

**IMPLEMENTASI *PERCEPTRON* UNTUK MEMBERIKAN
PERILAKU *NON PLAYER CHARACTER* (NPC) PADA
GAME ARABIC HUNTER BERBASIS ANDROID**

SKRIPSI

oleh:
SYAFEI KARIM
NIM. 10650056



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2014**

**IMPLEMENTASI *PERCEPTRON* UNTUK MEMBERIKAN
PERILAKU *NON PLAYER CHARACTER* (NPC) PADA
GAME ARABIC HUNTER BERBASIS ANDROID**

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:
SYAFEI KARIM
NIM. 10650005**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2014**

**IMPLEMENTASI *PERCEPTRON* UNTUK MEMBERIKAN
PERILAKU *NON PLAYER CHARACTER* (NPC) PADA
GAME ARABIC HUNTER BERBASIS ANDROID**

SKRIPSI

Oleh :

Nama : Syafei Karim

NIM : 10650056

Jurusan : Teknik Informatika

Fakultas : Sains dan Teknologi

Telah diperiksa dan disetujui untuk diuji :

Tanggal :

Dosen Pembimbing I

Dosen Pembimbing II

Yunifa Miftachul Arif, M.T

M. Ainul Yaqin, M.Kom

NIP. 19830616 201101 1 004

NIP. 19761013 200604 1 004

Mengetahui,

Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdiان

NIP. 19740424 200901 1 008

**IMPLEMENTASI *PERCEPTRON* UNTUK MEMBERIKAN
PERILAKU *NON PLAYER CHARACTER* (NPC) PADA
GAME ARABIC HUNTER BERBASIS ANDROID**

SKRIPSI

Oleh :

Syafei Karim

NIM. 10650056

Telah dipertahankan di depan dewan penguji skripsi
dan dinyatakan diterima sebagai salah satu persyaratan
untuk memperoleh gelar sarjana komputer (S.Kom)

Tanggal : 11 Juli 2014

Susunan Dewan Penguji:

Tanda Tangan

- | | | | |
|-----------------------|---|---|---|
| 1. Penguji Utama | : Fresy Nugroho, M.T
NIP. 19710722 201101 1 001 | (|) |
| 2. Ketua Penguji | : <u>Fachrul Kurniawan, M.MT</u>
NIP. 19771020 200901 1 001 | (|) |
| 3. Sekretaris Penguji | : <u>Yunifa Miftachul Arif, MT</u>
NIP. 19830616 201101 1 004 | (|) |
| 4. Anggota Penguji | : <u>M. Ainul Yaqin, M.Kom</u>
NIP. 19761013 200604 1 004 | (|) |

Mengetahui,
Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdian

NIP. 19740424200901 1 008

PERNYATAAN KEASLIAN PENELITIAN

Saya yang bertandatangan di bawah ini:

Nama : Syafei Karim

NIM : 10650056

Fakultas/Jurusan : Sains dan Teknologi / Teknik Informatika

Judul Penelitian : Implementasi *Perceptron* untuk Memberikan Perilaku *Non Player Character* (NPC) pada *Game Arabic Hunter* berbasis Android

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilalihan 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, 08 Juli 2014
Yang membuat pernyataan,

Syafei Karim
NIM. 10650056

MOTTO

“To get a success, your courage must be greater than your fear.”

“The best sword that you have is a limitless patience.”

PERSEMBAHAN

Segala Puji Syukur atas limpahan rahmat, taufik dan hidayah kepada Allah SWT atas terselesaikannya skripsi ini. Tak lupa ucapan terimakasih sebanyak – banyaknya kepada :

- ❖ Ayah dan Ibu yang selalu memberi dukungan mulai dari materi dan doa yang tiada henti-hentinya untuk saya. **YOU ARE MY EVERYTHING.**
- ❖ Kakak dan adik-adikku yang selalu menyemangati dan memberikan dukungan do'a.
- ❖ Dosen-dosen di jurusan Teknik Informatika yang telah membimbing saya selama menjalani studi di jurusan Teknik Informatika.
- ❖ Sahabat-sahabat MMM: Dita, Ade, Dewi, Ita, Puspa, Vivid, Dian, Haris, Zaenal, Novi, Gery, Naufal, Agus dan Jun yang selalu setia mendukung dan menjadi keluarga baru ku selama di Malang. Terimakasih buat kalian yang selalu membuat waktu ku lebih menyenangkan bersama kalian. **YOU GUYS WILL ALWAYS BE IN MY HEART.**
- ❖ Teman-Teman Kontrakan yang selalu memberikan masukan dan semangat dan juga selalu mengganggu ku dalam pembuatan skripsi ini.
- ❖ Rizal dan Taufan teman seperjuanganku, terima kasih selama ini kalian selalu membantu ku dalam menuntut ilmu di kampus kita tercinta.
- ❖ Terima kasih buat Saifuddin yang selalu membantu ku dalam kesusahan dan berjuang bersama dalam menyelesaikan skripsi ini.
- ❖ Teman – teman seperjuangan skripsi, terimakasih telah memberikan semangat yang luar biasa semoga setelah perjuangan ini kita tetap saling memberikan semangat.
- ❖ Teman-teman INFINITY terimakasih atas do'a dan dukungannya. Semoga kita sukses bersama.

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Alhamdulillah rabbil'Alamin penulis haturkan kehadiran Allah SWT yang telah memberikan rahmat, hidayah, dan ridha-Nya, sehingga penulis dapat menyelesaikan studi di jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang sekaligus menyelesaikan skripsi ini dengan baik dan lancar.

Selanjutnya penulis menghaturkan ucapan terima kasih kepada semua pihak yang telah memberikan doa, harapan, dan semangat untuk terselesaikannya skripsi ini. Ucapan terima kasih, penulis sampaikan kepada :

1. Prof. Dr. H. Mudjia Rahardjo, M.Si selaku rektor UIN Maulana Malik Ibrahim Malang, yang telah banyak memberikan pengetahuan dan pengalaman yang berharga.
2. Dr. drh. Bayyinatul Muchtaromah, M.Si selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Cahyo Crysdiان selaku ketua Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Yunifa Miftachul Arif, M.T dan M. Ainul Yaqin, M.Kom selaku dosen pembimbing skripsi, yang telah banyak memberikan pengarahan dan pengalaman yang berharga.
5. Segenap civitas akademika jurusan Teknik Informatika, terutama seluruh dosen, terima kasih atas segenap ilmu dan bimbingannya.

6. Ayahanda Sabaruddin dan Ibunda Laila tercinta yang senantiasa memberikan doa, spirit, tenaga, biaya, dan restunya kepada penulis dalam menuntut ilmu.
7. Semua pihak yang ikut membantu dalam menyelesaikan skripsi ini baik berupa materiil maupun moril.

Peneliti menyadari bahwa dalam penyusunan skripsi ini masih terdapat kekurangan dan peneliti berharap semoga skripsi ini bisa memberikan manfaat kepada para pembaca khususnya bagi peneliti secara pribadi. *Amin Ya Rabbal Alamin.*

Wassalamu'alaikum Wr. Wb.

Malang, Juli 2014

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN.....	ii
LEMBAR PERSETUJUAN.....	iii
LEMBAR PENGESAHAN.....	iv
HALAMAN PERNYATAAN	v
MOTTO	vi
PERSEMBAHAN.....	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xiv
ABSTRAK	xv
ABSTRACT	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah.....	4
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
1.6 Sistematika Penulisan	5
1.7 Metodologi Penelitian.....	6
BAB II TINJAUAN PUSTAKA	8
2.1 Sejarah dan Gambaran Umum Android.....	8
2.2 Permainan (<i>Game</i>).....	10
2.2.1 Jenis Permainan (<i>Genre Games</i>).....	12
2.2.2 Komponen Permainan (<i>Component Games</i>).....	14
2.3 Permainan Kecerdasan Buatan (<i>Game AI</i>).....	16
2.4 Non Player Character (NPC).....	17
2.5 Jaringan Saraf Tiruan (JST)	18
2.6 Perceptron	24
BAB III ANALISA DAN PERANCANGAN SISTEM.....	28
3.1 Analisa dan Perancangan Sistem.....	28

3.1.1	Keterangan Umum <i>Game</i>	28
3.1.2	Penampilan <i>Game</i>	29
3.1.3	Deskripsi Karakter.....	29
3.2	Perancangan Kecerdasan Buatan Perilaku NPC	30
3.2.1	FSM NPC Penolong	32
3.2.2	FSM NPC Jahat	33
3.2.3	Perancangan Perceptron pada NPC	33
3.3	Perancangan Aplikasi <i>Game</i>	36
3.3.1	Perancangan Antarmuka <i>Splashscreen</i>	36
3.3.2	Perancangan Antarmuka Menu <i>Game</i>	37
3.3.3	Perancangan Antarmuka Tentang <i>Game</i>	38
3.3.4	Perancangan Antarmuka <i>Game</i>	39
3.3.5	Perancangan Level Permainan	39
3.3.6	Cara Memainkan <i>Game</i>	40
3.3.7	Kebutuhan Smartphone Pemain	41
BAB IV HASIL DAN PEMBAHASAN		42
4.1	Implementasi Sistem.....	42
4.2	Implementasi Kecerdasan Buatan Perilaku NPC	43
4.2.1	Kecerdasan Buatan Perilaku NPC Pocong	44
4.2.2	Kecerdasan Buatan Perilaku NPC Jin.....	47
4.2.3	Kecerdasan Buatan Perilaku NPC Ustadz	48
4.3	Implementasi Aplikasi <i>Game</i>	50
4.3.1	Antarmuka <i>Splashscreen</i>	50
4.3.2	Antarmuka Menu <i>Game</i>	51
4.3.3	Scene <i>Game</i> Pertama	53
4.3.4	Scene <i>Game</i> Kedua.....	55
4.3.5	Scene <i>Game</i> Ketiga	58
4.4	Uji Coba.....	60
4.4.1	Uji Coba Algoritma Perceptron.....	60
4.4.2	Uji Coba Aplikasi	70
4.5	Integrasi <i>Game</i> Arabic Hunter dengan Islam.....	72
BAB V PENUTUP		75
5.1	Kesimpulan	75
5.2	Saran	76

