilaku NPC.....	41
Tabel 4.3.	Individu dan Kromosom.....	42
Tabel 4.4.	Perilaku dan Kromosom.....	42
Tabel 4.5.	Individu Pertama.....	44
Tabel 4.6.	Individu Kedua.....	44
Tabel 4.7.	Individu Ketiga.....	44
Tabel 4.8.	Individu Keempat.....	45
Tabel 4.9.	Individu Fitness.....	45
Tabel 4.10.	Probabilitas Seleksi.....	46
Tabel 4.11.	Cross Over Individu I dan II.....	48
Tabel 4.12.	Hasil Cross Over.....	50
Tabel 4.13.	Identifikasi Mutasi.....	50
Tabel 4.14.	Pembangkitan Bilangan Acak.....	50
Tabel 4.15.	Perbandingan Probabilitas Mutasi.....	51
Tabel 4.16.	Pengacakan Kembali Kromosom.....	51
Tabel 4.17.	Hasil Mutasi.....	52
Tabel 4.18.	Populasi Baru.....	52
Tabel 4.19.	Evolusi Populasi.....	53

DAFTAR LAMPIRAN

- Lampiran 1. Pseudocode Pembentukan Individu
- Lampiran 2. Pseudocode Pembangkitan Populasi
- Lampiran 3. Pseudocode Seleksi
- Lampiran 4. Pseudocode Proses Cross Over
- Lampiran 5. Pseudocode Proses Mutasi



ABSTRAK

Annuwar, Achmad Khoirul 2015. **Desain Perilaku NPC Menggunakan Metode Algoritma Genetika Pada Game Pengenalan Huruf Hijaiyah**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Pembimbing : (I) Yunifa Miftachul Arif, M.T (II) Fachrul Kurniawan, M,M.T

Kata Kunci : *Algoritma Genetika, NPC, Perilaku*

Huruf hijaiyah merupakan dasar untuk membaca dan mempelajari kitab suci Al-Quran. Diperlukan metode pembelajaran khusus yang menarik sehingga dapat membantu anak-anak dalam belajar mengenali, memahami, mengerti pengucapan lafaz huruf hijaiyah dengan baik dan benar. Dan game adalah salah satu metode yang bisa digunakan untuk anak-anak dan orang dewasa. Kepuasan pemain dalam game komputer tergantung pada kontribusi perilaku dan strategi lawan dalam game. Algoritma genetika dapat memainkan peran sebagai sistem adaptif melalui reproduksi buatan dan evolusi. Maka, peneliti akan membangun aplikasi game pengenalan huruf hijaiyah dengan menggunakan metode algoritma genetika pada NPC untuk mengoptimalkan kepuasan pemain. Dengan tujuan penelitian; 1) Membuktikan bahwa Algoritma Genetika dapat digunakan untuk desain perilaku NPC pada game “*Fight For Hijaiyah*”, 2) Mengukur tingkat kecepatan peningkatan *fitness* yang terjadi dari generasi ke generasi pada NPC dengan metode Algoritma Genetika pada game “*Fight For Hijaiyah*”.

Penelitian ini dilakukan dalam dua tahap, yaitu perancangan sistem dan perancangan game. Tahapan perancangan sistem yaitu mendiskripsikan alur metode dan jalannya permainan. Tahapan perancangan game yaitu mendiskripsikan desain antarmuka, latar permainan dan karakter game.

Pada penelitian ini algoritma genetika diimplementasikan pada game “*Fight For Hijaiyah*” untuk desain perilaku NPC musuh. Algoritma genetika dalam game bekerja ketika NPC musuh pertama kali *spawning*, setiap individu dalam hal ini adalah NPC musuh disisipkan karakteristik dan perilaku sesuai dengan aturan tertentu. Perilaku dalam individu disesuaikan dengan kelemahan individu tersebut, sebagai contoh ketika individu lemah pada kecepatan jalan, individu itu akan memiliki perilaku menghilang. Setiap generasinya individu akan mengalami evolusi dan akan menambah kesulitan permainan. Hasil uji coba algoritma genetika, dari 13 iterasi generasi atau populasi diperoleh prosentase tingkat kecepatan peningkatan *fitness* yang terjadi dari generasi ke generasi pada NPC sebesar 84.6%

ABSTRACT

Annular, Achmad Khoirul 2015. **NPC Behaviour Design Using Genetic Algorithm Method In Introduction Hijaiyah Letter Game**. Thesis. Department of Informatics Faculty of Science and Technology of State Islamic University of Maulana Malik Ibrahim Malang.

Advisors: (I) Yunifa Miftachul Arif, M.T (II) Fachrul Kurniawan, M,M.T

Kata Kunci : *Genetic Algorithms, NPC, Behavior, Hijaiyah Letters*

Hijaiyah letter is the basis for reading and studying the holy book of Al-Quran therefore extremely important to introduce the letter hijaiyah early on, it would require a special teaching methods that draw so that it can help children to learn to recognize, understand, understand the pronunciation of the letter hijaiyah well and true. And the game is one of the methods that can be used for children and adults. Satisfaction players in computer games depends on the contribution of the behavior and the strategy of opponents in the game. Genetic algorithms can play a role as an adaptive system through artificial reproduction and evolution. Then, researchers will build gaming applications hijaiyah letter recognition using a genetic algorithm to optimize NPC player satisfaction. With research purposes; 1) Prove that Genetic Algorithm can be used to design the behavior of NPCs in the game "Fight For hijaiyah", 2) Measuring the speed of increase in fitness levels that occur from generation to generation in the NPC with the Genetic Algorithm method in game "Fight For Hijaiyah".

This research was conducted in two stages, namely the system design and game design. The system design stages that describe flow methods and course of the game. Stages of designing games that describe interface design , background games and game characters .

In this study, genetic algorithm implemented in the game "Fight For Hijaiyah" for the design of the enemy NPC behavior. Genetic algorithms in the game works when enemy NPCs spawning, every individual in this case is the enemy NPC inserted characteristics and behavior in accordance with certain rules. Behavior in individuals adjusted to the individual weaknesses, for example when the individual is weak on movement speed, the individual would have invisibility behavior. Each generation of individuals will evolve and will add to the difficulty of the game. The trial results of genetic algorithm, of 13 iterations generation or gained population percentage increase in fitness level of speed that occurs from generation to generation in the NPC at 84.6%

مستخلص

أنوار، أحمد خيرول ٢٠١٥. تصميم السلوك NPC باستخدام الخوارزمية الجينية في التعرف اللعبة الأحروف الهجائية. البحث العلمي. شعبة المعلوماتية كلية العلوم والتكنولوجي بالجامعة الحكومية الإسلامية مولانا مالك إبراهيم مالانج.
المشرف: (١) يونيتا مفتاح العارف، الماجستير، (٢) فخر الكورنياوان الماجستير.
الكلمة : الخوارزمية الجينية، NPC، الأحروف الهجائية

الهجائية هي الأساس للقراءة ودراسة القرآن لذا فإن من المهم للغاية أن يعرض الهجائية الالكترونية في المبكر. واللعبة هي الواحدة من الطرق التي يمكن استخدامها للأطفال والكبار. و ارتياح اللاعبين في ألعاب الكمبيوتر يعتمد على مساهمة السلوك والاستراتيجية من المعارضين في اللعبة. يمكن الخوارزميات الجينية تلعب دورا كنظام التكيف من خلال التكاثر الاصطناعي والتطور. ثم، فإن الباحثين بناء تطبيقات الألعاب الهجائية الالكتروني الاعتراف باستخدام الخوارزمية الجينية لتحسين NPC رضا لاعب. مع أغراض البحوث؛ (١) ان يثبت الخوارزميات الجينية يمكن استخدامها لتصميم سلوك الشخصيات في اللعبة "الكفاح من الهجائية"، (٢) قياس سرعة الزيادة في مستويات اللياقة البدنية التي تحدث من جيل إلى جيل في مجلس الشعب مع أسلوب الخوارزميات الجينية في اللعبة "الكفاح من الهجائية".

وقد أجريت هذه الدراسة على مرحلتين، وهما تصميم النظام وتصميم لعبة المراحل التي تصف طريقة التدفقات تصميم النظام و سير المباراة . التي تصف تصميم لعبة مرحلة تصميم واجهة ، والألعاب الخلفية و شخصيات اللعب.
في هذه الدراسة، الخوارزمية الجينية تنفيذها في لعبة "الكفاح من الهجائية" لتصميم سلوك مجلس الشعب العدو. الخوارزميات الجينية في اللعبة تعمل عند العدو الشخصيات

التفريخ، كل فرد في هذه الحالة هو مجلس الشعب العدو خصائص المدرجة والسلوك وفقا لقواعد معينة. السلوك لدى الأفراد تعديلها لنقاط الضعف الفردية. وكل جيل من الأفراد تتطور وسيضيف إلى صعوبة المباراة. نتائج محاكمة الخوارزمية الجينية، من ١٣ جيل التكرار أو السكان اكتسبت نسبة الزيادة في مستوى اللياقة البدنية من السرعة التي تحدث من جيل إلى جيل في مجلس الشعب في ٨٤,٦٪.



BAB I

PENDAHULUAN

1.1. Latar Belakang

Huruf hijaiyah merupakan dasar untuk membaca dan mempelajari kitab suci Al-Quran oleh sebab itu sangat penting mengenalkan huruf hijaiyah sejak dini, karena apabila sudah sering membaca huruf hijaiyah dengan ucapan yang salah maka akan terbiasa dan dapat menimbulkan arti yang berbeda sehingga kita dituntut untuk membaca secara baik dan benar. Dalam pengenalan dan pembelajaran huruf hijaiyah kepada anak-anak harus dengan metode yang menarik perhatian. Dewasa ini sudah beraneka ragam metode pembelajaran Al-Quran, antara lain adanya buku paduan membaca Al-Quran, diterbitkannya buku Iqra untuk anak-anak, meskipun demikian masyarakat khususnya anak-anak banyak yang belum bisa membaca Al-Quran. Kurangnya minat membaca ini disebabkan metode pembelajaran Al-Quran yang kurang menarik dan cenderung monoton (Ernawati, 2012), ditambah lagi bila hanya membaca buku tidak akan bisa mendengar lafaz yang baik dan benar. Diperlukan metode pembelajaran khusus yang menarik sehingga dapat membantu anak-anak dalam belajar mengenali, memahami, mengerti pengucapan lafaz huruf hijaiyah dengan baik dan benar.

Surat Al Baqarah ayat 121 :

الَّذِينَ ءَاتَيْنَاهُمُ الْكِتَابَ يَتْلُونَهُ حَقَّ تِلَاوَتِهِ أُولَٰئِكَ يُؤْمِنُونَ بِهِۦٓ وَمَنْ يَكْفُرْ بِهِۦٓ فَأُولَٰئِكَ هُمُ الْخٰسِرُونَ ۙ ۱۲۱

“121. Orang-orang yang telah Kami berikan Al Kitab kepadanya, mereka membacanya dengan bacaan yang sebenarnya, mereka itu beriman kepadanya. Dan barangsiapa yang ingkar kepadanya, maka mereka itulah orang-orang yang rugi.”

Tafsir dan penjelasan QS. Al-Baqarah ayat 121 dalam kitab tafsir Ibnu Katsir bahwasannya Abdur Razzaq meriwayatkan dari Ma'mar, dari Qatadah, yang dimaksud dengan orang-orang tersebut adalah orang Yahudi dan Nasrani. Pendapat ini merupakan Abdur Rahman ibnu Zaid Aslam dan dipilih oleh Ibnu Jarir. Sa'id meriwayatkan Qatadah, bahwa mereka adalah sahabat-sahabat Rasulullah Saw. Ibnu Abu Hatim mengatakan, telah menceritakan kepada kami ayahku, telah menceritakan kepada kami Ibrahim ibnu Musa dan Abdullah ibnu Imran Al-Asbahani yang mengatakan, telah menceritakan kepada Yahya ibnu Yaman, telah menceritakan kepada kami Usama ibnu Zaid, dari ayahnya, dari Umar ibnul Khattab, sehubungan dengan tafsir firman-Nya, *”Mereka membacanya dengan bacaan yang sebenarnya”* (Al-Baqarah 121). Yang dimaksud dengan bacaan yang sebenarnya ialah apabila si pembaca melewati penyebutan tentang surga, maka dia memohon surga kepada Allah. Apabila ia melewati penyebutan neraka, maka ia meminta perlindungan dari neraka.

Game Komputer merupakan program komputer yang memungkinkan pemain dapat berinteraksi lewat kontrol untuk memenuhi tujuan tertentu. Game komputer banyak dimainkan oleh berbagai kalangan dari anak-anak sampai orang tua. Selain memberikan kesenangan kepada pemain game tersebut, game juga dapat menjadi sarana untuk melatih kecerdasan, kemampuan menyelesaikan masalah,

kemampuan menganalisa dan dapat memberikan informasi atau wawasan yang ada dalam game tersebut. Telah dikemukakan bahwa pengembangan teknologi informasi tidak hanya mempengaruhi bentuk bermain, tetapi juga membawa game sebagai sebuah konsep lebih jelas ke dalam kehidupan anak-anak dimasa yang akan datang (Sheridan S, Pramling-Samuelsson I. 2003). Game edukasi unggul dalam beberapa aspek jika dibandingkan dengan metode pembelajaran konvensional. Salah satu keunggulan yang signifikan adalah adanya animasi yang dapat meningkatkan daya ingat sehingga anak dapat menyimpan materi pelajaran dalam waktu yang lebih lama dibandingkan dengan metode pengajaran konvensional (Clark, 2006).

Kepuasan pemain dalam game komputer tergantung pada berbagai faktor, diantara lain pada kontribusi perilaku dan strategi lawan dalam game. Lawan harus beradaptasi dengan cepat terhadap perubahan dalam strategi pemain, jika tidak, game menjadi membosankan. Sebuah metrik kuantitatif dari ' *interestingness* ' menyatakan perilaku lawan ditentukan berdasarkan pertimbangan kualitatif apa yang menyenangkan dalam game tersebut, dan formulasi matematis didasarkan pada data yang berasal dari observasi (Georgios N. Yannakakis 2005).

Holland (1975) menunjukkan bahwa algoritma genetika dapat memainkan peran sebagai sistem adaptif melalui reproduksi buatan dan evolusi. .Algoritma genetika diimplementasikan sebagai simulasi komputer di mana sebuah populasi dari representasi abstrak (disebut kromosom, genotipe, atau genom) kandidat solusi (disebut individu, makhluk, atau fenotipe). Biasanya, solusi direpresentasikan dalam biner sebagai string yang terdiri dari 0 dan 1, namun penggunaan metode

encoding lain juga mungkin dipakai. Berbeda dengan teknik pencarian konvensional, algoritma genetika bermula dari himpunan solusi yang dihasilkan secara acak. Kromosom-kromosom berevolusi dalam suatu proses iterasi yang berkelanjutan yang disebut generasi. Pada setiap generasi, kromosom dievaluasi berdasarkan suatu fungsi evaluasi. Setelah beberapa generasi maka algoritma genetika akan konvergen pada kromosom terbaik, yang diharapkan merupakan solusi optimal (Mohammad Gilang 2007).

Sesuai dengan permasalahan yang dikemukakan diatas, peneliti akan membangun aplikasi game pengenalan huruf hijaiyah dengan menggunakan metode algoritma genetika pada NPC untuk mengoptimalkan kepuasan pemain, diharapkan dengan game ini dapat membantu anak-anak dalam belajar mengenali, memahami, mengerti pengucapan lafaz huruf hijaiyah dengan baik dan benar.

1.2. Identifikasi Masalah

Adapun identifikasi masalah dari penelitian ini antara lain :

- a. Bagaimana mengimplementasikan Algoritma Genetika untuk desain perilaku NPC pada game “*Fight For Hijaiyah*”?
- b. Seberapa cepat peningkatan *fitness* yang terjadi dari generasi ke generasi pada NPC dengan metode Algoritma Genetika pada game “*Fight For Hijaiyah*”?

1.3. Tujuan Penelitian

Tujuan dari penelitian ini adalah :

- a. Membuktikan bahwa Algoritma Genetika dapat digunakan untuk desain perilaku NPC pada game “*Fight For Hijaiyah*”.
- b. Mengukur tingkat kecepatan peningkatan *fitness* yang terjadi dari generasi ke generasi pada NPC dengan metode Algoritma Genetika pada game “*Fight For Hijaiyah*”.

1.4. Manfaat Penelitian

Aplikasi game ini dibangun untuk menunjang pembelajaran pengenalan huruf hijaiyah yang menggunakan simulasi 3D agar memberikan informasi yang menarik dan edukatif kepada pemain, selain itu dilengkapi mahraj dan bunyi tiap-tiap huruf hijaiyah memberikan informasi tentang pengucapan lafaz huruf hijaiyah yang baik dan benar.

1.5. Batasan Masalah

Agar penelitian ini tidak menyimpang dari permasalahan maka perlu adanya batasan masalah, yaitu :

- a. Aplikasi hanya dapat digunakan pada platform desktop.

- b. Aplikasi game hanya dapat berjalan pada operasi system Windows 7, 8 dan 8.1.
- c. Aplikasi game direkomendasikan untuk anak-anak usia 5 sampai 10 tahun.
- d. Konten permainan berdasarkan materi-materi dasar pengenalan huruf hijaiyah.
- e. Aplikasi dibuat dengan menggunakan Unity 3D versi 5.0 (*Free Version*)

1.6. Sistematika Penulisan

Penulisan skripsi ini tersusun dalam lima bab dengan sistematika penulisan sebagai berikut :

BAB I Pendahuluan

Pendahuluan, membahas tentang latar belakang masalah, identifikasi masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penyusunan tugas akhir.

BAB II Landasan Teori

Landasan teori berisikan beberapa teori yang mendasari dalam penyusunan tugas akhir ini. Adapun yang dibahas dalam bab ini adalah dasar teori yang berkaitan dengan pembahasan game edukasi, algoritma genetika, dan NPC.

BAB III Desain dan Perancangan

Menganalisa kebutuhan sistem untuk membuat program aplikasi meliputi spesifikasi kebutuhan software dan langkah-langkah pembuatan program aplikasi game pengenalan huruf hijaiyah dengan menggunakan metode algoritma genetika pada perilaku NPC.

BAB IV Hasil dan Pembahasan

Menjelaskan tentang pengujian program aplikasi game pengenalan huruf hijaiyah dengan menggunakan metode Algoritma Genetika pada perilaku NPC yang telah diterapkan.

BAB V Penutup

Berisi kesimpulan dan saran yang diharapkan dapat bermanfaat untuk pengembangan pengembangan aplikasi selanjutnya.

BAB II

KAJIAN PUSTAKA

2.1. Penelitian terkait

Omid E. David, et al, 2014, penelitian yang mendemostrasikan game catur dengan menggunakan algoritma genetika. Penelitian tersebut mengembangkan algoritma genetika dengan fungsi 1). Evaluasi pemain tingkat *grandmaster* 2). Mekanisme pencarian pada game catur, nilai parameter yang dipakai diinisialisasi secara acak. Fungsi evaluasi dipelajari dari *knowledge database* pemain tingkat *grandmaster*. Hasil penelitian menunjukkan bahwa program berevolusi melebihi juara komputer catur dunia.

Nirvana S. Antonio, et al, 2013 penelitian tentang fungsi evaluasi untuk langkah terbaik pada game Domino menggunakan algoritma genetika (GA). Fungsi evaluasi terdiri dari penggabungan strategi game didefinisikan sebagai penekanan ke lawan , dan gerak adaptif ke lawan. Hasil yang diperoleh selama simulasi menunjukkan bahwa tim (terdiri dari dua pemain) menggunakan fungsi evaluasi dengan koefisien dioptimalkan oleh GA menang lebih dari 69,18 % dari total pertandingan.

Dini Nur Setyorini, et al, 2012 dalam penelitian mereka menyimpulkan hasil perhitungan nilai *fitness* untuk waktu terbaik atau penyelesaian terbaik kurang akurat. Proses algoritma genetika sangatlah berpengaruh pada hasil random dan iterasi. Apabila dilakukan percobaan ulang, hasil yang diperoleh akan

lebih baik atau lebih buruk. Berdasarkan percobaan dan perbandingan hasil optimasi GA dengan perhitungan manual yang nyata dilapangan, sistem ini terbukti dapat menyelesaikan masalah sehingga didapatkan waktu terbaik dan optimal.

Moshe Sipper, et al, 2008, dalam penelitiannya merumuskan pemrograman genetika bertindak sebagai alat untuk mengembangkan strategi dan desain mesin untuk penyusun strategi. Hasilnya menyatakan pemrograman genetika pada permainan telah berevolusi dengan kecerdasan buatan, menghasilkan strategi yang kompetitif menyaingi logika manusia. Game mampu menahan atau beradaptasi sesuai kemampuan manusia, bahkan seringkali menang terhadap manusia.

Hans K. G. Fernlund, et al, 2006 dalam penelitiannya membahas perilaku rekan tim pada game simulasi militer dengan algoritma GA. Peneliti memberikan pendekatan baru yang mempekerjakan pemrograman genetika dalam hubungannya dengan penalaran berbasis konteks perkembangan agen taktis berdasarkan observasi otomatis manusia melakukan misi di simulator . Kesimpulan dari hasil penelitian tersebut adalah program genetika mampu menciptakan agen baru dengan mempelajari performa, taktik dan strategi manusia pada simulasi tersebut.

Francisco Azuaje, 2003, dalam penelitiannya mengangkat tema pendekatan komputasi evolusioner pada strategi game dan kerjasama makhluk virtual. Penelitian ini menggunakan algoritma generika dimana akan dicari satu per satu individu-individu yang memiliki nilai *fitness* kompetitif. Satu individu

mengimplementasikan aturan game didasarkan dari seperangkat kemampuan kognitif. Individu juga menentukan pola strategi game yang menarik.

Utomo Sarjono Putro, et al, 2000, penelitian tentang pembelajaran adaptif untuk mendukung kelompok pembuat keputusan dengan seperangkat strategi dan preferensi yang menghadapi perilaku tidak menentu. Menggambarkan situasi keputusan sebagai situasi hypergame, di mana setiap pengambil keputusan secara eksplisit diasumsikan memiliki kesalahan persepsi tentang set strategi alam. GA dipakai untuk menoptimalkan presepsi terdapat 3 prosedur belajar yang berbeda satu sama lain sehubungan seperti evaluasi *fitness*, *cross-over* yang dimodifikasi, dan pilihan tindakan strategi,

Dalam penelitian Sayed Fachrurrazi, 2012, menyebutkan algoritma genetika untuk penyelesaian masalah pencarian jarak terpendek dikatakan dapat mengatasi masalah. Optimasi ditekankan pada pencarian jalur terpendek, dimana akan dicari beberapa alternatif solusi penyelesaian yang lebih efektif dan efisien. Algoritma genetika terbukti dapat memberi solusi jalur yang lebih baik.

Dalam penelitian Johan Saputra, 2008, et al, menggunakan algoritma genetika mendapat rata-rata tingkat keberhasilan untuk keseluruhan pengujian adalah 70.80%. Tingkat nilai *fitness cost* dipengaruhi oleh jumlah data-data yang diproses dalam perangkat lunak. Metode cukup baik untuk digunakan dalam obyek penelitiannya terlebih lagi prosentase keberhasilan sangat tinggi.

Fathelalem F. Ali, et al, 2000, penelitiannya menjelaskan tentang strategi NPC pada game *Rock-Paper-Scissor*. NPC bias membuat keputusan, dan aturan diadopsi dari game dalam beberapa putaran terakhir, terus diamati dan kemudian

pengetahuan sementara dibuat. Proses penyusunan strategi seperti pengkodean ke string genetika dan algoritma genetika (GA) bekerja pada populasi string tersebut. String yang memiliki nilai *fitness* diproduksi di generasi berikutnya. Kesimpulan penelitian ini adalah algoritma genetika dapat mengubah game NPC sama baiknya dengan game manusia.

2.2. Studi Literatur

2.2.1 Algoritma genetika

Algoritma ini ditemukan di Universitas Michigan, Amerika Serikat oleh John Holland (1975) melalui sebuah penelitian dan dipopulerkan oleh salah satu muridnya, David Goldberg (1989). Dimana mendefinisikan algoritma genetika ini sebagai metode algoritma pencarian berdasarkan pada mekanisme seleksi alam dan genetika alam. Algoritma genetika adalah algoritma yang berusaha menerapkan pemahaman mengenai evolusi alamiah pada tugas-tugas pemecahan-masalah (*problem solving*). Pendekatan yang diambil oleh algoritma ini adalah dengan menggabungkan secara acak berbagai pilihan solusi terbaik di dalam suatu kumpulan untuk mendapatkan generasi solusi terbaik. Metode ini juga akan merepresentasikan perbaikan-perbaikan pada populasi dari setiap generasi. Dengan melakukan proses ini secara berulang, algoritma ini diharapkan dapat mensimulasikan proses evolusioner.

Pada akhirnya, akan didapatkan solusi-solusi yang paling tepat bagi permasalahan yang dihadapi. Dalam menggunakan algoritma genetika, solusi permasalahan direpresentasikan sebagai kromosom. Tiga aspek yang penting untuk penggunaan algoritma genetika:

- a. Defenisi fungsi *fitness*
- b. Defenisi dan implementasi representasi genetika
- c. Defenisi dan implementasi operasi geneti

Algoritma genetika memberikan suatu pilihan bagi penentuan nilai parameter dengan meniru cara reproduksi genetika, pembentukan kromosom baru serta seleksi alami seperti yang terjadi pada makhluk hidup. Golberg (1989) mengemukakan bahwa algoritma genetika mempunyai karakteristik-karakteristik yang perlu diketahui sehingga dapat terbedakan dari prosedur pencarian atau optimasi yang lain, yaitu:

- a. Algoritma genetika dengan pengkodean dari himpunan solusi permasalahan berdasarkan parameter yang telah ditetapkan dan bukan parameter itu sendiri.
- b. Algoritma genetika pencarian pada sebuah solusi dari sejumlah individu-individu yang merupakan solusi permasalahan bukan hanya dari sebuah individu.

- c. Algoritma genetika informasi fungsi objektif (*fitness*), sebagai cara untuk mengevaluasi individu yang mempunyai solusi terbaik, bukan turunan dari suatu fungsi.
- d. Algoritma genetika menggunakan aturan-aturan transisi peluang, bukan aturan-aturan deterministik.

Variabel dan parameter yang digunakan pada algoritma genetika adalah:

- a. Fungsi *fitness*(fungsi tujuan) yang dimiliki oleh masing-masing individu untuk menentukan tingkat kesesuaian individu tersebut dengan criteria yang ingin dicapai.
- b. Populasi jumlah individu yang dilibatkan pada setiap generasi.
- c. Probabilitas terjadinya persilangan (*cross-over*) pada suatu generasi.
- d. Probabilitas terjadinya mutasi pada setiap individu.
- e. Jumlah generasi yang akan dibentuk yang menentukan lama penerapan algoritma genetika

Pada algoritma ini, teknik pencarian dilakukan sekaligus atas sejumlah solusi yang mungkin yang dikenal dengan istilah populasi. Individu yang terdapat dalam satu populasi disebut dengan istilah

kromosom. Kromosom ini merupakan kumpulan gen-gen yang berbentuk simbol. Populasi awal dibangun secara acak, sedangkan populasi berikutnya merupakan hasil evolusi kromosom-kromosom melalui iterasi yang disebut dengan fungsi *fitness*. Nilai *fitness* dari suatu kromosom akan menunjukkan kualitas kromosom dalam populasi tersebut. Gambaran algoritma sederhana dari metode algoritma genetika ini sendiri dapat terlihat seperti berikut ini :

(Ricketts, 1997)

$t := 0;$

Generate Initial Population(Pt)

Evaluate(Pt)

Repeat

Inc(t)

Generate Population(Pt) from Population(Pt-1)

Evaluate(Pt)

Until Solution found or time runs out.

Proses standar dalam Algoritma Genetika (Pandjaitan, 2007):

- a. Membuat populasi awal dari n buah gen membentuk suatu kromosom.

- b. Menghitung *fitness cost* dari generasi pertama.
- c. Melakukan pengulangan proses regenerasi sebagai berikut :
 - a) Seleksi: memilih kromosom terbaik.
 - b) *Cross-over*
 - c) Mutasi
 - d) Memasukkan kromosom-kromosom hasil proses kedalam generasi baru.
 - e) Memproses generasi baru tersebut untuk proses selanjutnya.
 - f) Evaluasi apakah proses akan diulang.
 - g) Kembali ke langkah b.

2.2.2. Game Edukasi

Game yang memiliki konten pendidikan lebih dikenal dengan istilah game edukasi. Game berjenis edukasi ini bertujuan untuk memancing minat belajar anak terhadap materi pelajaran sambil bermain, sehingga dengan perasaan senang diharapkan anak bisa lebih mudah memahami materi pelajaran yang disajikan. jenis ini sebenarnya lebih mengacu kepada isi dan tujuan game, bukan jenis

yang sesungguhnya. Menurut Edward (2009) game merupakan sebuah media yang efektif untuk mengajar karena mengandung prinsip-prinsip pembelajaran dan teknik instruksional yang efektif digunakan dalam penguatan pada level-level yang sulit. Menurut Virvou (2005) teknologi game edukasi dapat memotivasi pembelajaran dan melibatkan pemain, sehingga proses pembelajaran lebih menyenangkan. Demikian pula hasil penelitian yang dilakukan oleh Randel pada tahun 1991, tercatat bahwa pemakaian game sangat bermanfaat pada materi-materi yang berhubungan dengan matematika, fisika dan kemampuan berbahasa (Dillon, 2004). Dalam konteks pembelajaran, penggunaan game edukasi dalam pembelajaran matematika merupakan hal yang positif. Hal ini sesuai dengan pendapat Strangman & Hall (2003) yang menyatakan bahwa game komputer menjadi sebuah pendekatan yang efektif untuk meningkatkan pembelajaran siswa. Tiga hasil belajar utamanya telah ditunjukkan yakni perubahan secara konseptual, pengembangan ketrampilan dan bidang pengetahuan.