DAFTAR PUSTAKA	77
LAMPIRAN-LAMPIRAN.....	79



DAFTAR TABEL

Tabel 2.1 Operator AND.....	22
Tabel 2.2 Operator OR.....	23
Tabel 2.3 Operator NOT.....	23
Tabel 2.4 Perhitungan Pelatihan Perceptron.....	27
Tabel 3.1 Perhitungan Manual Perceptron.....	36
Tabel 3.2 Keterangan Fungsi Tombol Menu.....	38
Tabel 3.3 Daftar Spesifikasi Smartphone.....	41
Tabel 4.1 Keterangan Fungsi Tombol Menu Permainan.....	51
Tabel 4.2 Hasil Uji Coba Nilai Epoh.....	61
Tabel 4.3 Hasil Uji Coba Waktu Eksekusi.....	64
Tabel 4.4 Data NPC Jahat.....	67
Tabel 4.5 Data NPC Baik.....	68
Tabel 4.6 Hasil Pengujian Aplikasi.....	71

DAFTAR GAMBAR

Gambar 1.1 Metodologi Penelitian.....	7
Gambar 2.1 Arsitektur Android.....	10
Gambar 2.2 Hirarki Gerak Perilaku.....	18
Gambar 2.3 Struktur Dasar Jaringan Saraf Tiruan	20
Gambar 2.4 Elemen JST	20
Gambar 2.5 Fungsi Pengaktif.....	21
Gambar 2.6 Arsitektur Jaringan Sederhana Operasi AND	22
Gambar 2.7 Arsitektur Jaringan Sederhana Operasi OR	23
Gambar 2.8 Arsitektur Jaringan Sederhana Operasi NOT.....	24
Gambar 2.9 Single Perceptron.....	25
Gambar 2.10 Contoh Kasus Peceptron	27
Gambar 3.1 FSM NPC Penolong	32
Gambar 3.2 FSM NPC Jahat	33
Gambar 3.3 Blok Diagram Implementasi Perceptron.....	35
Gambar 3.4 Antarmuka <i>Splashscreen</i>	37
Gambar 3.5 Antarmuka Menu <i>Game</i>	37
Gambar 3.6 Antarmuka Tentang <i>Game</i>	38
Gambar 3.7 Antarmuka <i>Game</i>	39
Gambar 4.1 Tampilan <i>Splashscreen</i>	50
Gambar 4.2 Tampilan Menu Utama	52
Gambar 4.3 Tampilan Menu Suara Aktif.....	52
Gambar 4.4 Tampilan Menu Suara Nonaktif.....	52
Gambar 4.5 Tampilan Informasi Permainan	53
Gambar 4.6 Tampilan <i>Loading</i>	53
Gambar 4.7 Cara Bermian Scene Pertama	54
Gambar 4.8 Tampilan Scene Pertama.....	54
Gambar 4.9 Tampilan NPC Mengejar dengan Pelan.....	55
Gambar 4.10 Tampilan NPC Baik Menolong Player	55
Gambar 4.11 Cara bermain Scene Kedua	56
Gambar 4.12 Tampilan Scene Kedua	56
Gambar 4.13 Tampilan Poin Berkurang	57
Gambar 4.14 Tampilan NPC Jahat Bertambah Besar.....	57
Gambar 4.15 Tampilan NPC Baik Menolong Pemain	58
Gambar 4.16 Cara Bermain Scene Ketiga	59
Gambar 4.17 Tampilan Scene Ketiga	59
Gambar 4.18 Tampilan NPC Baik Menolong Pemain	60
Gambar 4.19 Tampilan Permainan Selesai	60
Gambar 4.20 Hasil Uji Coba NPC Jahat.....	69
Gambar 4.21 Hasil Uji Coba NPC Baik	70

ABSTRAK

Karim, Syafei. 2014. **Implementasi Perceptron untuk Memberikan Perilaku Non Player Character (NPC) pada Game Arabic Hunter berbasis Android.** Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Yunifa Miftachul Arif, M.T (II) M. Ainul Yaqin, M.Kom

Kata Kunci: *Bahasa Arab, Game Side Scrolling, Jaringan Syaraf Tiruan, Perceptron*

Al-Qur'an diturunkan dengan menggunakan bahasa arab. Bahasa arab sangat penting untuk dipelajari. Banyak cara untuk belajar bahasa arab. Berbagai metode dan media belajar dibuat untuk memudahkan dalam pembelajaran. Salah satunya adalah media *game*. *Game* ini berbentuk *side scrolling 2D* dimana pemain berpetualang untuk belajar bahasa arab. Bahasa arab yang dipelajari pada *game* ini berupa mufradat. Tujuan dari permainan ini adalah mengumpulkan huruf-huruf mufradat dan poin yang berada di area permainan.

Berdasarkan hasil pengujian dapat disimpulkan bahwa implementasi perceptron untuk memberikan perilaku pada NPC pada *game* Arabic hunter dapat berjalan dengan baik. Algoritma perceptron diterapkan pada NPC. Algoritma ini digunakan untuk memberikan perubahan perilaku NPC. Perubahan perilaku akan diproses menyesuaikan jumlah mufradat dan jumlah poin yang didapatkan pemain. Dari uji coba algoritma, pada proses *learning* dapat diketahui bahwa semakin besar nilai *learning rate* maka semakin kecil nilai epoch yang didapat. Dari uji coba tersebut dihasilkan *learning rate* = 1, *threshold* = 0.6, nilai bias = -1, waktu eksekusi = 1.433 detik. Pada proses *learning* dihasilkan 93% berhasil dan 7% gagal.

ABSTRACT

Karim, Syafei. 2014. **Implementation Perceptron to Give Behavior Non Player Character (NPC) in Game Arabic Hunter Based Android.** Thesis. Informatics Department of Faculty of Science and Technology. Maulana Malik Ibrahim State Islamic University, Malang. Promotor: (I) Yunifa Miftachul Arif, M.T (II) M. Ainul Yaqin, M.Kom

Keywords: *Arabic, Game Side Scrolling, Neural Network, Perceptron*

Al-Qur'an was revealed in Arabic language. Arabic is very important to learn. Many ways to learn Arabic. Many ways methods and media of learning were made to facilitate learning. One is the media game. This game type 2D side-scrolling adventure in which the player to learn the Arabic language. Arabic learned in this game is mufradat. The objective of the game is to collect mufradat and points that are in the game area.

Based on the test results it can be concluded that the Implementation Perceptron to Give Behavior Non Player Character (NPC) in Game Arabic Hunter can be run well. Perceptron algorithm applied to the NPC. This algorithm is used to give behavior change to NPC. Behavior change will be processed adjust the number of mufradat and the number of points earned player. From the test algorithm, the training process can be seen that the greater the value of learning rate, the smaller the epoch value obtained. From the test resulting learning rate = , threshold = 0.6, bias value = -1, and execution time = 1.433 sec. In the learning process produced 93% success and 7% failed.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Jaringan Saraf Tiruan merupakan bagian ilmu dari kecerdasan buatan (*Artificial Intelligence*). JST dibangun bertujuan untuk meniru kerja otak makhluk hidup dalam menyimpan, belajar, dan mengambil kembali pengetahuan yang telah tersimpan dalam sel saraf atau neuron. JST memiliki beberapa metode pelatihan terbimbing yang telah diciptakan oleh peneliti, diantaranya Hebbian, ADALINE, Hopfield, Backpropagation dan tak terkecuali Perceptron. Perceptron merupakan salah satu metode pembelajaran terawasi (*supervise learning method*) dalam jaringan saraf tiruan (JST). Umumnya, perceptron digunakan untuk mengklasifikasikan suatu pola tertentu.

Permainan (*game*) adalah aplikasi pembunuh untuk *smartphone* saat ini. Menurut seorang analis, lebih dari 23% dari seluruh pengguna ponsel selama 13 tahun di Amerika Serikat bermain *game* di ponsel mereka dan prosentase terus meningkat, terutama untuk 60 juta lebih pengguna *smartphone* (comScore, 2010). Menurut Nielsenwire (2010), 65% pengguna *smartphone* telah memainkan *game* mobile pada ponsel mereka di beberapa titik. Dalam perhitungannya, yang berarti sekitar 40 juta orang saat ini bermain *game* di *smartphone* mereka.

Teknologi *game* memiliki banyak genre seperti petualangan, olahraga, puzzle, strategi, bertarung, dan tak terkecuali *game* edukasi. Dengan *game* edukasi seorang *player* dapat bermain dan belajar. Sehingga *player* tidak hanya

mendapatkan kesenangan dalam bermain tetapi juga mendapatkan ilmu pengetahuan. Dalam firman Allah SWT pada Surat al-Alaq ayat 1-5.

أَقْرَأْ بِاسْمِ رَبِّكَ الَّذِي خَلَقَ ﴿١﴾ خَلَقَ الْإِنْسَانَ مِنْ عَلَقٍ ﴿٢﴾ أَلَمْ يَكُنْ مِنْ رَجُلٍ وَرَجُلٍ كَافٍ ﴿٣﴾ خَلَّمَ بِالْقَلَمِ ﴿٤﴾ عَلَّمَ الْإِنْسَانَ مَا لَمْ يَعْلَمْ ﴿٥﴾

Artinya:

(1) Bacalah dengan (menyebut) nama Tuhanmu Yang menciptakan. (2) Dia telah menciptakan manusia dari segumpal darah. (3) Bacalah, dan Tuhanmulah Yang Maha Pemurah (4) Yang mengajar (manusia) dengan perantaran qalam (alat tulis). (5) Dia mengajar kepada manusia apa yang tidak diketahuinya.