2.2.3. Non-Player Character

Menurut Febrian Bahari Adi, et al, (2010) *Non-Player Characters (NPC)* atau disebut juga agen adalah suatu entitas dalam game yang tidak dikendalikan secara langsung oleh pemain. NPC

dikendalikan secara otomatis oleh komputer. NPC bisa berupa teman, musuh atau netral. NPC diinginkan dapat berperilaku cerdas layaknya manusia. Dia bisa mengindra lingkungan, berpikir, memilih aksi lalu bertindak sebagai respon atas perubahan pada lingkungannya. Untuk dapat memperoleh perilaku cerdas dari NPC digunakan kecerdasan buatan atau *Artificial Intelligence* (AI). Penggunaan AI pada NPC dilakukan dengan pemberian algoritma khusus sesuai dengan perilaku cerdas yang diharapkan.

Pada adegan (*scene*) pertempuran dari game computer, prajurit pemain dan prajurit musuh merupakan contoh dari NPC. Perilaku seorang prajurit dalam medan pertempuran bervariasi mulai dari mengikuti pimpinan, menghindari halangan, berlari, berjalan, menjauhi musuh, bertarung, membantu teman, dan lainnya

2.2.4. Pengenalan Huruf Hijaiyah

Huruf hijaiyah tidak bisa dipisahkan dari bahasa arab. Baik dalam membaca Al-Quran, sholat, dzikir dan sebagainya. Huruf Hijaiyah merupakan huruf penyusun kata dalam Al Qur an seperti halnya di Indonesia yang memiliki huruf alfabet dalam menyusun sebuah kata menjadi kalimat, huruf hijaiyah juga memiliki peran yang sama. Terdapat 30 huruf dasar (asas/asli) di dalam Al-Quran

dan 2 huruf pengganti yang dikenal juga dengan nama huruf-huruf Hijaan atau Hijaiyah, yaitu:

Tabel 2.1. Huruf Hijaiyah dan Bacaan

Huruf	Bacaan
ا	Alif
ب	Ba
ت	Ta
ث	Tsa
ج	Jim
ح	Ha
خ	Kha
د	Dal
ذ	Dzal
ر	Ra
ز	Zal
س	Sin
ش	Syin
ص	Syad
ض	Dhad
ط	Tha''
ظ	Zha''
ع	'ain
غ	Ghain
ف	Fa
ق	Qaf
ك	Kaf
ل	Lam
م	Mim
ن	Na
و	Wau
ه	Ha
ء	Hamzah
-	Lam alif
ي	Ya

Sumber :Jane Wightwick dan Mahmoud Gaafar, 2005

2.2.7. Unity 3D (*Free Version*)

Prof Fabian Winkler, 2013, Unity Engine suatu game engine yang terus berkembang. Penggunaan engine versi free dibatasi dengan beberapa fitur yang dikurangi atau bonus modul / *prefab* tertentu yang ditiadakan dan hanya tersedia untuk pengguna berbayar. Unity Engine dapat mengolah beberapa data seperti objek tiga dimensi, suara, tekstur, dan lain sebagainya. Keunggulan dari unity engine ini dapat menangani grafik dua dimensi dan tiga dimensi. Namun engine ini lebih konsentrasi pada pembuatan grafik tiga dimensi. Dari beberapa game engine yang sama-sama menangani grafik tiga dimensi, Unity Engine dapat menangani lebih banyak. Sistem inti engine ini menggunakan beberapa pilihan bahasa pemrograman, diantaranya C#, javascript maupun boo. Unity3D editor menyediakan beberapa alat untuk mempermudah pengembangan yaitu Unity Tree dan *terrain creator* untuk mempermudah pembuatan vegetasi dan terrain serta MonoDevelop untuk proses pemrograman().

2.2.8. Bahasa Pemrograman C#

Andrew Stellman & Jenniffer Greene, 2010, C# (disebut juga sebagai “C sharp”) adalah sebuah bahasa pemrograman *object-oriented* seperti java dan merupakan bahasa *component-*

oriented yang pertama. Bahasa pemrograman ini dibuat berbasiskan bahasa C++ yang telah dipengaruhi oleh aspek-aspek ataupun fitur bahasa yang terdapat pada bahasa-bahasa pemrograman lainnya seperti java, visual basic, dan lain-lain dengan beberapa penyerdehanaan.: Beberapa kelebihan dari C# yaitu :

- a. Kode pada C# lebih sederhana dan lebih modern.
- b. Kombinasi fitur-fitur terbaik dari beberapa bahasa pemrograman.
- c. Memiliki sintaks yang konsisten.

BAB III

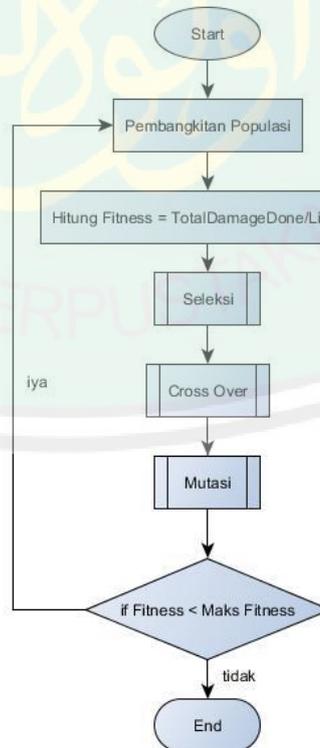
DESAIN DAN PERANCANGAN

3.1. Perencanaan dan Desain Aplikasi

Sistem dalam penelitian ini dibagi menjadi dua, yaitu: sistem aplikasi serta algoritma dan desain tatap muka (*interface*). Dalam sistem aplikasi serta algoritma dibahas mengenai sistem kerja atau alur proses utama. Dalam desain tatap muka (*interface*) dibahas mengenai desain tampilan dari aplikasi (*User Interface*).

3.1.1. Perancangan Sistem

3.1.1.1. Flowchart Algoritma Genetika



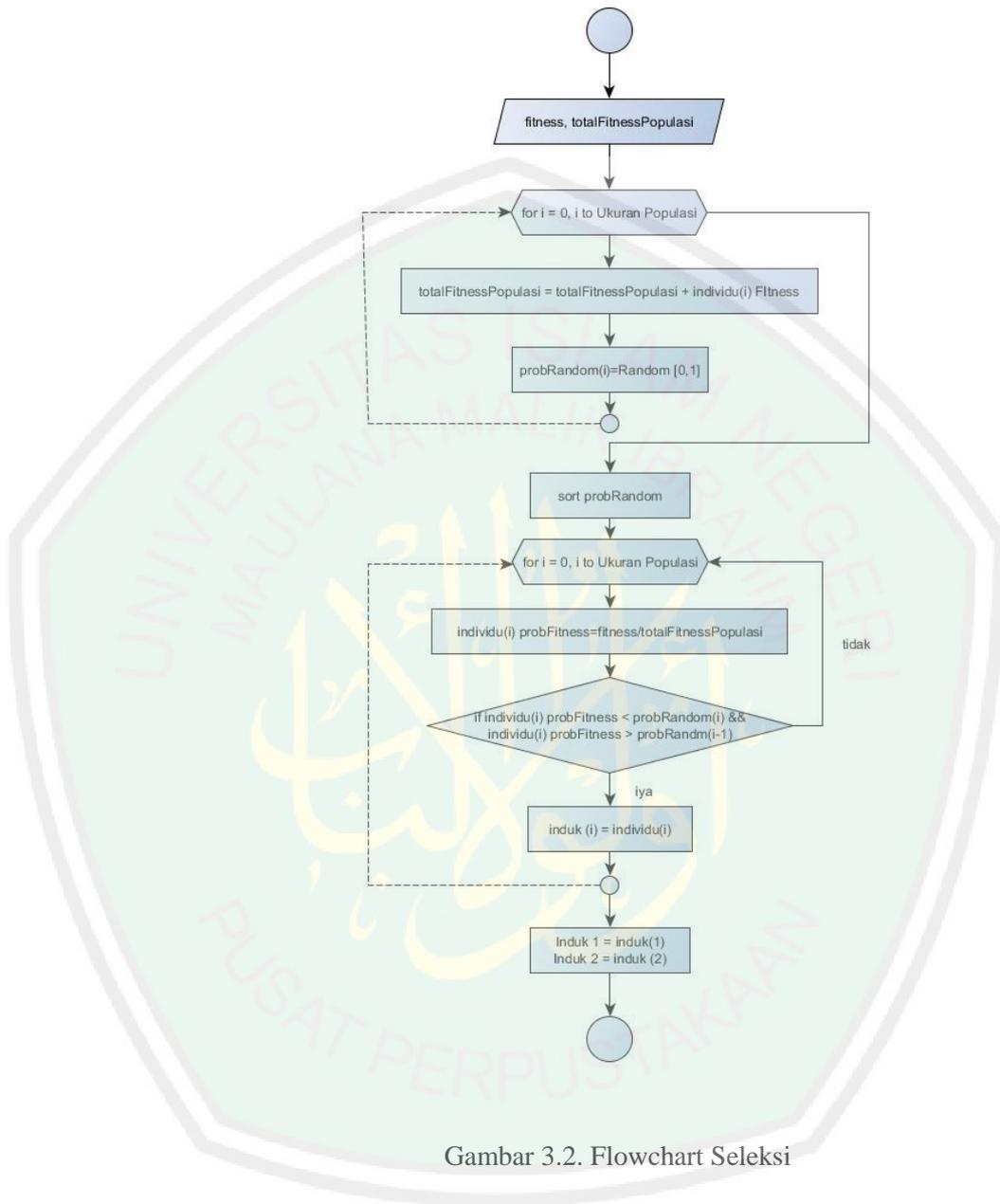
Gambar 3.1. Flowchart Algoritma Genetika

Alur algoritma genetika yang dilakukan dalam dalam penelitian ini adalah

1. Pembangkitan populasi : Pembangkitan populasi awal dilakukan dengan menggunakan pembangkitan acak seperti.
2. Mendefinisikan nilai *fitness*: Yang merupakan ukuran baik-tidaknya sebuah individu atau baik-tidaknya solusi yang didapatkan.
3. Proses Seleksi menggunakan seleksi *Roulette Wheel Selection*.
4. Proses cross-over (perkawinan silang).
5. Proses mutasi gen.

3.1.1.2. Flowchart Seleksi Roulette Wheel

Metode seleksi yang digunakan dalam penelitian ini adalah *Roulette Wheel Selection* yaitu dengan memilih individu tertentu untuk menjadi induk dengan probabilitas sama dengan *fitness* dibagi dengan total *fitness* populasi. Berikut adalah *flowchart diagram* dari metode seleksi *Roulette Wheel* :



Gambar 3.2. Flowchart Seleksi

Seleksi dengan metode Roulette Wheel dilakukan dengan tahapan-tahapan sebagai berikut :

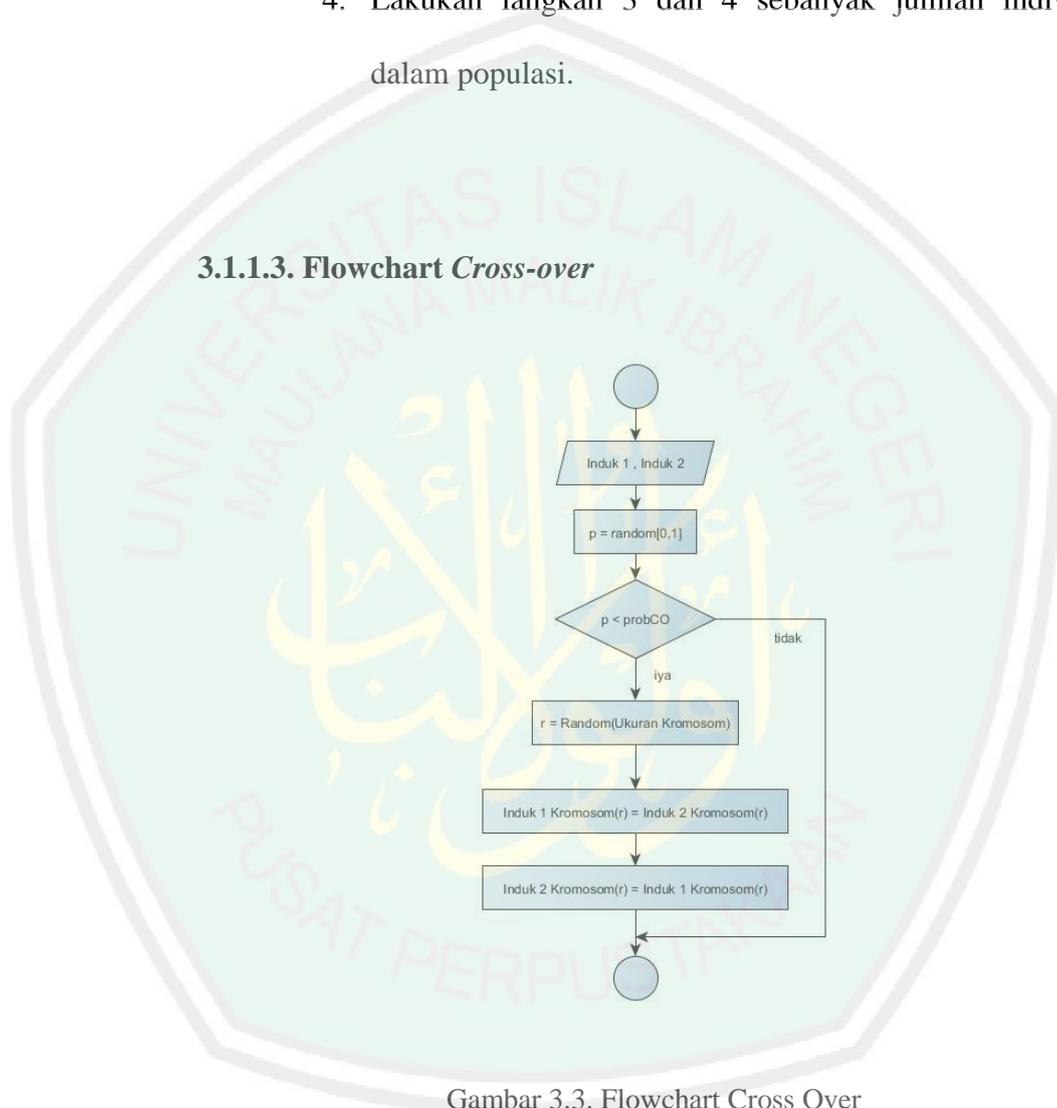
1. Menghitung nilai probabilitas seleksi(p_k) setiap individu

$$\text{dimana } p_k = \frac{f_k}{\sum f_k}$$

2. Mengacak suatu bilangan [0,1].

3. Memilih individu dimana bilangan *random* tersebut berada sebagai induk.
4. Lakukan langkah 3 dan 4 sebanyak jumlah individu dalam populasi.

3.1.1.3. Flowchart *Cross-over*



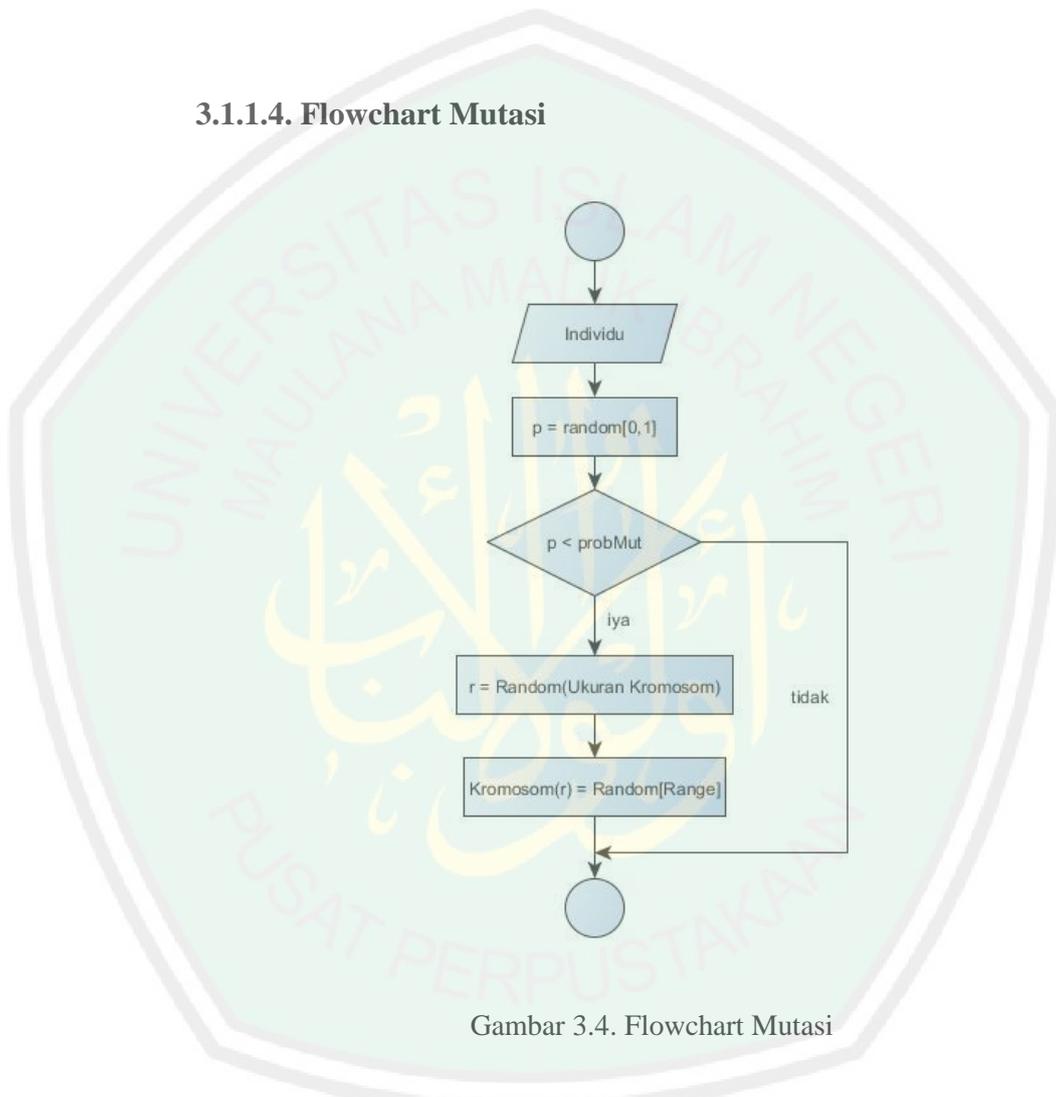
Gambar 3.3. Flowchart Cross Over

Hal-hal penting ketika menggunakan *Cross Over* adalah :

1. *Cross Over* melibatkan dua individu untuk menghasilkan keturunan yang baru.
2. Dilakukan dengan melakukan pertukaran kromosom dari dua individu secara acak.

3. Proses *cross-over* dilakukan pada setiap individu dengan probabilitas *cross-over* yang ditentukan.

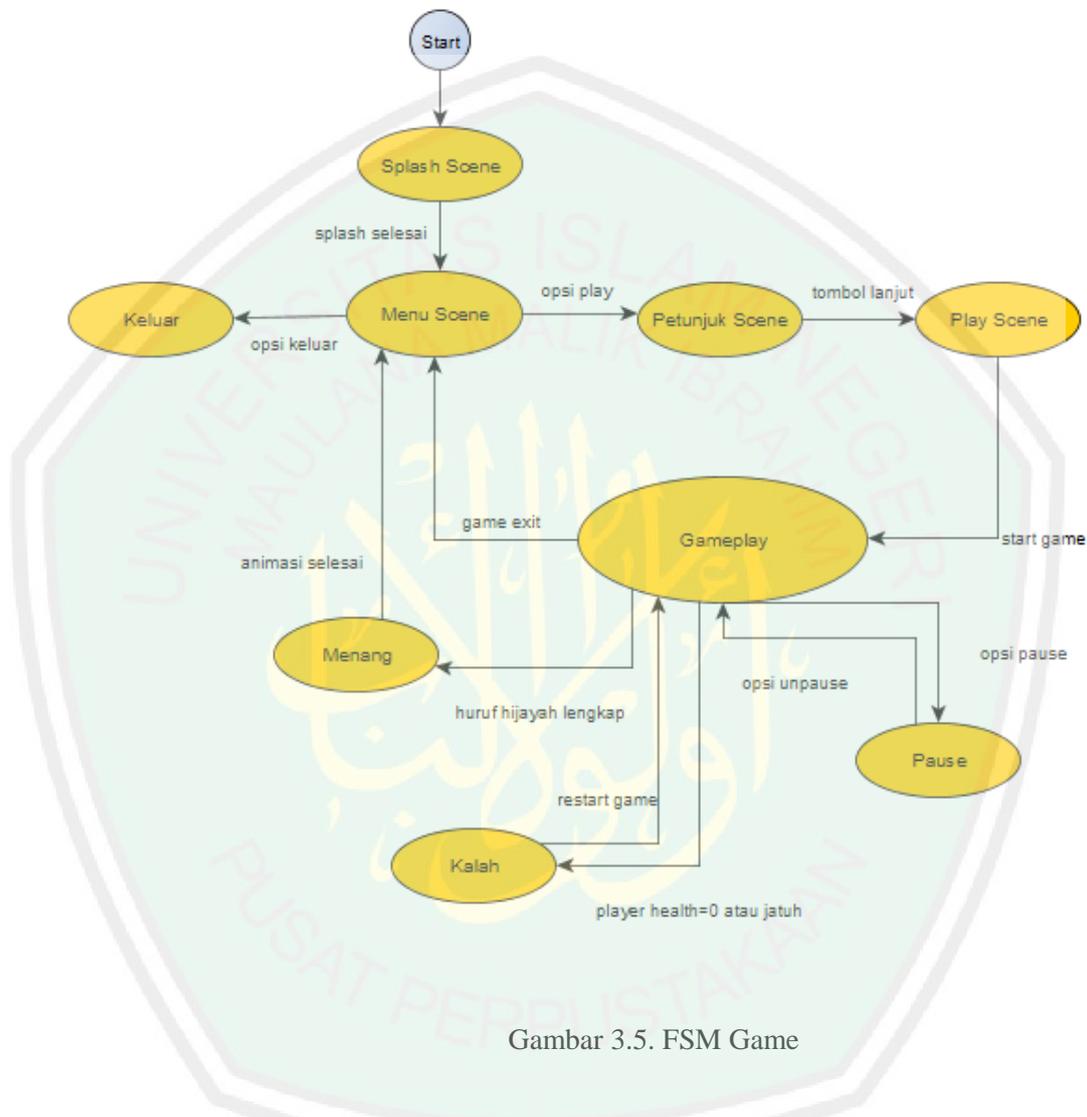
3.1.1.4. Flowchart Mutasi



Gambar 3.4. Flowchart Mutasi

Setiap individu mengalami mutasi gen dengan probabilitas mutasi yang ditentukan secara *random*. Mutasi dilakukan dengan memberikan nilai *random* dengan batas range yang telah ditentukan kepada individu yang terpilih.

3.1.1.5. FSM Game



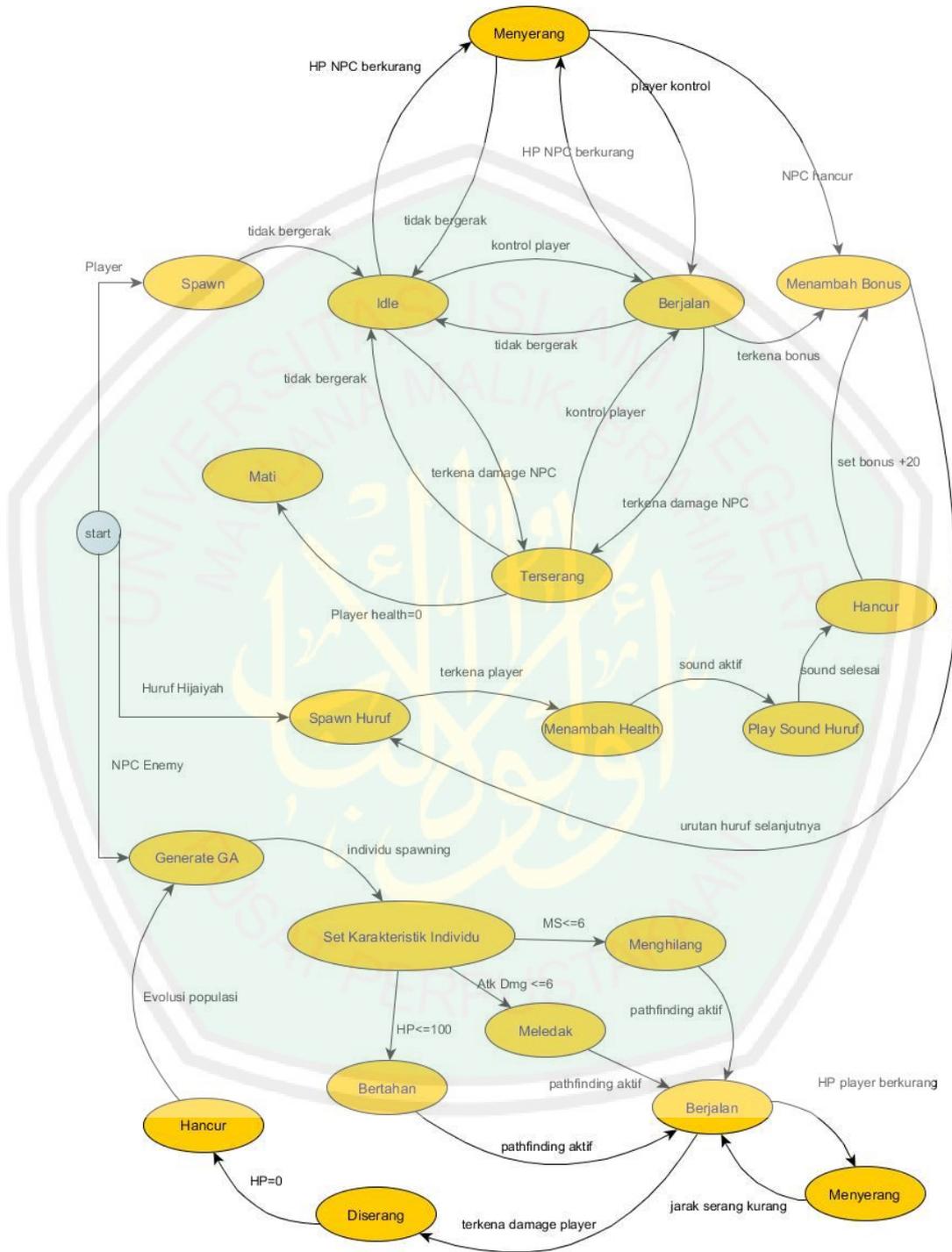
Gambar 3.5. FSM Game

FSM Game dimulai ketika game dijalankan, state awal adalah splash scene, setelah animasi dari splash scene selesai akan berpindah ke *state* Menu scene, pada *state* ini pemain dapat memilih 2 opsi yaitu *play* (bermain), *exit* (keluar), apabila pemain memilih opsi *exit*, game akan segera menghentikan sistem yang bekerja dan menutup aplikasi. Apabila pemain memilih opsi *play*, akan pindah ke *state* petunjuk scene, pada *state* ini pemain diarahkan untuk menjalani misi yang sudah ditentukan. *State*

gameplay dimana permainan sudah berjalan, ada beberapa *state* yang terhubung dengan *state* ini, pertama *state* pause dengan transisi opsi pause dan kembali lagi ke *gameplay* dengan transisi opsi unpauses, kedua *state* kalah dengan transisi jika *player health=0* atau pemain jatuh, *state* kalah kembali ke *gameplay* dengan transisi restart game. Dan yang ketiga adalah *state* menang dimana akan pindah dari *state* ketika huruf hijiyah berhasil dikumpulkan semua atau lengkap. Dari *state* menang akan kembali ke *state* menu scene dengan transisi animasi selesai.

3.1.1.6. FSM Gameplay

FSM Gameplay ini merupakan model dari kondisi-kondisi yang terjadi ketika permainan berlangsung. Memiliki sejumlah *state* yang berhubungan dengan jalannya permainan “Fight For Hijaiyah”. Berikut merupakan gambar dari FSM Gameplay :



Gambar 3.6. FSM Gameplay

FSM Gameplay dibagi menjadi 3 sub FSM yaitu player, huruf hijaiyah dan NPC Enemy. FSM player dimulai ketika karakter pemain spawn, akan berpindah ke state *idle* ketika karakter pemain tidak bergerak, akan berpindah dari *state idle* ke *state* berjalan ketika ada kontrol dari pemain. Akan berpindah ke *state* terserang ketika karakter pemain terkena damage dari NPC enemy, apabila *player health* sudah mencapai 0 maka *state* berpindah ke *state* mati. Dari *state idle* atau berjalan akan berpindah ke *state* menyerang dengan transisi HP NPC enemy berkurang. Dari *state* menyerang akan berpindah ke *state* menambah bonus dengan transisi NPC hancur. Dari *state* berjalan akan berpindah ke *state* menambah bonus dengan transisi terkena bonus. Selanjutnya sub FSM huruf hijaiyah dimulai dari *state* spawn huruf. Dari *state* spawn huruf akan berpindah ke *state* menambah health dengan transisi terkena *player*. Dari *state* menambah health akan pindah ke *state play sound* huruf dengan transisi sound aktif. Dari *state play sound* huruf akan berpindah ke *state* hancur dengan transisi *sound* selesai. Dan akan pindah ke *state* menambah bonus dengan transisi set bonus +20. Yang terakhir adalah sub FSM NPC *enemy* dimulai pada *state generate GA*, pada *state* ini proses pembangkitan populasi dijalankan. State set karakteristik individu adalah *state* ketika individu diberikan nilai *health*, *attack damage*, *attack range*, *attack speed*, dan *movement speed* secara acak. Ada 3 transisi untuk menentukan perilaku individu, pertama pada *state* bertahan ketika *health* ≤ 100 , kedua pada *state* meledak ketika *atk dmg* ≤ 6 , dan yang ketiga pada *state* menghilang ketika *movement speed* ≤ 6 . Dari *state* bertahan, meledak, atau menyerang akan berpindah ke *state* berjalan dengan transisi pathfinding aktif. Dari *state* berjalan akan berpindah

ke *state* menyerang ketika HP *player* berkurang, dan sebaliknya kan berpindah dari *state* menyerang ke *state* berjalan ketika jarak serang kurang. NPC *enemy* pada *state* diserang ketika terkena damage *player*, dan akan ke *state* hancur dengan transisi HP=0. Dari *state* hancur akan pindah ke *state* generate GA dengan transisi evolusi populasi artinya setelah individu hancur akan memunculkan individu yang baru.