Di dalam ayat yang mula turun ini telah jelas penilaian yang tertinggi kepada kepandaian membaca dan menulis. Dalam ayat tersebut terdapat perintah Allah untuk membaca. Dengan membaca seseorang dapat mengetahui dari apa yang telah ia baca. Membaca yang dimaksud bukan hanya membaca sebuah tulisan tetapi juga membaca situasi maupun fenomena alam. Terdapat hubungan antara ayat 1 dan 4 yaitu perintah Allah untuk membaca dan yang dibaca adalah sebuah tulisan. Sehingga dengan tulisan yang dibaca maka seseorang dapat belajar dan mendapatkan ilmu pengetahuan dari membaca tulisan. Dari membaca tulisan itulah maka manusia sangat diwajibkan untuk memiliki ilmu pengetahuan. Seperti dengan hadits yang diriwayatkan oleh Ibnu Abdil Bari.

طَلَبُ الْعِلْمِ فَرِيضَةٌ عَلَى كُلِّ مُسْلِمٍ وَ مُسْلِمَةٍ - رواه ابن عبد البر

Artinya : “menuntut ilmu wajib atas setiap muslim dan muslimah” (HR. Ibnu Abdil Bari).

Hadits diatas menunjukkan bahwa setiap muslim dan muslimah sangat wajib menuntut ilmu. Menuntut yang dimaksud adalah mempelajari atau menguasai ilmu pengetahuan. Ilmu yang dipelajari adalah bahasa arab karena bahasa arab itu

adalah bahasa al-Qur'an. Al-Qu'an merupakan kitab suci umat beragama muslim. Setiap orang muslim harus bisa membaca al-Qur'an. Karena itulah bahasa arab wajib untuk dipelajari. Agar pembelajaran bahasa arab lebih menyenangkan dan mudah dipelajari maka dibuatkanlah sebuah media yang menyenangkan seperti *game*. Bahasa arab yang dipelajari pada *game* Arabic Adventures adalah mufradat.

Pada khususnya *game* memiliki beberapa komponen yang penting yaitu skenario (alur cerita), level (tingkatan), skor (nilai), karakter, dan obstacle (rintangan). Penelitian tentang AI (Artificial Intelligence) pada NPC (Non-Player Character) dalam *game*, hingga saat ini masih terus di kembangkan (JinHyuk Hong dan Sung-Bae Cho, 2010). AI tersebut dikembangkan untuk merancang perilaku NPC. Ketika mengatakan bahwa *game* sudah mempunyai AI yang baik, berarti bahwa karakter permainan menunjukkan perilaku yang konsisten dan realistis, bereaksi dengan tepat kepada tindakan pemain dan karakter lain.

Pada penelitian ini, peneliti akan meneliti perilaku NPC pada *game* "Arabic Hunter" dengan menerapkan Jaringan Saraf Tiruan Perceptron. *Game* tersebut akan dibuat ke dalam *mobile OS* android. Dengan mengandalkan teknologi informasi dan *game* sekiranya dapat membantu *player* belajar kosa kata dalam bahasa arab.

1.2 Identifikasi Masalah

Berdasarkan uraian latar belakang diatas, penulis ingin menerapkan *Jaringan Saraf Tiruan Perceptron* pada perilaku NPC untuk *game Arabic Hunter* berbasis android.

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Variabel yang digunakan adalah jumlah koin dan jumlah mufradat.
2. Aplikasi bertipe 2 dimensi
3. Aplikasi dibangun berbasis *mobile* dengan *OS Android*.
4. Aplikasi bersifat *single player*.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk mengembangkan aplikasi *game mobile* yang bertemakan islam yang berguna bagi anak-anak dan masyarakat dapat bermain *game* dan belajar ilmu pengetahuan.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah aplikasi *game mobile* yang bertemakan islam semakin berkembang. Semakin banyaknya anak-anak yang dapat bermain *game* yang bertemakan islam.

1.6 Sistematika Penulisan

Sistematika penulisan ini disusun untuk mempermudah dan memahami pembahasan penulisan skripsi. Terdapat lima bab dalam sistematika penulisan sebagai berikut:

BAB I Pendahuluan

Pendahuluan membahas tentang Latar Belakang, Identifikasi Masalah, Batasan Masalah, Tujuan Penelitian, Manfaat Penelitian, Sistematika Penulisan dan Metodologi Penelitian.

BAB II Tinjauan Pustaka

Landasan teori yang berisikan teori-teori yang didapat dari studi literatur yang mendasari dalam penyusunan tugas akhir ini, beserta dengan penyelesaian masalah yang diambil dalam penyusunan tugas akhir.

BAB III Analisa dan Perancangan Sistem

Pada bab ini berisi menganalisa kebutuhan *game* yang akan dibangun meliputi kebutuhan software dan langkah-langkah pembuatan *game*.

BAB IV Hasil dan Pembahasan

Menjelaskan tentang implementasi dan pengujian aplikasi serta uji coba dari aplikasi yang sudah dibuat.

BAB V Penutup

Berisi intisari dari hasil pembahasan yang berisikan kesimpulan dan saran yang dapat digunakan untuk penelitian selanjutnya.

1.7 Metodologi Penelitian

Pada penelitian ini, peneliti melakukan langkah-langkah yang harus dilalui.

Berikut tahapan prosedur penelitian:

1. Merumuskan Hipotesis

Pada tahap ini melakukan anggapan yang sifatnya sementara tentang objek yang diteliti.

2. Menentukan Desain Penelitian

Desain penelitian ini menggunakan model matematika.

3. Mengumpulkan Data

Pada tahap ini akan dilakukan pengumpulan data dari literatur-literatur terkait dengan penelitian ini. Literatur didapatkan dari buku, jurnal, atau skripsi terdahulu.

4. Mengolah Data

Setelah data dikumpulkan akan dilakukan analisis terhadap hasil pengumpulan data dari literatur yang telah didapatkan.

5. Membuat Kesimpulan

Pada tahap ini peneliti membuat kesimpulan yang sesuai dengan hipotesis yang diajukan dan membuat laporan hasil dari penelitian.



Gambar 1.1 Metodologi Penelitian

BAB II

TINJAUAN PUSTAKA

2.1 Sejarah dan Gambaran Umum Android

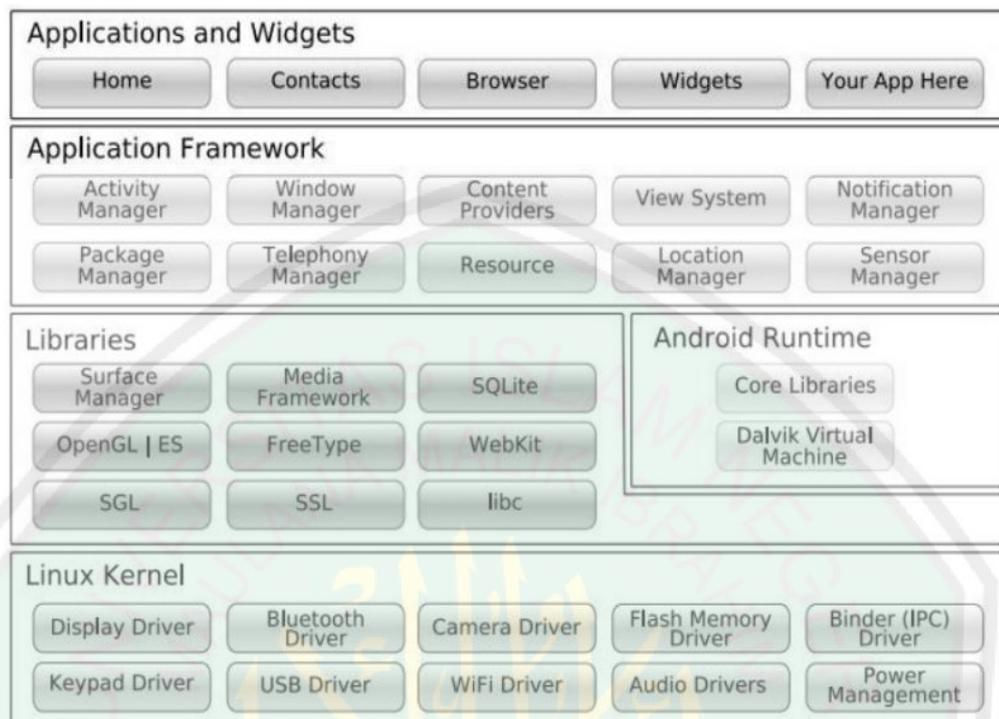
Google (2014) memberikan pengertian bahwa android adalah sistem operasi yang terdapat pada piranti bergerak yang berbasis Linux dan bersifat terbuka (open source). Sehingga, para pengembang sangat terbuka untuk menciptakan aplikasi mereka sendiri agar dapat digunakan oleh bermacam peranti bergerak.

Google membeli perusahaan Android Inc., yang merupakan sebuah perusahaan kecil berbasis pengembangan perangkat lunak untuk ponsel, Google membeli perusahaan tersebut pada tahun 2005 untuk memulai pengembangan pada platform Android. Tokoh utama pada Android Inc. meliputi Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Pada tanggal 5 November 2007, kelompok pemimpin industri bersama-sama membentuk *Open Handset Alliance* (OHA) yang diciptakan untuk mengembangkan standart terbuka bagi perangkat mobile.

Sistem operasi Android memiliki beberapa versi yang sudah diluncurkan ke pasar. Berikut merupakan berbagai versi Android. Android versi 1.1, Android versi 1.5 (Cupcake), Android versi 1.6 (Donut), Android versi 2.0/2.1 (Eclair), Android versi 2.2 (Froyo: Frozen Yoghurt), Android versi 2.3 (Gingerbread), Android versi 3.0/3.1 (Honeycomb), Android versi 4.0 (ICS :Ice Cream Sandwich), Android versi 4.1/4.2/4.3 (Jelly Bean), dan yang terbaru Android versi 4.4 (KitKat) (Google, 2014). Android merupakan sebuah perangkat lunak untuk

perangkat mobile, yang mana terdiri dari sebuah sistem operasi. Rancangan arsitektur android (Safaat, 2011) terlihat pada gambar 2.1:

- a. *Applications*: Android akan menggabungkan dengan aplikasi inti seperti klien email, program SMS, kalender, peta, browser, kontak, dan lain-lain.
- b. *Application Framework*: Dengan menyediakan sebuah platform pengembangan yang terbuka, pengembang Android menawarkan kemampuan untuk membangun aplikasi yang akan dijalankan di sistem android.
- c. *Libraries*: Android termasuk satu set pustaka C/C++ yang digunakan oleh berbagai komponen sistem Android.
- d. *Android Runtime*: Android termasuk satu set perpustakaan inti yang menyediakan sebagian besar fungsi yang tersedia di perpustakaan inti dari bahasa pemrograman Java.
- e. *Linux Kernel*: Android bergantung pada Linux versi 2.6 untuk layanan sistem inti seperti keamanan, manajemen memori, manajemen proses, network stack, dan model pengemudi. Kernel juga bertindak sebagai lapisan abstraksi antara hardware dan software.



Gambar 2.1 Arsitektur Android

2.2 Permainan (*Game*)

Menurut kamus bahasa Indonesia *game* memiliki arti adalah permainan. Permainan adalah kegiatan yang didalamnya terdapat beberapa peraturan. Dalam permainan terdapat sebuah konflik buatan yang dibuat untuk pemain sehingga dapat berinteraksi dengan sistem dan konflik tersebut. Sebuah *game* memiliki sebuah tujuan untuk menyelesaikan masalah/konflik yang dibuat. *Game* adalah lingkungan pelatihan yang baik bagi dunia nyata dalam organisasi yang menuntut pemecahan masalah secara kolaborasi.

Dalam Al-Qur'an surat Muhammad ayat 36 disebutkan bahwa bahwa dunia ini adalah tempat permainan dan senda gurau.

إِنَّمَا الْحَيَاةُ الدُّنْيَا لَعِبٌ وَلَهُوَ جَعْلٌ وَإِنْ تَوَمَّنُوا وَتَتَّقُوا يُؤْتِكُمْ أَجْرَكُمْ وَلَا يَسْأَلْكُمْ أَمْوَالَكُمْ ﴿٣٦﴾

Artinya:

“*Sesungguhnya kehidupan dunia hanyalah permainan dan senda gurau. Dan jika kamu beriman dan bertakwa, Allah akan memberikan pahala kepadamu dan Dia tidak akan meminta harta-hartamu.*”

Ibnu Katsir dalam menafsirkan Q.S. Muhammad ayat 36 berkata:

“Allah SWT berfirman sebagai penghinaan terhadap urusan dunia dan peremehan terhadapnya: *Sesungguhnya kehidupan dunia hanyalah permainan dan senda gurau*. Maksudnya, demikianlah hasilnya, kecuali jika dimaksudkan untuk beribadah kepada Allah SWT. Oleh karena itu, Dia berfirman: *Dan jika kamu beriman dan bertakwa, Allah akan memberikan pahala kepadamu dan Dia tidak akan meminta-minta hartamu*. Maksudnya, Dia tidak pernah butuh kepada kalian, Dia tidak meminta sesuatu pun dari kalian.”

Dunia adalah tempat bermain, berarti pemainnya itu sendiri adalah semua makhluk yang hidup di bumi. Kehidupan yang ada diatas bumi ini merupakan sebuah permainan. Semua orang maupun makhluk hidup lainnya memiliki peran dalam permainan yang berlangsung di bumi ini. Semua orang menyukai permainan sehingga dalam segala urusan di bumi diperlakukan seperti permainan. Oleh sebab itu, terdapat aturan bermain untuk menjalankannya. Hal ini termasuk juga dalam pembelajaran bahasa arab. Agar pembelajaran bahasa arab berlangsung menyenangkan maka diperlukan sebuah permainan untuk mempelajari bahasa arab itu.

Orang-orang telah bermain *game* di komputer selama komputer telah ada, dan beragam permainan telah ditemukan. National Writing Project (2011)

memberikan penjelasan dalam bukunya Jane McGonigal “Reality is Broken”, mengatakan bahwa permainan yang paling memiliki empat atribut:

- Tujuan (*goal*): permainan jelas mendefinisikan tujuan bagi para pemain untuk mencapainya. Tujuan yang menantang sangat penting, namun tujuan tersebut dapat dicapai.
- Aturan (*rules*): permainan memiliki aturan yang harus diikuti oleh pemain. Aturan sering membuat tujuan yang dicapai menjadi sulit, sehingga mengharuskan pemain menjadi kreatif.
- Arus balik (*feedback*): sebuah permainan harus memberitahu apa yang harus dilakukan seorang pemain.
- Partisipasi sukarela (*Voluntary Participation*): Aspek permainan menyiratkan penerimaan pemain dari tujuan, aturan, dan sistem *feedback*.

2.2.1 Jenis Permainan (*Genre Games*)

Pengembang *game* tidak memulai mengkategorikan permainan yang mereka buat. Namun demikian, dari waktu ke waktu permainan telah dikelompokkan ke dalam kelas oleh orang yang berbeda dengan cara yang berbeda. Kategori yang diidentifikasi dalam bagian ini tidak dimaksudkan untuk menjadi resmi. Dengan sejalanannya waktu, permainan dikategorikan ke dalam beberapa jenis. Berikut adalah jenis permainan yang dapat diklasifikasikan menjadi beberapa jenis (Rogers, 2012) antara lain adalah:

1. *Skill or Action Games*

Action Games biasanya harus menggunakan beberapa keterampilan (misalnya, melompat per barel pada waktu yang tepat, menembak target yang bergerak)

untuk menyelesaikannya. Contohnya seperti berikut: *Maze Games*, *Shooters*, *one-on-one fighting gamer*, *one-on-many fighting*, *First-person shooters (FPS)*.

2. *Strategy Games*

Permainan ini mengharuskan pemain untuk melibatkan strategi bermain, taktik dan logika.

3. *Adventure or Storytelling Games*

Jenis permainan ini dibangun dengan alur cerita yang banyak dengan pengembangan karakter yang baik dan cerita yang mendefinisikan tujuan pemain dalam bermain permainan.

4. *Simulation Games*

Game simulasi menggambarkan situasi yang nyata seperti pemain dapat mengoperasikan kendaraan. *Game* ini meniru fisik situasi yang nyata dan cukup bagus digunakan untuk intruksi walaupun hanya dengan bermain *game*. Contoh *game* tipe ini adalah *sports simulators*, *flight simulators*, *driving simulators*, *submarine simulators*, dan *life simulators*.

5. *Puzzle Games*

Jenis *game* ini dibangun untuk memecahkan masalah tertentu. Biasanya tantangan *game* ini menggunakan logika yang dibatasi oleh waktu. Contoh jenis *game* ini seperti Sudoku, crosswords, minesweeper.

6. *Augmented-Reality Games*

Jenis *game* ini masih terbilang baru dalam dunia *game*. *Game* ini dapat memberikan kehidupan nyata menjadi lebih mudah. Contohnya seperti *game* Jetset (sebuah permainan yang mensimulasikan keamanan garis di bandara).

2.2.2 Komponen Permainan (*Component Games*)

Dalam pembuatan *game* terdapat beberapa komponen penting yang harus diperhatikan. Berikut komponen dalam membangun *game* (Rogers, 2012) itu sendiri yaitu:

1. *Opening (Splash) Screen* (Pembukaan)

Splash Screen ini opsional dalam pembuatan *game* tapi lebih baik di masukkan kedalam *game* yang dibuat. Hal ini dilakukan agar pengguna mengetahui bahwa *game* ini berjalan seperti seharusnya.

2. *Menu Screen* (Tampilan Menu)

Sejak *game* ini berjalan, biasanya pengguna akan mencari menu untuk mematikan/menyalakan suara atau mencari cara bermain *game* yang dimainkan.

3. *Music* (Musik)

Komponen ini merupakan komponen yang penting karena dapat membangun suasana pengguna memainkan *game* yang dibuat.

4. *Sound Effects* (Efek Suara)

Sound effects dapat membuat *game* menjadi lebih menyenangkan. Ketika terdapat dua objek bertubrukan pengguna dapat mendengar suara seperti *clang*, *thud*, atau *boing*.

5. *Time* (Waktu)

Kebanyakan *game* menggunakan waktu untuk menyelesaikan *game* seperti menyelesaikan *game* teka-teki berdasarkan waktu yang dibutuhkan untuk memecahkan teka-teki.

6. *Lives* (Nyawa)

Game harus mempunyai tantangan agar menjadi menyenangkan, sehingga pemain dapat gagal dalam bermain. Membunuh *player* adalah cara yang pantas dalam memberikan kegagalan. Beberapa *game* memberikan pemain beberapa nyawa per sesi dan ada juga yang hanya memiliki nyawa.

7. *Obstacles* (Rintangannya)

Setiap *game* memiliki *obstacle* yang berbeda-beda. Untuk mencapai tujuan pemain harus bisa melewati *obstacle* yang diberikan oleh *game*.

8. *Levels* (Tingkatan)

Dalam sebuah *game* sangat penting untuk memberikan berbagai tantangan, sehingga pemain dapat mulai bermain dengan tantangan yang mudah dan secara bertahap berjalan menuju tantangan yang lebih tinggi. Tantangan tergantung dengan tingkatan yang dihadapi pemain. Dengan tingkatan dapat memberikan keterampilan pemain dalam bermain.

9. *Adversaries* (Lawan/Musuh)

Lawan dalam permainan kadang-kadang disebut sebagai entitas. Karakter ini adalah penjahat yang harus dihadapi pemain. Lawan berbeda dengan rintangan dalam mengambil tindakan terhadap pemain. Hambatan sedikit lebih pasif ketimbang lawan dalam mengambil tindakan.

10. *Player* (Pemain)

Pemain adalah komponen paling penting dari setiap *game*. Intinya adalah untuk menjaga pemain untuk terlibat dan tertarik sehingga pemain akan terus bermain *game* yang dimainkan.