3.1.1.7. Analisa Kebutuhan Sistem

1. Kebutuhan Hardware

Spesifikasi hardware yang digunakan untuk penelitian adalah :

- a. AMD Quad-Core A6
- b. RAM 4GB DDR3
- c. AMD Radeon HD 8400

2. Kebutuhan Software

Software pendukung yang dibutuhkan untuk penelitian

adalah :

- a. OS Windows 8.1 64bit (*Trial Version*)
- b. Unity 3D versi 5 (*Free Version*)
- c. Blender (*Free Version*)
- d. Mock Up (*Trial Version*)
- e. Photoshop (*Trial Version*)

3.1.2. Perancangan Game

3.1.2.1 Informasi umum

Game yang dirancang ini memiliki judul “*Fight For Hijaiyah*”. Jenis game yang dirancang dan dikembangkan ini adalah game edukasi yang hanya dapat dimainkan oleh satu orang atau single pemain. Game ini dirancang dan dibuat pada platform desktop. Judul “*Fight For Hijaiyah*” diambil karena sesuai dengan jenis game yang dibuat yaitu game pengenalan huruf hijaiyah, yang cara bermainnya adalah pemain berperan sebagai seseorang anak laki-laki yang akan menjalani beberapa misi, tentunya dengan misi-misi yang sudah dipersiapkan untuk menambah pengetahuannya tentang huruf hijaiyah. Misi utama dari game ini adalah mengumpulkan huruf hijaiyah dengan berbagai tantangan yang menarik. Game ini berbasis *health*, dimana user mengumpulkan huruf hijaiyah dengan *health point* yang tidak sampai habis. Fitur dari game ini adalah game ini akan menambahkan vokal pada setiap huruf hijaiyah. Terdapat keterangan mahraj pada setiap huruf hijaiyah.

3.1.2.2. Objektif game

Mengajak pemain untuk menjadi tokoh utama dalam game dan mengikuti misi yang ada didalam game. Untuk menyelesaikan misi didalam game tersebut, pemain harus menghadapi tantangan yang

diberikan dan juga pemain diajak untuk belajar bersama tentang huruf hijaiyah.

3.1.2.3. Target Game

Target dalam game ini adalah untuk memberikan suasana baru dalam pembelajaran. Mengajak pemain bermain sambil belajar dan juga memberikan suatu cerita singkat agar game lebih menarik dan memiliki storyline. Dengan memakai strategi mengajar yang awalnya mudah dan semakin lama semakin sulit, diharapkan pemain dapat berlatih melalui aplikasi ini.

3.1.2.4. Storyline

Game ini menceritakan tentang seorang anak laki-laki dengan latar belakang ksatria yang akan mengambil kitab ajaib. Kitab ajaib tersegel sampai anak itu mendapatkan seluruh huruf hijaiyah untuk membuka segelnya. Sembari mengumpulkan huruf hijaiyah anak laki-laki itu harus mampu bertahan dari berbagai musuh yang menghadang.

3.1.2.5. Konsep

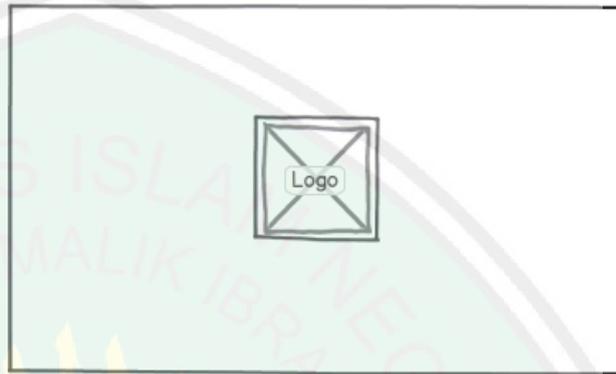
Pada saat pemain menjalankan launcher game, tampilan pertama yang muncul adalah intro (*Splash Scene*). Pemain akan memilih beberapa opsi di tampilan menu, memilih *play* untuk bermain dan *exit* untuk keluar dari game. Apabila pemain memilih opsi *play* maka pemain langsung di arahkan ke tampilan petunjuk permainan. Pada tampilan petunjuk permainan pemain diberikan misi-misi yang harus diselesaikan.

Dalam *gameplay* terdapat *score*, *healthbar*, *pause*, *quit*, kotak huruf, arena, pemain. Pemain harus melakukan hal sesuai misi yang diberikan, NPC musuh ditambahkan kecerdasan buatan dengan metode algoritma genetika. Apabila pemain kalah dalam game maka akan muncul tampilan *end game* dan kembali ke menu utama.

3.1.2.6. Storyboard

Berikut adalah *storyboard* yang akan dibuat dalam penelitian ini adalah :

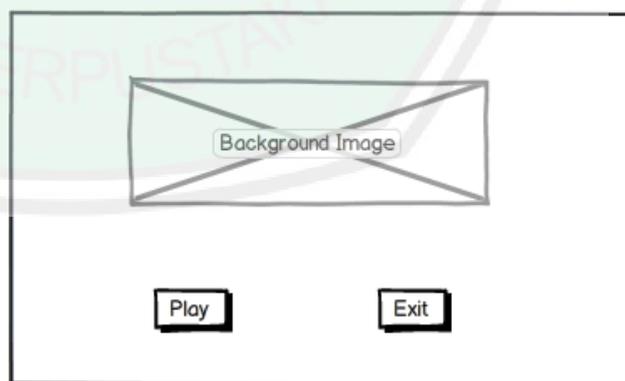
1. Halaman Intro



Gambar 3.7. Desain Splash Scene Game

Animasi splash scene berjalan setelah launcher dieksekusi, dan akan menuju ke tampilan menu setelah animasi berakhir.

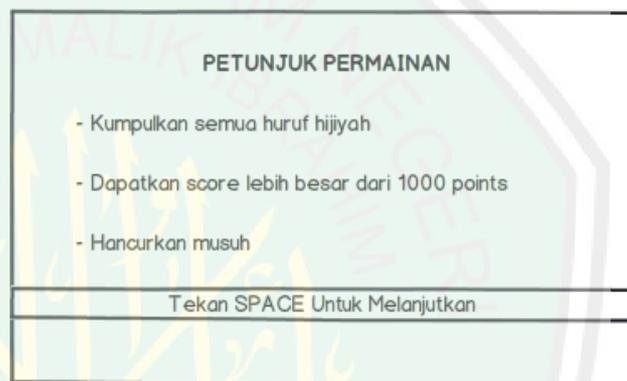
2. Halaman Menu



Gambar 3.8. Desain Menu Utama Game

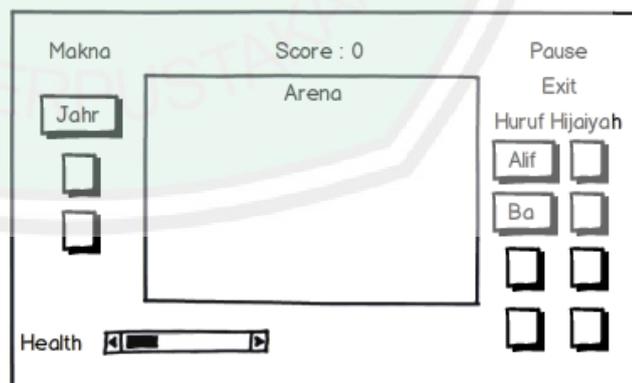
Halaman menu adalah halaman dimana pemain akan memilih untuk bermain atau keluar, jika i-ngin melanjutkan permainan klik *button play* dan jika ingin keluar klik *button exit*.

3. Halaman Petunjuk Permainan



Gambar 3.9. Halaman Petunjuk Permainan

4. Halaman Gameplay



Gambar 3.10. Halaman Gameplay

Halaman *gameplay* dimana pemain akan berada di halaman ini ketika permainan

berlangsung. Terdapat *health*, *score*, *pause*, *exit*, box hijau, dan makna kata.

5. Halaman Game Menang



Gambar 3.11. Halaman Pemain Menang

Halaman yang akan tampil ketika permainan selesai dan player memenangkannya.

6. Halaman Game Kalah

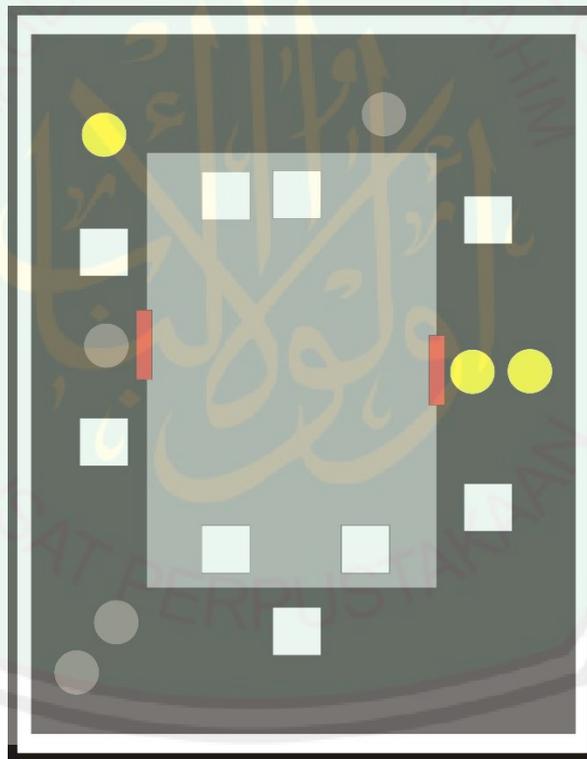


Gambar 3.12. Halaman Pemain Kalah

Halaman yang akan tampil ketika pemain jatuh atau mati.

3.1.2.7. Desain *Environment*

Environment di desain dengan bentuk arena dengan beberapa properti didalamnya. Berikut desain *environment* :



Gambar 3.13. Desain Environment

Deskripsi *environment* :

1. Warna Hitam : Jurang(Pemain yang jatuh akan mati)
2. Warna Putih : Patung bidadari

3. Warna Abu-abu : Lantai
4. Warna Kuning : Level atas
5. Warna Abu-Abu Kehitaman : Level bawah
6. Warna Merah : Segel Api

3.1.2.8. Desain Karakter

Permainan ini terdapat 2 karakter utama, karakter pertama adalah pemain, dan karakter lainnya adalah NPC musuh.

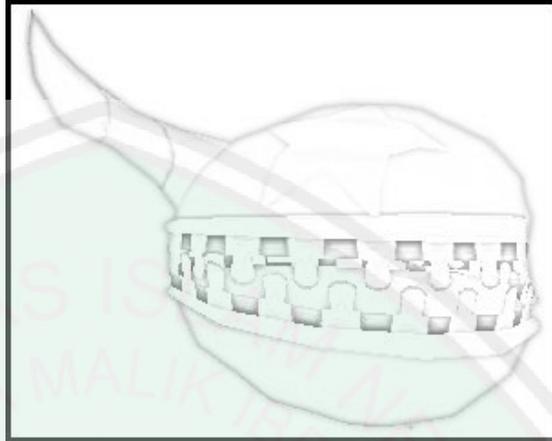
1. Karakter Player



Gambar 3.14. Sketsa Karakter Player

Pemain digambarkan seorang anak-anak yang memakai baju ksatria. Makna dari desain tersebut bahwasanya seorang anak harus memiliki keberanian dalam setiap tindakan.

2. Karakter NPC Musuh



Gambar 3.15. Sketsa Karakter NPC Musuh

Desain sketsa NPC Musuh digambarkan dengan monster berbentuk bulat, mempunyai gigi besar dan bertanduk setan.

BAB IV

HASIL DAN PEMBAHASAN

Pada bab ini menjelaskan mengenai implemementasi game yang telah dirancang sebelumnya. Impelemenasi mencakup desain sistem, desain antarmuka, dan implementasi algoritma genetika pada NPC. Setelah semua implementasi dilakukan selanjutnya menjalankan evaluasi dan uji coba. Uji Coba dan Evaluasi dilakukan dengan menganalisis hasil dari proses algoritma genetika.

4.1. Implementasi Algoritma Genetika pada NPC

Dalam pengembangan aplikasi, algoritma genetika akan diterapkan pada perilaku NPC. Berikut adalah langkah-langkah implementasi Algoritma Genetika:

Langkah pertama yang harus kita lakukan adalah pembentukan individu. Pembentukan individu dimulai dengan deklarasi kromosom yang dibutuhkan. Nilai dari kromosom adalah angka acak dalam rentang tertentu. Dari nilai-nilai kromosom yang acak ditentukan perilaku yang mungkin untuk memaksimalkan potensi tiap individu.

Tabel 4.1. Kromosom

Kromosom
Health Point
Attack Damage
Attack Speed
Attack Range
Movement Speed

Tabel 4.2. Perilaku NPC

Perilaku NPC
NPC Bertahan
NPC Menghilang
NPC Meledak

Setelah ditentukan kromosom dan perilaku NPC yang mungkin maka langkah selanjutnya adalah mengatur gen-gen dari tiap kromosom secara *random*. Peneliti tidak mengatur setiap individu menggunakan satu set konstanta yang telah ditetapkan agar populasi menjadi beragam sebisa mungkin. Sebuah populasi yang tidak beragam tidak akan efektif dalam menemukan solusi terbaik. Cara terbaik untuk membuat populasi yang beragam adalah untuk menetapkan perilaku secara acak.