11. *Scenes* (Adegan)

Setiap adegan memiliki grafis latar belakang yang tidak berubah banyak (meskipun titik pandang pemain mungkin berubah).

2.3 Permainan Kecerdasan Buatan (*Game AI*)

Manusia dikarunia Allah sebuah kecerdasan untuk berpikir dan menggunakan akal nya. Hal ini sudah dijelaskan dalam firman Allah pada surat al-Baqarah ayat 266 yang berbunyi:

أَيُّودٌ أَحَدَكُمُ أَنْ تَكُونَ لَهُ جَنَّةٌ مِّنْ نَّخِيلٍ وَأَعْنَابٍ تَجْرِي مِنْ تَحْتِهَا الْأَنْهَارُ لَهُ فِيهَا مِنْ كُلِّ الثَّمَرَاتِ وَأَصَابَهُ الْكِبَرُ وَلَهُ ذُرِّيَّةٌ ضِعْفَاءُ فَصَابَهَا إِعْصَارٌ فِيهِ نَارٌ فَاحْتَرَقَتْ كَذَلِكَ يُبَيِّنُ اللَّهُ لَكُمْ الْآيَاتِ لَعَلَّكُمْ تَتَفَكَّرُونَ ﴿٢٦٦﴾

Artinya:

“Apakah ada salah seorang di antaramu yang ingin mempunyai kebun kurma dan anggur yang mengalir di bawahnya sungai-sungai; dia mempunyai dalam kebun itu segala macam buah-buahan, kemudian datanglah masa tua pada orang itu sedang dia mempunyai keturunan yang masih kecil-kecil. Maka kebun itu ditiup angin keras yang mengandung api, lalu terbakarlah. Demikianlah Allah menerangkan ayat-ayat-Nya kepada kamu supaya kamu memikirkannya.”

Penjelasan dari ayat ini bahwasanya Allah telah menjelaskan kepada para hambanya tentang tanda-tanda kekuasaannya yang syar’i dan tanda-tanda kekuasaanNya dalam alam semesta ini, dan ini semua telah di jelaskan didalam kitabnya dengan sesempurna mungkin. Anjuran untuk memikirkan (dari tanda-tanda kekuasaan Allah), dan inilah tujuan yang paling utama dalam ayat ini (لَعَلَّكُمْ تَتَفَكَّرُونَ): “supaya kamu memikirkannya”. Dari penjelasan inilah Allah memerintahkan umatnya untuk berpikir. Dengan berpikir, masalah yang dihadapi

dapat diselesaikan. Untuk menyelesaikan masalah maka dibutuhkanlah suatu kecerdasan.

Kecerdasan manusia dapat diberikan pada komputer yang biasa disebut dengan kecerdasan buatan. Kecerdasan buatan adalah cara manusia memberikan suatu aturan pengetahuan terhadap komputer sebagai alat bantu untuk menyelesaikan masalah. Kecerdasan buatan juga digunakan untuk mensimulasikan kecerdasan manusia seperti perilaku cerdas atau himpunan bagian dari perilaku cerdas terutama dalam *Non Player Character* (NPC). Permainan AI yang digunakan dalam berbagai bidang cukup berbeda dalam permainan. Yang paling jelas adalah dalam pengendalian setiap NPC dalam permainan.

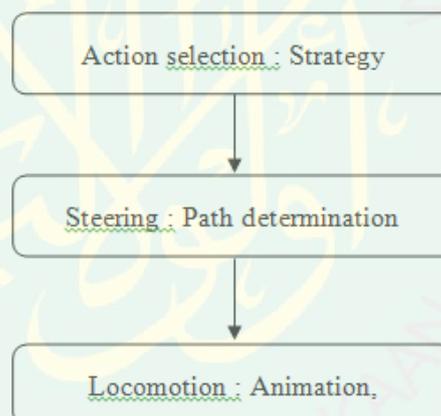
Permainan AI juga terlibat dengan permainan dinamis yang menyesuaikan kesulitan dalam permainan berdasarkan kemampuan pemain. Peranan AI dalam hal interaksi pemain dengan permainan adalah pada penggunaan interaksi yang bersifat alami yaitu yang biasa digunakan manusia berinteraksi dengan sesama manusia. Contoh media interaksi tersebut adalah penglihatan (*vision*), suara (*voice*), ucapan (*speech*), dan gerakan anggota badan (*gesture*).

2.4 Non Player Character (NPC)

Autonomous character adalah jenis otonomous agent yang ditujukan untuk penggunaan komputer animasi dan media interaktif seperti *games* dan *virtual reality* (Arif, 2010). Agen ini mewakili tokoh dalam cerita atau permainan dan memiliki kemampuan untuk improvisasi tindakan mereka. Ini adalah kebalikan

dari seorang tokoh dalam sebuah film animasi, yang tindakannya ditulis di muka, dan untuk “avatar” dalam sebuah permainan atau virtual reality, tindakan yang diarahkan secara real time oleh pemain. Dalam permainan, karakter otonom biasanya disebut *Non Player Character* (NPC).

Perilaku karakter yang otonom dapat lebih baik dipahami dengan membaginya menjadi beberapa lapisan. Lapisan ini dimaksudkan hanya untuk kejelasan dan kekhususan dalam diskusi yang akan mengikuti. Gambar 2.2 menunjukkan sebuah divisi gerak perilaku otonom hirarki karakter menjadi tiga lapisan: seleksi tindakan, steering, dan penggerak.



Gambar 2.2 Hirarki Gerak Perilaku

2.5 Jaringan Saraf Tiruan (JST)

Jaringan saraf tiruan pertama kali didesain oleh Warren Mc-Culloch dan Walter Pitts. Jaringan saraf tiruan merupakan bagian ilmu dari kecerdasan buatan (*Artificial Intelligence*). JST memiliki cara kerja seperti otak makhluk hidup yaitu sel saraf (*neuron*). *Neuron* adalah satuan unit pemroses terkecil pada otak. Aryulina dkk (2004) secara garis besar *neuron* dibagi tiga yaitu badan sel (*cell*

body), dendrit (*dendrites*), dan akson (*axon*). **Badan sel** yaitu bagian utama *neuron* yang memiliki inti sel. **Dendrit** yaitu penjurukan ke arah luar badan sel yang berperan sebagai penerima sinyal untuk diantarkan inti sel ke badan sel. **Akson** merupakan penjurukan badan sel yang berfungsi untuk mengirimkan sinyal-sinyal dari badan sel ke akson pada badan sel *neuron* yang lain.

Jaringan saraf tiruan didefinisikan sebagai suatu sistem pemrosesan informasi yang mempunyai karakteristik menyerupai jaringan saraf manusia (Hermawan, 2006). Jaringan saraf tiruan tercipta sebagai suatu generalisasi model matematis dari pemahaman manusia yang didasarkan atas asumsi sebagai berikut:

1. Pemrosesan terjadi pada *neuron*.
2. Sinyal-sinyal dilewatkan antara *neuron* yang saling berhubungan
3. Setiap sambungan penghubung memiliki bobot yang bersesuaian.
4. Setiap sel saraf akan menerapkan fungsi aktivasi terhadap jumlah bobot yang masuk untuk menentukan hasil keluarannya.

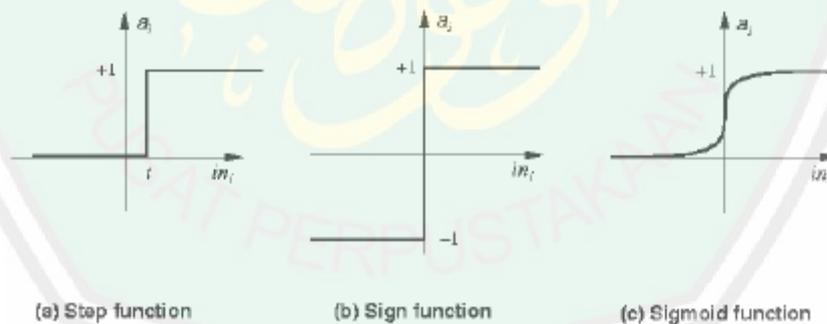
Jaringan saraf terdiri dari beberapa *neuron* dan memiliki hubungan antara *neuron-neuron* tersebut. Beberapa *neuron* akan mentransformasikan informasi yang diterimanya melalui sambungan keluaran menuju *neuron-neuron* yang lain. Dengan kata lain, *neuron* adalah sebuah unit pemroses informasi yang merupakan dasar operasi jaringan saraf tiruan. McCulloch-Pitts menemukan bahwa dengan mengkombinasikan banyak *neuron* sederhana sehingga menjadi sebuah sistem saraf merupakan sumber peningkatan tenaga komputasional (Puspitaningrum, 2006:21). Struktur jaringan saraf tiruan dan struktur *neuron* sederhana digambarkan pada gambar 2.3.

- Threshold : nilai ambang internal dari node θ adalah besarnya offset yang mempengaruhi aktivasi dari output node y :

$$y = \sum_{i=1}^n X_i W_i - \theta$$

- Fungsi aktivasi : merupakan operasi matematik yang dikenakan pada sinyal output y .

Mengaktifkan jaringan saraf tiruan berarti mengaktifkan setiap neuron yang dipakai pada jaringan tersebut. banyak fungsi aktivasi yang digunakan sebagai pengaktif, seperti fungsi-fungsi gonimetri dan hiperboliknya, fungsi unit step, impulse, sigmoid, dan lain sebagainya seperti pada gambar 2.5, tetapi yang lazim digunakan adalah fungsi sigmoid, karena dianggap lebih mendekati kinerja sinyal pada otak.