Tabel 4.3. Individu dan Kromosom

Individu	Kromosom				
	A	Health Point = 75 - 150	Attak Damage = 5 - 10	Attack Speed = 1 - 3	Attack Range = 1 - 3
B	Health Point = 75 - 150	Attak Damage = 5 - 10	Attack Speed = 1 - 3	Attack Range = 1 - 3	Move Speed = 4 - 8
C	Health Point = 75 - 150	Attak Damage = 5 - 10	Attack Speed = 1 - 3	Attack Range = 1 - 3	Move Speed = 4 - 8
.....
Z	Health Point = 75 - 150	Attak Damage = 5 - 110	Attack Speed = 1 - 3	Attack Range = 1 - 3	Move Speed = 4 - 8

Tabel 4.4. Perilaku dan Kromosom

Perilaku	Kromosom
NPC Bertahan	Health Point \leq 100
NPC Menghilang	Move Speed \leq 5
NPC Meledak	Attack Damage \leq 6

Setiap kromosom diberikan nilai secara acak dengan ketentuan masing-masing. Perilaku ditentukan dari kelemahan-kelemahan tiap individu, sehingga akan bertindak lebih agresif kepada pemain. Sebagai contoh menghilang adalah perilaku terbaik ketika individu lemah di kecepatan berjalan.

Langkah selanjutnya adalah penentuan nilai *fitness* dari tiap individu. Yang dicari saat ini adalah anggota populasi yang merupakan tantangan terbesar bagi para pemain. Perlu beberapa cara untuk mengukur tingkat tantangan. Dapat mempertimbangkan beberapa pendekatan yang berbeda. Game pengenalah huruf

hijayah menggunakan health pada pemain. *Health* berkurang apabila pemain terkena damage dari serangan individu. Pemain kalah setelah point hit mencapai nol. Jadi, salah satu cara untuk mengukur tingkat tantangan adalah untuk menjaga total kumulatif jumlah kerusakan *health-point* dilakukan untuk pemain. Indikator lain yang bisa digunakan untuk menghitung nilai *fitness* adalah point. Jika point yang didapatkan pemain sedikit dalam waktu tertentu maka individu saat itu adalah individu terbaik.

$$\text{fitness} = \frac{\text{total damage done}}{\text{lifetime}}$$

Total Damage Done : total damage yang telah diberikan oleh NPC kepada pemain-

Life Time : jumlah waktu NPC dari kemunculan sampai ke generasi berikutnya.

Setelah menentukan rumus fungsi *fitness*, selanjutnya adalah menentukan nilai maksimum *fitness*. Maksimum *fitness* digunakan sebagai ukuran nilai *fitness* yang ingin dicapai. Solusi dikatakan optimal ketika angka *fitness* sama dengan nilai maksimum *fitness*.

Langkah selanjutnya adalah proses pembentukan populasi, inilah individu-individu dibangkitkan dalam artian semua kromosom diatur atau diberi nilai acak dan selanjutnya disimpan pada array populasi. Langkah berikutnya adalah mengeksekusi *fitness* populasi pada generasi pertama. Perhitungan nilai *fitness* setelah totalDamageDone dan Lifetime didapatkan dan diakumulasikan.

Berikut merupakan contoh penyelesaian kasus dengan menggunakan metode algoritma genetika :

Misalkan ada 4 individu dalam populasi saat ini dengan kromosom dan perilaku yang telah diketahui seperti terlihat pada tabel dibawah, dan ingin diketahui populasi baru yang paling optimal.

Tabel 4.5. Individu Pertama

KROMOSOM		PERILAKU		
HP	139		TOTAL DAMAGE	40
ATKDAMAGE	5	MELEDAK	LIFETIME	19
ATKSPEED	2			
ATKRANGE	3			
MOVESPEED	8			

Tabel 4.6. Individu Kedua

KROMOSOM		PERILAKU		
HP	87	BERTAHAN	TOTAL DAMAGE	39
ATKDAMAGE	7		LIFETIME	26
ATKSPEED	2			
ATKRANGE	1			
MOVESPEED	5	MENGHILANG		

Tabel 4.7. Individu Ketiga

KROMOSOM		PERILAKU		
HP	105		TOTAL DAMAGE	16
ATKDAMAGE	6	MELEDAK	LIFETIME	7
ATKSPEED	3			
ATKRANGE	1			
MOVESPEED	7			

Tabel 4.8. Individu Keempat

KROMOSOM		PERILAKU	
HP	120	TOTAL DAMAGE	20
ATKDAMAGE	8	LIFETIME	14
ATKSPEED	3		
ATKRANGE	1		
MOVESPEED	6		

Langkah selanjutnya menghitung nilai fitness dari masing-masing individu.

$$Fitness = \frac{totalDamageDone}{lifetime}$$

Individu 1 $Fitness = \frac{40}{19} = 2.105263$

Individu 2 $Fitness = \frac{39}{26} = 1.5$

Individu 3 $Fitness = \frac{16}{7} = 2.285714$

Individu 4 $Fitness = \frac{20}{14} = 1.428571$

Dari perhitungan diatas dapat dibuat tabel seperti dibawah ini :

Tabel 4.9. Individu Fitness

Individu ke-	1	2	3	4
Fitness	2.105263	1.5	2.285714	1.428571

Pada langkah seleksi dipilih individu yang memiliki sifat dan perilaku terbaik untuk ditanamkan pada generasi berikutnya. Dalam proses seleksi biasanya disebut fungsi *fitness* untuk mengidentifikasi individu yang digunakan

untuk membuat generasi berikutnya. Pada proses seleksi, individu dalam populasi yang memiliki *fitness* tertinggi berkemungkinan besar untuk menjadi inputan proses *cross-over*.

Langkah-langkah penyelesaian seleksi dengan metode *Roulette Wheel* sebagai berikut:

1. Hitung total nilai Fitness ($\sum f_k$)

$$\text{Total nilai Fitness } (\sum f_k) = 2.105263 + 1.5 + 2.285714 + 1.428571 = 7.31954$$

2. Hitung probabilitas seleksi (p_k)

$$p_k = \frac{f_k}{\sum f_k}$$

$$\text{Individu 1} \quad p_k = \frac{2.105263}{7.31954} = 0.287622$$

$$\text{Individu 2} \quad p_k = \frac{1.5}{7.31954} = 0.204930$$

$$\text{Individu 3} \quad p_k = \frac{2.285714}{7.31954} = 0.31227$$

$$\text{Individu 4} \quad p_k = \frac{1.428571}{7.31954} = 0.195172$$

Lakukan perhitungan yang sama untuk seluruh individu sehingga didapat nilai probabilitas seleksi untuk masing-masing individu seperti pada tabel berikut:

Tabel 4.10. Probabilitas Seleksi

Individu ke-	1	2	3	4
Fitness	2.105263	1.5	2.285714	1.428571
Probabilitas	0.287622	0.204930	0.31227	0.195172

3. Bangkitkan bilangan secara acak [0,1]

Misal bilangan acak yang dibangkitkan untuk masing-masing iterasi sebagai berikut:

iterasi/trial ke-1 : 0.72

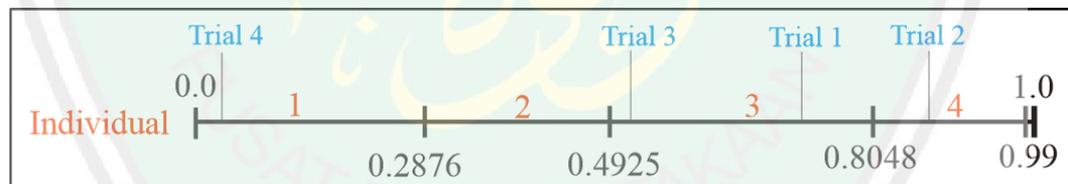
iterasi/trial ke-2 : 0.89

iterasi/trial ke-3 : 0.53

iterasi/trial ke-4 : 0.04

4. Pilih individu dimana bilangan acak tersebut berada sebagai induk.

Hasil pemilihan individu dari bilangan acak yang dibangkitkan untuk setiap iterasi adalah seperti terlihat pada gambar.



Gambar 4.1. Rentang *Roulette Wheel*

Keterangan gambar : Batas per individu berdasarkan rentangan nilai probabilitas individu.

Dari gambar dapat dilihat bahwa pada trial pertama, dibangkitkan bilangan *random* 0.72, bilangan tersebut berada pada rentangan individu ke 3, maka individu tersebut dipilih sebagai induk pertama, begitu juga selanjutnya..

Cross-Over (Perkawinan Silang) mengkombinasikan dua individu untuk memperoleh individu-individu baru yang diharapkan mempunyai *fitness* lebih baik. Prosesnya menyalin isi kromosom individu I ke dalam kromosom *offsprings* 1 dan menyalin kromosom individu II ke dalam kromosom *offsprings* 2, selanjutnya menurunkan isi dari kedua kromosom tersebut. Banyaknya pasangan induk yang mengalami *cross-over* ditentukan dengan nilai probabilitas *cross-over*. Pada proses *cross-over* dipilih inputan yaitu 2 induk dari hasil seleksi. Kromosom yang ditukar sesuai dengan syarat probabilitas rate *cross-over*.

Langkah-langkah penyelesaian *cross-over* sebagai berikut:

1. Identifikasi induk yang akan dijadikan inputan *cross-over*

Tabel 4.11. Cross Over Induk I dan II

Individu	Kromosom				
	3	HP=105	ATKDMG=6	ATKSPEED=3	ATKRANGE=1
4	HP=120	ATKDMG=8	ATKSPEED=3	ATKRANGE=1	MOVESPEED=6

2. Bangkitkan bilangan secara acak [0,1] untuk menentukan kromosom yang akan digunakan sebagai pertukaran nilai.

Misal bilangan acak yang dibangkitkan sebagai berikut:

Kromosom 1 : 0.32

Kromosom 2 : 0.43

Kromosom 3 : 0.86

Kromosom 4 : 0.09

Kromosom 5 : 0.92

3. Membandingkan bilangan acak dengan probabilitas *cross over rate*.

Misalkan probabilitas *cross over* ditentukan dengan nilai 0.5, maka bilangan acak yang kurang atau sama dengan 0.5 akan dijadikan untuk pertukaran nilai kromosom/

Kromosom 1 : $0.32 \leq 0.5$? (Ya)

Kromosom 2 : $0.43 \leq 0.5$? (Ya)

Kromosom 3 : $0.86 \leq 0.5$? (Tidak)

Kromosom 4 : $0.09 \leq 0.5$? (Ya)

Kromosom 5 : $0.92 \leq 0.5$? (Tidak)

4. Melakukan perkawinan silang sesuai dengan kromosom yang dipilih

Hasil *cross over* dapat dilihat pada tabel dibawah ini :

-

Tabel 4.12 Hasil Cross Over

Individu	Kromosom				
	Offspring1	HP=120	ATKDMG=8	ATKSPEED=2	ATKRANGE=1
Offspring2	HP=105	ATKDMG=6	ATKSPEED=2	ATKRANGE=3	MOVESPEED=5

Selanjutnya proses mutasi dimana akan menetapkan kembali kromosom secara acak. Langkah-langkah penyelesaian mutasi sebagai berikut:

1. Identifikasi individu

Tabel 4.13. Identifikasi Mutasi

Individu	Kromosom				
	Offspring1	HP=120	ATKDMG=8	ATKSPEED=2	ATKRANGE=1

2. Bangkitkan bilangan secara acak $[0,1]$ untuk menentukan kromosom yang akan dimutasi.

Misal bilangan acak yang dibangkitkan sebagai berikut:

Tabel 4.14. Pembangkitan Bilangan Acak

Individu	Bilangan				
	1	0.83	0.12	0.22	0.45
2	0.15	0.87	0.99	0.11	0.1
3	0.97	0.01	0.42	0.51	0.66
4	0.88	0.2	0.1	0.22	0.71

3. Membandingkan bilangan acak dengan probabilitas mutasi *rate*.

Misalkan probabilitas *cross over* ditentukan dengan nilai 0.3 maka bilangan acak yang kurang atau sama dengan 0.3 akan dimutasi.

Tabel 4.15. Perbandingan Probabilitas Mutasi

Individu	Bilangan				
Offspring	0.83	0.12	0.22	0.45	0.12
Bilangan \leq prob Mutasi rate?	Tidak	Ya	Ya	Tidak	Ya
Offspring	0.15	0.87	0.99	0.11	0.1
Bilangan \leq prob Mutasi rate?	Ya	Tidak	Tidak	Ya	Ya
Offspring	0.97	0.01	0.42	0.51	0.66
Bilangan \leq prob Mutasi rate?	Tidak	Ya	Tidak	Tidak	Tidak
Offspring	0.88	0.2	0.90	0.22	0.71
Bilangan \leq prob Mutasi rate?	Tidak	Ya	Tidak	Ya	Tidak

4. Melakukan mutasi sesuai dengan kromosom yang dipilih.

Tabel 4.16. Pengacakan Kembali Kromosom

Ind	Kromosom				
Offspring	HP=120	ATKDMG= Random(5,10)	ATKSPEED= Ra6dom(1,3)	ATKRANGE=1	MOVESPEED= 8
Offspring	HP= Random(75,150)	ATKDMG=8	ATKSPEED=2	ATKRANGE= Random(1,3)	MOVESPEED= Random(4,8)
Offspring	HP=120	ATKDMG= Random(5,10)	ATKSPEED=2	ATKRANGE=1	MOVESPEED= 8
Offspring	HP=120	ATKDMG= Random(5,10)	ATKSPEED=2	ATKRANGE= Random(1,3)	MOVESPEED= 8

Kromosom yang terpilih akan diacak kembali sehingga akan mendapatkan hasil seperti tabel dibawah ini :

Tabel 4.17. Hasil Mutasi

Individu	Kromosom				
	Offspring	HP=120	ATKDMG=9	ATKSPEED=1	ATKRANGE=1
Offspring	HP=145	ATKDMG=8	ATKSPEED=2	ATKRANGE=2	MOVESPEED=5
Offspring	HP=120	ATKDMG=5	ATKSPEED=2	ATKRANGE=1	MOVESPEED=8
Offspring	HP=120	ATKDMG=6	ATKSPEED=2	ATKRANGE=3	MOVESPEED=8

Pada proses mutasi diatas, kromosom dibentuk ulang atau diperbarui lagi diharapkan mendapatkan hasil yang lebih baik dari sebelumnya. Terkadang hasil mutasi bisa juga lebih buruk tergantung dengan probabilitas rate mutasi

Setelah proses *cross-over* dan mutasi selesai, hasil evolusi karakteristik dan perilaku ke populasi baru sudah dapat dilihat pada tabel berikut :

Tabel 4.18. Populasi Baru

Individu	Kromosom				
	1	HP=120	ATKDMG=9	ATKSPEED=1	ATKRANGE=1
Perilaku					
2	HP=145	ATKDMG=8	ATKSPEED=2	ATKRANGE=2	MOVESPEED=5
Perilaku					Menghilang
3	HP=120	ATKDMG=5	ATKSPEED=2	ATKRANGE=1	MOVESPEED=8
Perilaku		Meledak			
4	HP=120	ATKDMG=6	ATKSPEED=2	ATKRANGE=3	MOVESPEED=8
Perilaku		Meledak			

Tabel diatas adalah hasil dari evolusi yaitu terbentuknya populasi baru. Untuk individu 1 tidak terdapat perilaku karena beberapa kromosom tidak memenuhi syarat untuk disisipkannya perilaku artinya individu 1 memiliki nilai kromosom yang baik. Proses evolusi populasi akan terus-menerus dilakukan sampai mendapatkan solusi optimal atau nilai *fitness* sama dengan nilai maks *fitness*. Misalkan maks fitness adalah 10, maka perulangan akan berakhir ketika nilai *fitness* mencapai 10. Sebagai contoh adalah tabel berikut :

Tabel 4.19. Evolusi Populasi

Generasi	Fittest
1	2.28
2	2.28
3	2.48
4	3.45
5	4.21
6	4.21
7	5.24
8	6.65
9	7.01
10	9.98
11	10

Dari tabel diatas dapat dilihat bahwasanya mendapatkan solusi optimal pada generasi ke-11, maka perulangan dihentikan pada saat itu.