Gambar 2.5 Fungsi pengaktif

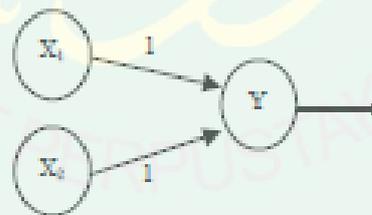
- Step: $y = f(x) = \begin{cases} 0 & \text{jika } x < 0 \\ 1 & \text{jika } x \geq 0 \end{cases}$
- Sign: $y = f(x) = \begin{cases} -1 & \text{jika } x < 0 \\ +1 & \text{jika } x \geq 0 \end{cases}$
- Sigmoid: $y = f(x) = \frac{1}{1+e^{-ax}}$ untuk $0 \leq f(x) \leq 1$

Beberapa contoh jaringan sederhana AND, OR, dan NOT dapat dijelaskan seperti dibawah ini.

Tabel 2.1 Operator AND

X1	X2	Y
1	1	1
1	0	0
0	1	0
0	0	0

Dari tabel 2.1 ada dua input (X1 dan X2) dan satu output (Y) dengan bobot yang sesuai untuk w_1 adalah 1 dan w_2 adalah 1 sehingga diperoleh nilai threshold yang sesuai dengan tabel adalah 2. Arsitektur jaringan sederhana untuk operator AND terlihat pada gambar 2.6.



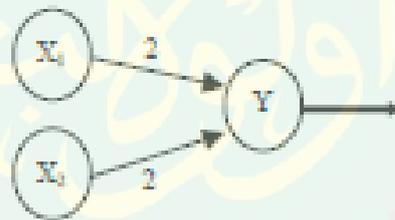
Threshold(Y) = 2

Gambar 2.6 Arsitektur jaringan sederhana operasi AND

Tabel 2.2 Operator OR

X1	X2	Y
1	1	1
1	0	0
0	1	0
0	0	0

Dari tabel 2.2 ada dua input (X1 dan X2) dan satu output (Y) dengan bobot yang sesuai untuk w_1 adalah 2 dan w_2 adalah 2 sehingga diperoleh nilai threshold yang sesuai dengan tabel adalah 2. Arsitektur jaringan sederhana untuk operator OR terlihat pada gambar 2.7.

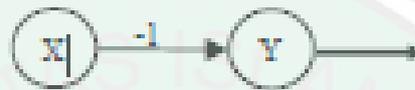


Threshold(Y) = 2

Gambar 2.7 Arsitektur jaringan sederhana operasi OR**Tabel 2.3** Operator NOT

X	Y
0	1
1	0

Dari tabel 2.3 ada satu input (X) dan satu output (Y) dengan bobot yang sesuai untuk w adalah -1 sehingga diperoleh nilai threshold yang sesuai dengan tabel adalah -0.5. Arsitektur jaringan sederhana untuk operator NOT terlihat pada gambar 2.8.

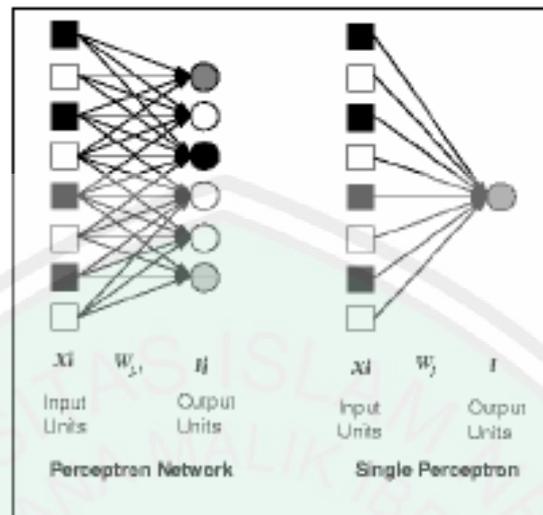


$$\text{Threshold}(Y) = -0.5$$

Gambar 2.8 Arsitektur jaringan sederhana operasi NOT

2.6 Perceptron

Single layer perceptron terdiri dari beberapa unit neuron seperti pada gambar 2.9 yang terhubung dan mempunyai beberapa masukan serta sebuah atau beberapa keluaran. Single perceptron sering disebut juga dengan Perceptron. Perceptron paling tipikal terdiri dari sebuah lapisan input yang terhubung oleh jalur-jalur, dengan bobot-bobot tetap, ke neuron-neuron asosiator (Puspitaningrum, 2006:22). Perceptron menghitung jumlah nilai perkalian penimbang dan masukan dari parameter permasalahan yang kemudian dibandingkan dengan nilai threshold. Bila nilai keluaran lebih besar dari threshold maka keluarannya adalah satu, sebaliknya nol.



Gambar 2.9 Single perceptron

Pelatihan pada perceptron dilakukan dengan cara mengajarnya dengan contoh-contoh kasus/pola sampai jaringan saraf tiruan berhasil mengenali pola tersebut. Setiap pola yang diberikan merupakan pasangan pola masukan dan pola yang diinginkan. Perceptron melakukan penjumlahan berbobot terhadap tiap-tiap masukannya dan menggunakan fungsi ambang untuk menghitung keluarannya/output. Keluaran ini kemudian dibandingkan dengan keluaran yang diinginkan/target, perbedaan yang dihasilkan dari perbandingan ini akan digunakan untuk merubah bobot-bobot yang ada pada jaringan. Setiap kali output tidak sesuai dengan target maka nilai bobotnya akan terus di-update sehingga sesuai dengan kebutuhan yang dilakukan dengan membandingkan output dari jaringan dengan targetnya. Hal ini akan terus dilakukan sampai tidak ada lagi bobot yang berubah untuk setiap pasangan latihan sensor dan target. Demikian dilakukan berulang-ulang sehingga dihasilkan keluaran yang sesuai dengan hasil yang diinginkan/target.

Kesesuaian antara keluaran dengan target yang diinginkan menunjukkan pencapaian kerja yang sedang dilatihkan dan dinyatakan dengan besarnya galat yang diperbolehkan agar keluaran hasil pengolahan jaringan dapat dianggap sama dengan pola yang diinginkan.

Langkah-langkah algoritma pelatihan Perceptron adalah sebagai berikut:

- Inisialisasi semua bobot dan bias (umumnya $W_{ij} = b = 0$). Set laju pembelajaran α . ($0 < \alpha \leq 1$) (untuk penyederhanaan set $\alpha = 1$). Kemudian set epoch = 0.

- Untuk setiap pasangan pembelajaran s-t kerjakan:

- Set aktivasi unit input $x_i = s_i$

- Hitung respons untuk unit output:

$$y = b + \sum_{i=1}^n X_i W_i \quad (1)$$

- Masukkan kedalam aktifasi:

$$f(x) = \begin{cases} 1, & \text{jika } y > \theta \\ 0, & \text{jika } -\theta \leq y \leq \theta \\ -1, & \text{jika } y < -\theta \end{cases} \quad (2)$$

- Bandingkan nilai output jaringan y dengan target. Jika $y \neq t$,

lakukan perubahan bobot dan bias dengan cara berikut:

$$w_i (\text{baru}) = w_i (\text{lama}) + \alpha t x_i \quad (3)$$

$$b (\text{baru}) = b (\text{lama}) + \alpha t \quad (4)$$

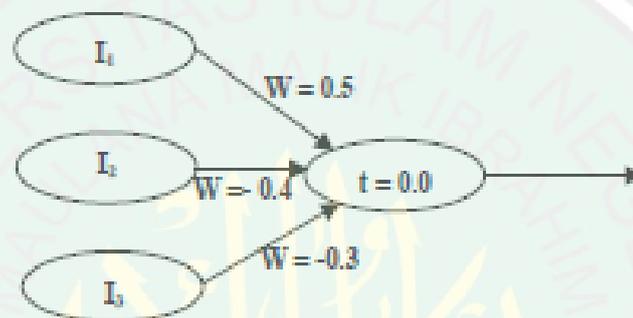
jika $y = t$, tidak ada perubahan bobot dan bias

$$w_i (\text{baru}) = w_i (\text{lama}) \quad (5)$$

$$b (\text{baru}) = b (\text{lama}) \quad (6)$$

- Lakukan iterasi terus menerus hingga semua pola memiliki output jaringan yang sama dengan targetnya dan iterasi dihentikan

Pada gambar 2.10 merupakan contoh kasus operator AND. Mula-mula dilakukan random pada bobot(W) kemudian dilakukan perhitungan perkalian antara input (I) dan bobot (W) seperti pada tabel 2.4.



Gambar 2.10 Contoh kasus perceptron

Tabel 2.4 Perhitungan pelatihan perceptron

I ₁	I ₂	I ₃	Penjumlahan	Output
0	0	1	$(0*0.5) + (0*-0.4) + (1*-0.3) = -0.3$	0
0	1	1	$(0*0.5) + (1*-0.4) + (1*-0.3) = -0.7$	0
1	0	1	$(1*0.5) + (0*-0.4) + (1*-0.3) = 0.2$	1
1	1	1	$(1*0.5) + (1*-0.4) + (1*-0.3) = -0.2$	0

Output diperoleh dengan melakukan fungsi aktivasi dalam kasus AND dilakukan dengan unit step yaitu bila kurang dari nol output = 0, bila lebih output = 1. Error diperoleh apabila terjadi perbedaan antara target dengan output jaringan. Jika error tidak sama dengan nol maka bobot (W) diperbaiki.

BAB III

ANALISA DAN PERANCANGAN SISTEM

3.1 Analisa dan Perancangan Sistem

Game yang dibangun adalah *game single player* yang ber-genre *sidescrolling*. Terdapat beberapa karakter pada *game* ini yang di desain agar *game* sangat menarik dimainkan. Karakter tersebut berupa pemain utama dan beberapa karakter NPC (*Non Player Character*). NPC akan diberikan sebuah perilaku yang cerdas agar *game* yang dimainkan jadi lebih menarik. Perilaku NPC dapat berubah berdasarkan dengan parameter yang telah diatur yaitu jumlah poin dan jumlah mufradat. Dalam sebuah *game*, terdapat beberapa pengembangan *game*, diantaranya:

3.1.1 Keterangan Umum *Game*

Game ini ber-genre *sidescrolling*, *game* yang dimainkan hanya ke arah samping kiri dan kanan seperti Mario Bros. Terdapat tiga *scene* pada *game* ini, setiap *scene* hanya dibedakan dari segi *background*, jenis musuh, jumlah poin dan jumlah mufradat. Dalam *game* ini pemain harus mengumpulkan beberapa koin dan mufradat dan juga harus melewati beberapa musuh yang mengejar si pemain. Seorang pemain tidak dapat menyerang ataupun menembak musuh karena pada *game* ini tidak ingin ada unsur kekerasan di dalamnya. Di mana sesuai dengan sasaran bahwa penikmat *game* ini adalah anak-anak usia sekolah dasar.