4.2. Implementasi Permainan

4.2.1. Model dan Karakter dalam Permainan

4.2.1.1. Environment Game



Gambar 4.2. Environment Game

Environment adalah lingkungan saat permainan berlangsung, ada beberapa misi yang harus dilakukan pemain untuk membuka beberapa segel *environment*. Properti bidadari digunakan sebagai penghias dan sebagai tempat perlindungan pemain agar tidak jatuh saat bertarung dengan NPC musuh.

4.2.1.2. Model Karakter

Ada 2 karakter dalam game ini yaitu karakter pemain dan karakter NPC musuh, model karakter dapat dilihat pada gambar dibawah ini :



Gambar 4.3. Model Karakter Pemain

Karakter pemain dimodelkan seorang anak-anak yang memakai baju ksatria. Karakter memegang sebuah pedang berwarna merah yang digunakan untuk menghancurkan NPC musuh.



Gambar 4.4. Model Karakter NPC Musuh

Dalam game terdapat 3 musuh, yaitu NPC musuh kelas A/*Mini Boss*(warna hitam) memiliki health point dan serangan yang tinggi, NPC musuh kelas B(warna hijau) memiliki health point dan serangan yang

rendah namun berjumlah banyak dan memiliki perilaku yang unik pada tiap individu, dan NPC musuh kelas C(warna kuning) memiliki serangan yang sangat rendah namun memiliki kecepatan dan jarak serang yang tinggi.

4.2.1.3. Model Huruf Hijaiyah



Gambar 4.5. Model Huruf Hijaiyah 3D

Model huruf hijaiyah dibuat 3 dimensi agar memudahkan player dalam mempelajari dan mengerti bentuk-bentuk huruf dengan rinci.

4.2.2. Antarmuka dalam Permainan

4.2.2.1. Splash Scene



Gambar 4.6. Splash Scene Game

Splash scene dibuat dengan animasi tulisan dan berjalan selama beberapa detik, setelah selesai lanjut ke tampilan Menu.

4.2.2.2. Menu Scene



Gambar 4.7. Menu Scene Game

Tampilan menu scene terdapat dua opsi, *Start* untuk memulai permainan dan *Exit* untuk keluar dari permainan.

4.2.2.3. Petunjuk Permainan Scene



Gambar 4.8. Petunjuk Permainan

Tampilan ini memberi informasi kepada pemain misi-misi apa yang harus diselesaikan selama permainan berjalan.

4.2.2.4. Gameplay Scene



Gambar 4.9. Gameplay Scene

Tampilan saat permainan dimulai, terlihat player dan NPC musuh yang akan bertarung.

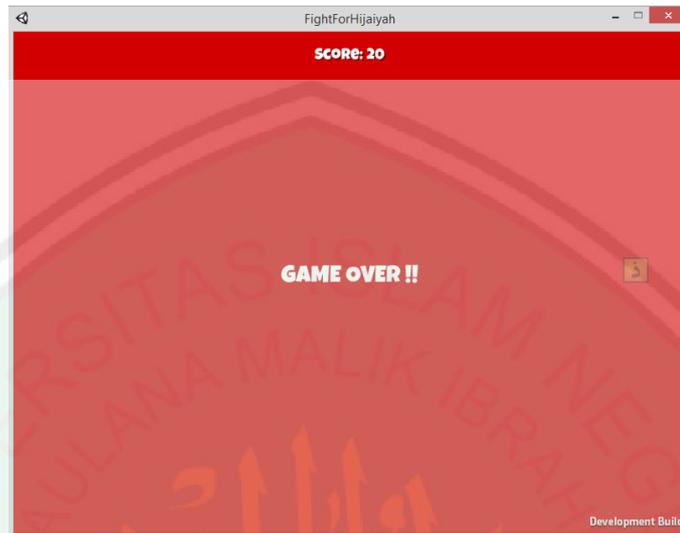
4.2.2.5. Menang Scene



Gambar 4.10. Scene Game Menang

Tampilan scene ketika pemain berhasil menyelesaikan semua misi-misi sesuai dengan ketentuan dalam permainan.

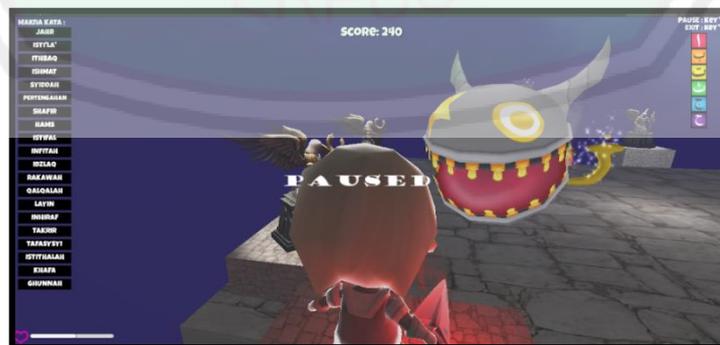
4.2.2.6. Kalah Scene



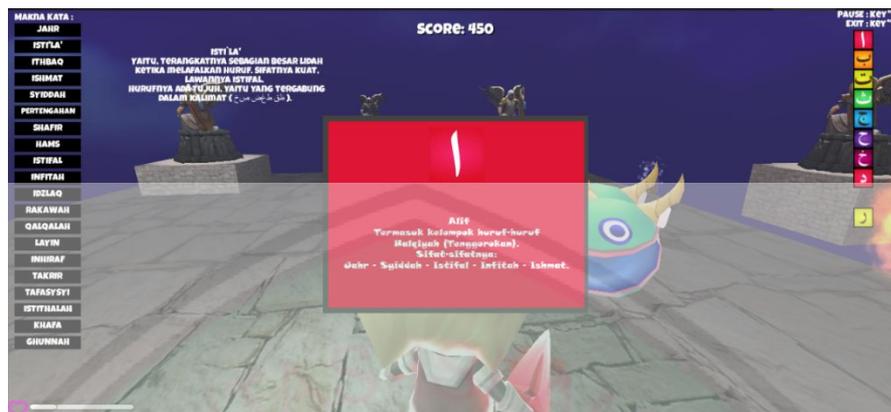
Gambar 4.11. Scene Game Kalah

Tampilan scene ketika pemain jatuh, mati atau kalah saat bertarung dengan NPC musuh.

4.2.3. Gameplay Permainan



Gambar 4.12. Gameplay Pemain Menghadapi Mini Boss



Gambar 4.13. Tampilan Mahraj dan Keterangan Huruf

Dalam *gameplay* terdapat *score*, *healthbar*, *pause*, *quit*, kotak huruf, arena, pemain. Ketika permainan dimulai pemain dan NPC musuh kelas B di spawn secara bersamaan. Algoritma Genetika mulai bekerja pada NPC musuh ketika spawn pertama kali. Selanjutnya musuh akan melakukan perubahan-perubahan perilaku dan karakteristik sesuai dengan generasinya. Pemain harus menghancurkan NPC atau mengambil huruf hijaiyah musuh untuk mendapatkan *score*. Huruf Hijaiyah di spawn berurutan dari alif sampai ya. Selain mendapatkan *score* mengambil huruf hijaiyah akan menambah *health* pemain sehingga mampu bertahan lebih lama dari NPC musuh dan juga dapat membuka kotak huruf hijaiyah yang masih tersegel, fungsi ketika huruf hijaiyah terbuka adalah agar pemain dapat melihat keterangan bacaan huruf, vokal dan mahrajnya. Untuk memenangkan game pemain harus melakukan hal sesuai misi yang diberikan. Apabila pemain kalah dalam game makan akan muncul tampilan *game over scene* dan kembali ke menu utama.

4.2.4. Hasil Uji Coba Algoritma Genetika

Tabel 4.20. Hasil Uji Coba Algoritma

Pop	Idv	Kromosom					Fit	Perilaku			Fittest
		1	2	3	4	5		1	2	3	
1	1	150	7	2.98	2.12	4.71	1.25			Menghilang	2.41
	2	92	9	2.98	3.28	6.55	2.41				
	3	115	5	2.36	2.69	7.62	0.62	Meledak			
	4	79	9	1.63	3.25	5.10	1.57		Bertahan		
2	1	150	9	2.98	2.12	6.01	2.12				2.24
	2	100	9	1.5	2.12	6.55	1.23		Bertahan		
	3	72	9	2.98	1.02	7.28	0.28		Bertahan		
	4	150	9	2	2.12	6.55	2.24				
3	1	72	9	2.98	1.02	6.01	2.41	Meledak	Bertahan		3.65
	2	102	9	1.21	2.31	5.03	3.65				
	3	80	5	2.98	1.02	6.01	3.01	Meledak			
	4	72	9	2.98	1.02	7.31	1.23	Meledak			
4	1	80	5	1.21	1.02	5.03	3.65	Meledak	Bertahan		3.65
	2	139	5	1.34	1.02	7.11	1.91	Meledak			
	3	120	7	1.21	1.02	5.03	2.23				
	4	80	5	1.21	2.56	6.79	2.44	Meledak	Bertahan		
5	1	139	5	1.21	1.02	5.03	4.21	Meledak			4.21
	2	102	5	2.82	1.02	4.59	3.22	Meledak		Menghilang	
	3	139	8	2.72	1.02	5.03	3.54				
	4	139	5	1.21	1.02	6.21	2.98	Meledak			
6	1	76	8	2.13	1.44	7.02	4.21		Bertahan		4.43
	2	139	6	1.21	1.02	5.03	3.67	Meledak			
	3	150	7	1.06	2.21	4.34	4.43			Menghilang	

	4	144	5	1.21	1.02	4.00	4.02	Meledak		Menghilang	
7	1	89	5	1.88	2.21	6.12	4.43	Meledak	Bertahan		5.11
	2	112	7	2.00	2.21	4.34	5.11			Menghilang	
	3	133	5	1.88	1.54	5.33	2.31	Meledak			
	4	123	8	2.97	2.21	4.34	3.88			Menghilang	
8	1	112	7	2.97	2.21	4.34	5.95			Menghilang	6.83
	2	137	10	1.78	2.21	7.21	6.83				
	3	112	5	2.95	2.76	4.34	4.46	Meledak		Menghilang	
	4	112	7	3.00	2.21	4.34	4.55			Menghilang	
9	1	137	10	1.78	2.76	7.21	6.83				7.14
	2	143	10	1.78	1.98	4.64	4.32			Menghilang	
	3	137	10	2.91	2.76	5.99	7.14				
	4	77	8	1.78	2.76	7.21	6.88		Bertahan		
10	1	80	6	2.91	1.98	5.52	8.21	Meledak	Bertahan		8.21
	2	137	10	3.0	1.43	5.99	6.04				
	3	137	10	2.91	3.00	4.44	7.42			Menghilang	
	4	137	10	2.91	2.76	5.99	5.27				
11	1	80	6	2.91	2.76	5.52	9.63	Meledak	Bertahan		9.63
	2	98	8	1.33	2.76	4.35	5.21		Bertahan	Menghilang	
	3	104	9	2.91	3.00	5.52	6.55				
	4	80	6	2.91	1.56	5.52	8.19	Meledak	Bertahan		
12	1	80	8	1.43	3.0	8.00	9.89	Meledak	Bertahan		9.89
	2	126	6	2.91	2.76	5.52	9.42	Meledak			
	3	140	6	1.00	2.76	7.50	8.32	Meledak			
	4	80	6	2.91	1.89	5.52	2.49	Meledak	Bertahan		
13	1	80	6	1.43	1.56	8.00	10.00	Meledak	Bertahan		10.00
	2	130	10	1.43	2.76	7.56	4.23				
	3	115	9	2.38	1.90	5.78	5.70				
	4	80	6	1.43	2.76	8.00	8.97	Meledak	Bertahan		

Keterangan :

Pop : Populasi

Idv : Individu

Fit : nilai *fitness*

Fittest : *Fitness* tertinggi dalam satu populasi

Dari hasil uji coba diatas dapat dilihat bahwa perilaku individu b8erubah dari generasi ke generasi sampai mendapatkan solusi optimal, namun ada beberapa individu yang tidak memiliki perilaku karena beberapa kromosomnya tidak memenuhi syarat untuk disisipkannya perilaku artinya individu tersebut memiliki nilai kromosom yang baik.

Tabel 4.21. Keterangan Hasil Uji Coba Algoritma

Populasi	Fittest	Keterangan
1	2.41	Pembangkitan awal
2	2.24	Hasil kurang sempurna karena nilai fittest turun dari generasi sebelumnya
3	3.65	Hasil sempurna karena nilai fittest meningkat dari sebelumnya
4	3.65	Hasil kurang sempurna karena nilai fittest tetap atau tidak ada peningkatan
5	4.21	Hasil sempurna karena nilai fittest meningkat dari sebelumnya
6	4.43	Hasil sempurna karena nilai fittest meningkat dari sebelumnya

7	5.11	Hasil sempurna karena nilai fittest meningkat dari sebelumnya
8	6.83	Hasil sempurna karena nilai fittest meningkat dari sebelumnya
9	7.14	Hasil sempurna karena nilai fittest meningkat dari sebelumnya
10	8.21	Hasil sempurna karena nilai fittest meningkat dari sebelumnya
11	9.63	Hasil sempurna karena nilai fittest meningkat dari sebelumnya
12	9.89	Hasil sempurna karena nilai fittest meningkat dari sebelumnya
13	10.00	Solusi optimal

Dari hasil uji coba algoritma dapat dilihat bahwa dari generasi ke generasi populasi mengalami evolusi. Nilai *fitness* akan terus ditingkatkan sampai menemukan solusi optimal. Dalam hal ini peneliti mengatur parameter nilai maksimum sebesar 10. Perulangan berakhir pada generasi ke-13 ketika angka fittest sudah mencapai 10. Setiap individu dalam populasi yang memiliki *fitness* tertinggi, individu akan diturunkan pada generasi selanjutnya. Sebagai contoh pada populasi 1, individu terbaik adalah individu 2 maka karakteristik dan perilaku individu tersebut akan diturunkan pada populasi baru. Pada populasi 2 hasil kurang maksimal dikarenakan nilai *fitness* individu yang di generate pada populasi 2 belum meningkat dari populasi sebelumnya. Pada populasi 3 individu mengalami peningkatan nilai *fitness* yaitu 3.65. Di populasi 13 akhirnya solusi terbaik didapatkan setelah mengalami perubahan-perubahan karakteristik dan

perilaku setiap generasinya. Prosentase individu mengalami peningkatan nilai *fitness* setiap generasinya dihitung berdasarkan rumus berikut :

$$\text{Prosentase Kecepatan} = \frac{\Sigma \text{populasi} - \Sigma \text{hasilkurangsempurna}}{\Sigma \text{populasi}} \times 100\%$$

Keterangan :

$\Sigma \text{populasi}$ = jumlah total populasi yang terjadi

$\Sigma \text{hasilkurangsempurna}$ = jumlah total hasil dimana fittest mengalami penurunan nilai fittest atau fittest tetap dari generasi sebelumnya.