Sedangkan untuk objek penelitiannya, yaitu perilaku NPC tersebut. Pada *game* ini NPC dibagi dua yaitu NPC jahat dan NPC penolong. NPC jahat memiliki perilaku yang mengejar pemain dan NPC penolong memiliki perilaku memberi poin dan mufradat jika pemain kekurangan poin dan mufradat.

3.1.2 Penampilan *Game*

Secara umum, *game* ini dibangun dengan grafis 2Dimensi. Semua objek yang ada pada *game* ini dibangun dengan objek 2Dimensi seperti *background*, karakter, koin, dan juga peta (*map*).

3.1.3 Deskripsi Karakter

Terdapat beberapa karakter yang dibangun pada *game* ini, yaitu:

3.1.3.1 Karakter Utama

Karakter ini yang memerankan sebagai seorang santri yang diberi misi untuk mengumpulkan mufradat. Untuk menyelesaikan misi tersebut, pemain harus mengumpulkan semua mufradat jika sudah selesai maka dapat untuk pindah ke *scene* selanjutnya.

3.1.3.2 NPC Jahat

Karakter ini berupa jin jahat yang memiliki tugas untuk mengejar pemain yang sedang mengumpulkan poin. Perilaku karakter ini bergantung dai AI yang sudah dibuat. Karakter ini juga dapat berubah menjadi besar jika pemain dapat mengumpulkan mufradat dengan banyak.

3.1.3.3 NPC Penolong

Karakter ini berupa ustadz yang memiliki tugas untuk menolong pemain yang kekurangan poin dan mufradat. Karakter ini hanya berdiam di suatu tempat menunggu pemain datang menghamipirinya.

3.1.3.4 Musuh Pengganggu

Karakter musuh ini dibuat dengan bentuk binatang yaitu ular. Karakter ini tidak memiliki tugas khusus dan AI. Tugasnya hanya jalan-jalan disekitar tempat yang sudah ditentukan. Walaupun tidak memiliki AI tetapi pemain harus menghindari karakter ini karena dapat merugikan pemain yaitu dapat mengurangi poin yang sudah didapat.

3.2 Perancangan Kecerdasan Buatan Perilaku NPC

Kecerdasan buatan dalam penelitian ini dikonsentrasikan pada perilaku NPC. Untuk memberikan perilaku pada NPC dibutuhkan beberapa data yang digunakan. Dalam mengolah data ini dibutuhkan sebuah variabel yang digunakan sebagai *input*. Variabel yang digunakan jumlah poin dan jumlah mufradat. Dari permainan ini hanya menyediakan tiga tingkatan/level. Variabel yang digunakan setiap tingkatan akan selalu berubah. Pada level pertama koin penuh yang disediakan berjumlah 20 koin sedangkan jumlah mufradat yang disediakan berjumlah 8 mufradat. Untuk level kedua jumlah poin semakin meningkat menjadi 30 koin sedangkan jumlah mufradat menjadi 12 mufradat. Pada level terakhir, jumlah koin menjadi 40 sedangkan jumlah mufradat menjadi 15 mufradat. 1 koin

dihargai dengan 10 poin. Semakin banyak koin yang didapat maka semakin banyak poin yang didapat.

Dari variabel yang sudah ditentukan tadi, maka dapat dibuat aturan untuk input pada perhitungan JST Perceptron. Berikut aturan yang digunakan pada setiap level:

- Level 1: $f(x) = \begin{cases} 0, & \text{koin} < 10 \\ 1, & \text{koin} \geq 10 \end{cases}$ dan $f(x) = \begin{cases} 0, & \text{mufradat} < 5 \\ 1, & \text{mufradat} \geq 5 \end{cases}$
- Level 2: $f(x) = \begin{cases} 0, & \text{koin} < 18 \\ 1, & \text{koin} \geq 18 \end{cases}$ dan $f(x) = \begin{cases} 0, & \text{mufradat} < 8 \\ 1, & \text{mufradat} \geq 8 \end{cases}$
- Level 3: $f(x) = \begin{cases} 0, & \text{koin} < 26 \\ 1, & \text{koin} \geq 26 \end{cases}$ dan $f(x) = \begin{cases} 0, & \text{mufradat} < 12 \\ 1, & \text{mufradat} \geq 12 \end{cases}$

Fungsi target yang digunakan yaitu menggunakan target berbentuk bipolar.

Aturan tersebut berlaku pada NPC musuh dan NPC penolong. Berikut hasil output dari input yang digunakan:

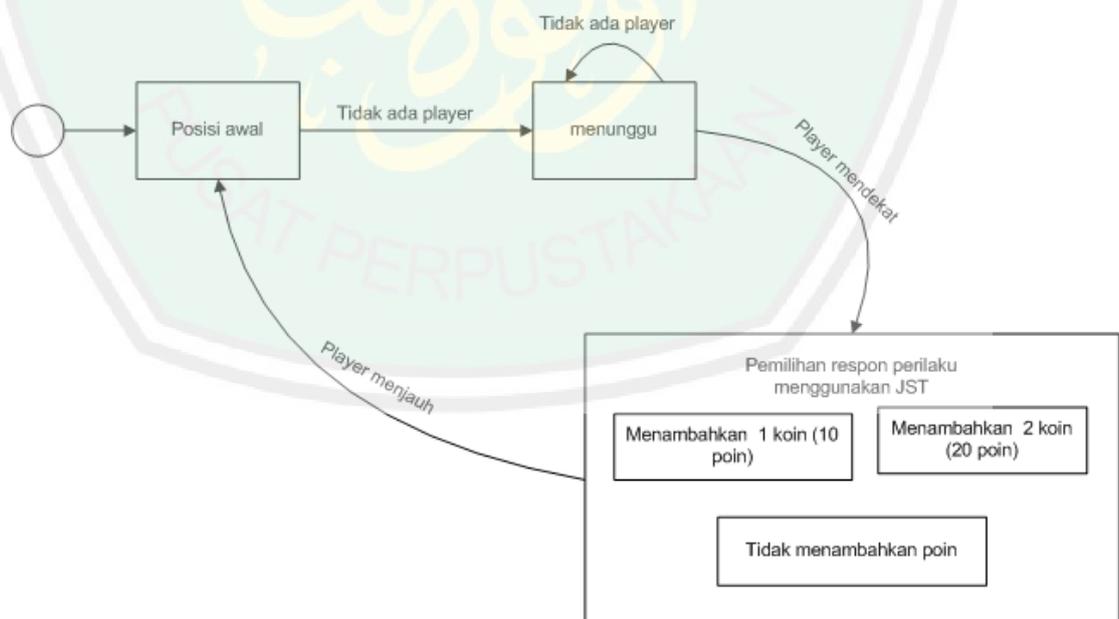
- NPC musuh : output 1 maka mengejar dengan cepat dan bertambah besar, output 0 maka mengejar dengan lambat, output -1 maka mengejar dengan sangat lambat
- NPC penolong: output 1 maka tidak menambahkan poin dan mufradat, output 0 maka menambahkan 1 koin (10 poin), output -1 maka menambahkan 2 koin (20 poin).

Pada *game* ini pemain memainkan *game Arabic Hunter* dengan misi mengumpulkan mufradat yang sudah disediakan. Pemain harus menghadapi tantangan yaitu melewati musuh/NPC jahat yang dapat mengejar pemain, jika NPC dapat mengenai pemain maka nilai poin akan berkurang minus sepuluh (-

10). Setelah mufradat berhasil di kumpulkan maka permainan dapat diselesaikan dan melanjutkan ke level selanjutnya. Alur dari cerita *game* dibuat dalam bentuk *Finite State Machine*. Berikut *Finite State Machine* (FSM) pada NPC:

3.2.1 FSM NPC Penolong

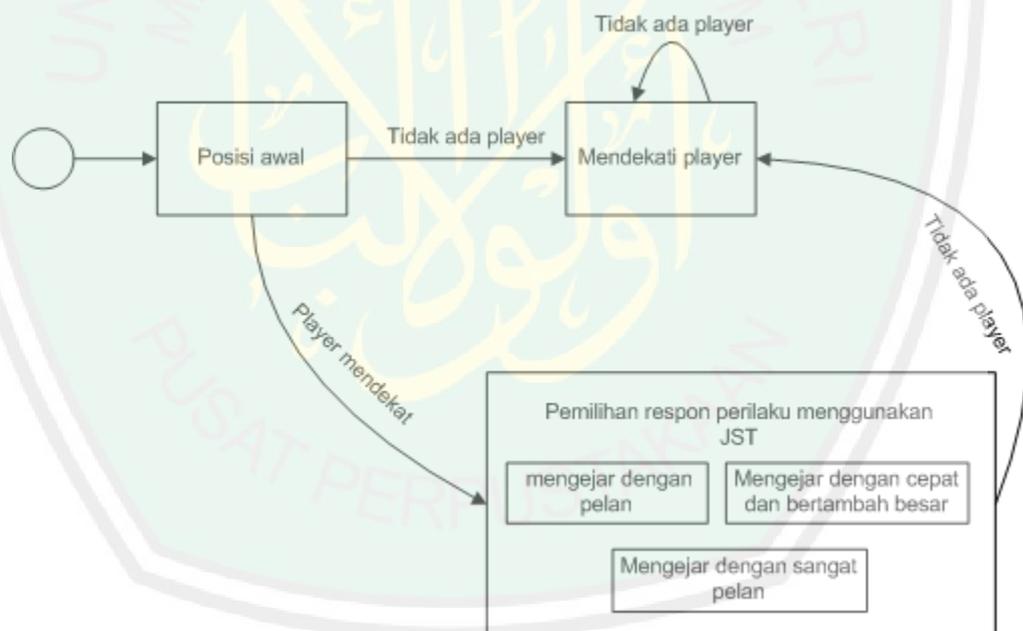
Saat pemain *start* memainkan *game* NPC ini akan berada pada posisinya menunggu pemain datang. NPC penolong akan tetap menunggu hingga pemain mendekat, jika pemain mendekat maka akan dilakukan pemilihan respon perilaku. Ada tiga respon perilaku yang dimiliki, yaitu menambahkan 2 koin (20 poin), menambahkan 1 koin (10 poin), dan tidak menambahkan koin. Pemberian perilaku berdasarkan aturan yang sudah dibuat. Pada gambar 3.1 menggambarkan FSM NPC Penolong.