Dari 13 iterasi populasi, prosentase kecepatan individu mengalami peningkatan nilai *fitness* setiap generasinya adalah :

$$\text{Prosentase Kecepatan} = \frac{13-2}{13} \times 100\% = 84.6\%$$

Hasil uji coba diatas menunjukkan bahwa NPC musuh semakin baik dalam karkteristik dan perilaku, hal itu akan menambah tantangan untuk pemain dalam menghadapi NPC musuh.

4.3. Integrasi Game *Fight For Hijaiyah* dengan Islam

Huruf hijaiyah merupakan pelajaran dasar untuk membaca Al-Quran. Dalam membaca pun tentunya harus hati-hati dan mengerti mahrajnya dengan benar. Dalam firman Allah Surat al-Baqarah ayat 121 :

الَّذِينَ آتَيْنَاهُمُ الْكِتَابَ يَتْلُونَهُ حَقَّ تِلَاوَتِهِ أُولَٰئِكَ يُؤْمِنُونَ بِهِ ۗ وَمَنْ يَكْفُرْ بِهِ فَأُولَٰئِكَ هُمُ الْخٰسِرُونَ ۙ ١٢١

“121. Orang-orang yang telah Kami berikan Al Kitab kepadanya, mereka membacanya dengan bacaan yang sebenarnya, mereka itu beriman kepadanya. Dan barangsiapa yang ingkar kepadanya, maka mereka itulah orang-orang yang rugi.”

Tafsir dan penjelasan QS. Al-Baqarah ayat 121 dalam kitab tafsir Ibnu Katsir bahwasannya Abdur Razzaq meriwayatkan dari Ma'mar, dari Qatadah, yang dimaksud dengan orang-orang tersebut adalah orang Yahudi dan Nasrani. Pendapat ini merupakan Abdur Rahman ibnu Zaid Aslam dan dipilih oleh Ibnu Jarir. Sa'id meriwayatkan Qatadah, bahwa mereka adalah sahabat-sahabat Rasulullah Saw. Ibnu Abu Hatim mengatakan, telah menceritakan kepada kami ayahku, telah menceritakan kepada kami Ibrahim ibnu Musa dan Abdullah ibnu Imran Al-Asbahani yang mengatakan, telah menceritakan kepada Yahya ibnu Yaman, telah menceritakan kepada kami Usama ibnu Zaid, dari ayahnya, dari Umar ibnul Khattab, sehubungan dengan tafsir firman-Nya, *”Mereka membacanya dengan bacaan yang sebenarnya”* (Al-Baqarah 121). Yang dimaksud dengan bacaan yang sebenarnya ialah apabila si pembaca melewati penyebutan tentang surga, maka dia memohon surga kepada Allah. Apabila ia melewati penyebutan neraka, maka ia meminta perlindungan dari neraka.

BAB V

PENUTUP

5.1. Kesimpulan

Algoritma Genetika diimplementasikan melalui beberapa tahapan yaitu dimulai dari pembangkitan populasi dimana setiap individu diatur nilai kromosomnya, selanjutnya tahapan seleksi dimana individu diseleksi menggunakan metode *Roulette Wheel*, hasil dari seleksi tersebut dijadikan masukan untuk proses *cross over* dan mutasi. Algoritma genetika dapat diterapkan dengan baik dan sangat cocok digunakan dalam Game “*Fight For Hijaiyah*” untuk menentukan karakteristik dan perilaku NPC musuh. Hal ini dibuktikan dengan hasil uji coba algoritma dimana nilai *fitness* dari individunya selalu bertambah baik setiap generasi artinya NPC musuh selalu berkembang karakteristik dan perilakunya dari generasi ke generasi. Dari beberapa proses iterasi populasi, sudah mendapatkan hasil yang cukup baik, prosentase kecepatan peningkatan *fitness* yang terjadi dari generasi ke generasi pada NPC dengan metode Algoritma Genetika pada game “*Fight For Hijaiyah*” sebesar 84.6%.

5.2. Saran

Penulis sadar bahwa permainan ini masih banyak kekurangan dan masih dalam tahap pengembangan. Oleh karena itu ada beberapa saran dari penulis untuk pengembangan selanjutnya, diantaranya :

1. Mengembangkan aplikasi game ini dengan tampilan aplikasi, model dan animasi yang menarik.
2. Mengembangkan aplikasi game ini pada platform yang diminati saat ini, yaitu Android dan IOS.
3. Mengembangkan NPC musuh dengan meningkatkan kemampuan adaptif strategi pada pemain.



DAFTAR PUSTAKA

Al-Quran dan Terjemahannya, Syamil Al-Quran

Abdullah. 2006. *Tafsir Ibnu Katsir Jilid 8*. Jakarta. Penerbit : Penebar Sunnah.

Ernawati, “*Pengaruh Metode Iqro Animasi Terhadap Kemampuan Mengeja Huruf Hijaiyah Pada Anak Tunagrahita Ringan*”, 2012

Pramling Samuelsson, I., & Sheridan, S. (2003). *Delaktighet som va`rdering och pedagogik [Involvement as evaluation and practice; in Swedish]*. Pedagogisk Forskning i Sverige. Tema: Barns perspektiv och barnperspektiv, 1–2, 70–84.

Clark, R. E. (2006) “*Evaluating the Learning and Motivation Effects of Serious Games. Rosier school of Education Center for Creative Technologies*” Available online at: http://projects.ict.usc.edu/itgs/talks/Clark_Serious

Nate Hagb, Oriël Bergig, Jihad El-Sana & Mark Billingham, “*Shape recognition and Pose Estimation for Mobile Augmented Reality*”, 2011

Georgios N. Yannakakis,” *AI in Computer Games: Generating Interesting Interactive Opponents by the use of Evolutionary Computation*”, 2005

Holland, J. H. (1975) *Adaptation in Natural and Artificial System*. University Michigan Press.

Mohammad Gilang Kautzar HW,” *Implementasi Algoritma Genetika Dalam Menyelesaikan Sebuah Persoalan Anagram Scrabble*”, 2007

Omid E. David, et al, *Genetic Algorithms for Evolving Computer Chess Programs*, 2014

Dini Nur Setyorini, “*Sistem Penjadwalan Proyek Jaringan Pipa Air Bersih Menggunakan Algoritma Genetika*”, 2012

Moshe Sipper, “*Evolving Game-Playing Strategies with Genetic Programming*”, 2008

Hans K. G. Fernlund, et al, “*Learning tactical human behavior through observation of human performance*”, 2006

Francisco Azuaje, “*A computational evolutionary approach to evolving game strategy and cooperation*”, 2003

Utomo Sarjono Putro, Kijima, K. And Takahashi, S, “*Adaptive Learning of Hypergame Situations by Using Genetic Algorithm*”, 2000

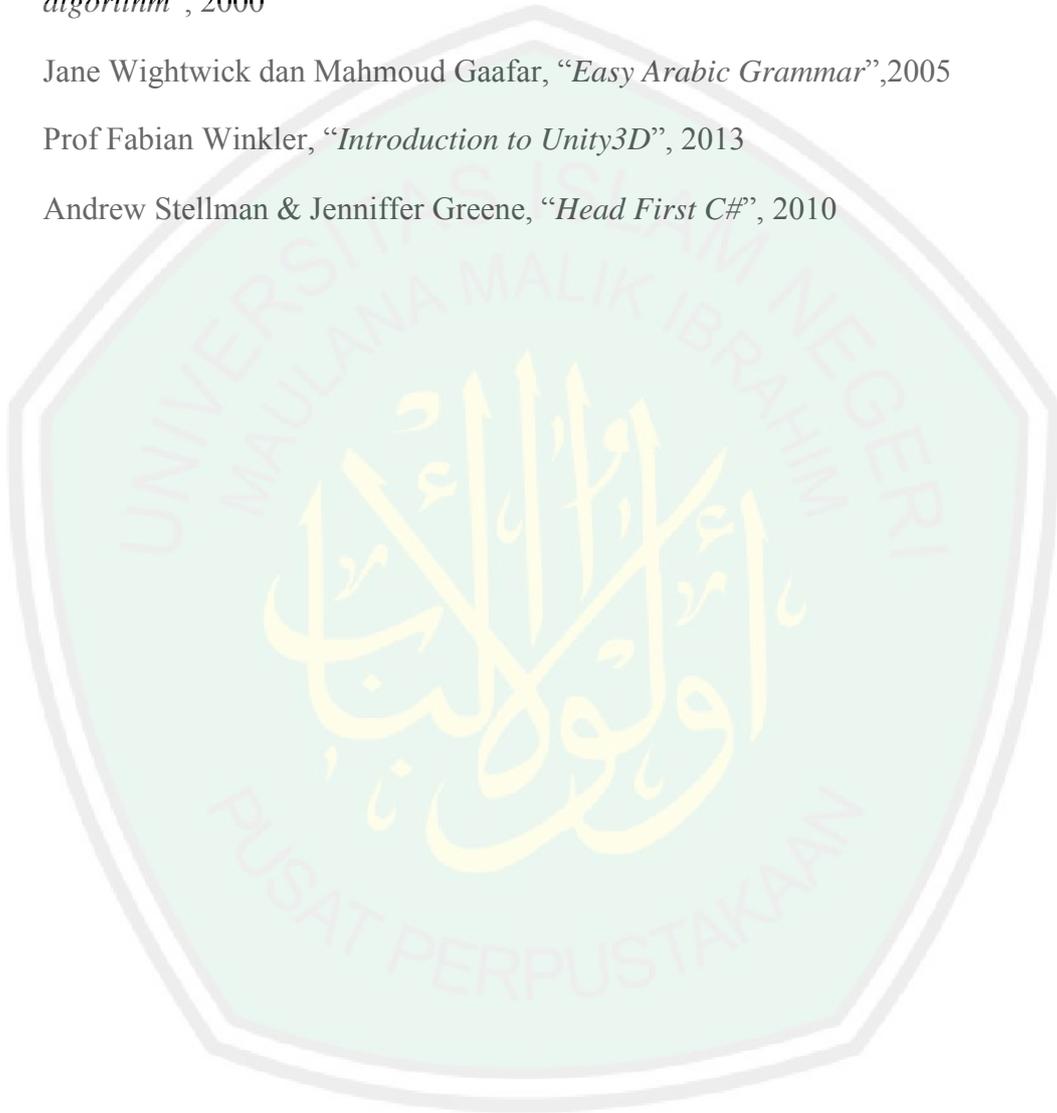
Johan Saputra, “*Perancangan dan pembuatan aplikasi untuk mengoptimasi penyusunan iklan gambar dengan metode algoritma genetik*”, 2008

Fathelalem F. Ali, et al, “*Playing the Rock-Paper-Scissors game with a genetic algorithm*”, 2000

Jane Wightwick dan Mahmoud Gaafar, “*Easy Arabic Grammar*”, 2005

Prof Fabian Winkler, “*Introduction to Unity3D*”, 2013

Andrew Stellman & Jenniffer Greene, “*Head First C#*”, 2010



LAMPIRAN

Pseodocode Pembentukan Individu

```

Program Individu
Deklarasi
Length : integer
Kromosom[Length] : float[]
healthPoint, attackDamage, Fitness, Lifetime, totalDamageDone : integer
attackSpeed, attackRange, movementSpeed : float
Perilaku : String

Proses
healthPoint = Random(value)
attackDamage = Random(value)
attackSpeed = Random(value)
attackRange = Random(value)
movementSpeed = Random(value)
Kromosom[0] = healthPoint
Kromosom[1] = attackDamage
Kromosom[2] = attackSpeed
Kromosom[3] = attackRange
Kromosom[4] = movementSpeed
Fitness = totalDamageDone/Lifetime
If healthPoint>100 then
    Perilaku="Meledak"
Else if healthPoint<=100 then
    Perilaku="Bertahan"
Else if movementSpeed<=6 then
    Perilaku="Menghilang"
Endif

return (Perilaku, Kromosom[], Fitness)
  
```

Lampiran 1. Pseodocode Pembentukan Individu

Pseodocode Pembangkitan Populasi

```

Program Population
Deklarasi
populationSize : integer
Population[populationSize] : Individu[] ← Inisialisasi dari program Individu
Individual : Individu ← Inisialisasi dari program Individu
  
```

```

Proses
For i:=0 to populationSize do
    PopulationSize[i]=Individual
Endfor

return(Population[])

```

Lampiran 2. Pseudocode Pembangkitan Populasi

Pseudocode Proses Seleksi

```

Program Seleksi
Deklarasi
populationSize : integer
individualFitness, fittestFitness : integer
Individual[]: Individu[] ← Inisialisasi dari program Individu
Fittest : Individu ← Inisialisasi dari program Individu

Proses
For i:=0 to populationSize do
    If fittestFitness <= individualFitness then
        Fittest=Individual[i]
    Endif
Endfor

return(Fittest)

```

Lampiran 3. Pseudocode Seleksi

Pseudocode Proses Cross Over

```

Program CrossOver
Deklarasi
Length, populationSize : integer
crossOverRate : float
individuKromosomA[Length], individuKromosomB[Length] : Individu

Proses
Input(individuKromosomA[], individuKromosomB[])
For i:=0 to populationSize do

```

```

If Random(value) <= crossOverRate then
    individuKromosomA[i]= individuKromosomB[i]
Endif
Endfor

Output (individuKromosomA[], individuKromosomB[])

```

Lampiran 4. Pseudocode Proses Cross Over

Pseudocode Proses Mutasi

```

Program Mutasi
Deklarasi
Length, populationSize : integer
mutationRate : float
individuKromosom[Length]: Individu ← Inisialisasi dari program Individu

Proses
For i:=0 to populationSize do
    If Random(value) <= mutationRate then
        If i = 0 then
            individuKromosom[0] = Random(value)
        Else If i = 1 then
            individuKromosom[1] = Random(value)
        Else If i = 2 then
            individuKromosom[2] = Random(value)
        Else If i = 3 then
            individuKromosom[3] = Random(value)
        Else If i = 4 then
            individuKromosom[4] = Random(value)
        EndIf
    Endif
Endfor

return (individuKromosom[])

```

Lampiran 5. Pseudocode Proses Mutasi