Gambar 3.1 FSM NPC Penolong

3.2.2 FSM NPC Jahat

Saat pemain *start* memainkan *game* NPC ini akan berada pada posisinya menunggu pemain datang. NPC jahat akan jalan-jalan menunggu hingga pemain mendekat, jika pemain mendekat maka akan dilakukan pemilihan respon perilaku. Ada tiga respon perilaku yang dimiliki, mengejar dengan cepat dan bertambah besar, mengejar dengan pelan, dan jalan-jalan atau tidak mengejar. Pemberian perilaku berdasarkan aturan yang sudah dibuat. Pada gambar 3.2 menggambarkan FSM NPC Jahat.



Gambar 3.2 FSM NPC Jahat

3.2.3 Perancangan Perceptron pada NPC

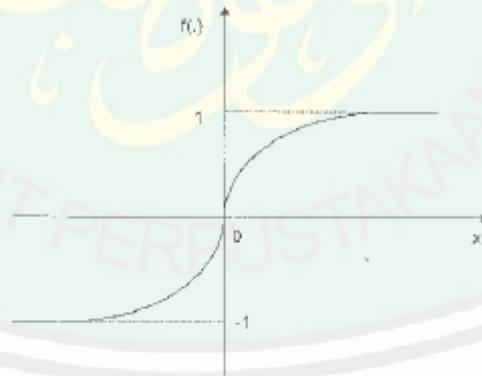
Penerapan algoritma terdapat pada NPC. Perceptron diterapkan untuk memberikan perilaku NPC yang sesuai dengan keadaan permainan.

A. Implementasi Perceptron

Algoritma perceptron diimplementasikan dengan alur sebagai berikut :

1. Melakukan permainan
2. Inisialisasi konten-konten yang ada dalam *game*
3. Menentukan input perceptron
4. Menentukan perilaku NPC menggunakan perceptron
5. NPC mengejar

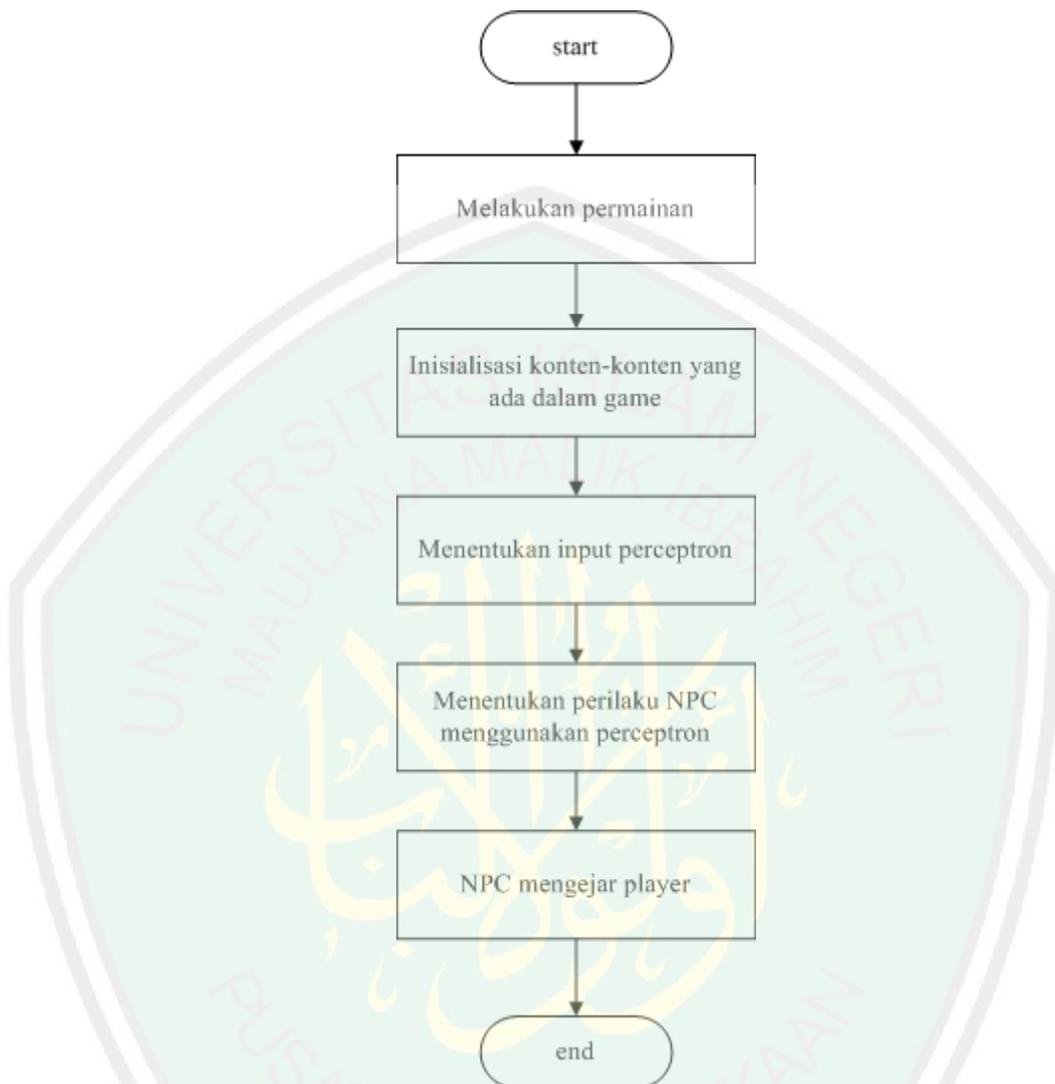
Input diambil dari jumlah skore dan mufradat. Terdapat dua input yang digunakan yaitu 0 dan 1. Input yang didapat akan dihitung dengan bobot dan bias yang sudah diketahui untuk mendapatkan nilai output. Fungsi aktivasi yang digunakan adalah fungsi bipolar. Adapun rumus untuk mencari nilai output adalah:



$$output = \begin{cases} 1, & \text{jika } y_{Out} > teta \\ 0, & \text{jika } -\theta \leq y_{Out} \leq teta \\ -1, & \text{jika } y_{Out} < -teta \end{cases}$$

aktivasi = (input1 * bobot1) + (input2 * bobot2);

yOut = bias + aktivasi;



Gambar 3.3 Blok Diagram Implementasi Perceptron

B. Simulasi Manual Perhitungan Perceptron

teta = 0.6, bias = -1, bobot1 = 1, bobot2 = 1;

aktivasi = (input1 * bobot1) + (input2 * bobot2);

yOut = bias + aktivasi;

Tabel 3.1 Perhitungan Manual Perceptron

Input1	Input2	yOut	Output
0	0	$((0 * 1) + (0 * 1)) + (-1) = -1$	-1
0	1	$((0 * 1) + (1 * 1)) + (-1) = 0$	0
1	0	$((1 * 1) + (0 * 1)) + (-1) = 0$	0
1	1	$((1 * 1) + (1 * 1)) + (-1) = 1$	1

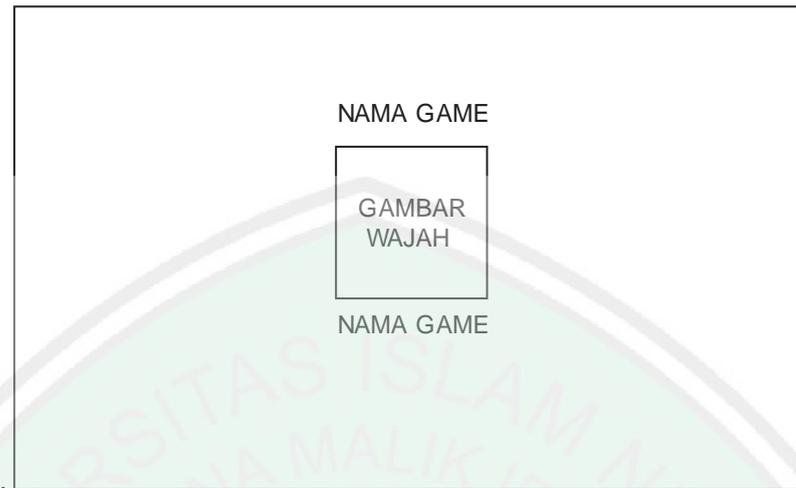
Dari hasil perhitungan manual tersebut maka dapat dihasilkan perilaku yang tepat sesuai dengan perancangan NPC pada sub bab 3.2.1 dan 3.2.2

3.3 Perancangan Aplikasi Game

Dalam pembuatan *game* diperlukan sebuah rancangan sebelum membangun *game* yang akan dibangun. Rancangan antarmuka digunakan untuk membuat tampilan dengan tujuan memberikan panduan dalam menggunakan *game* ini. Adapun perancangan antarmuka tersebut adalah sebagai berikut:

3.3.1 Perancangan Antarmuka *Splashscreen*

Pada saat *game* dijalankan maka akan muncul *splashscreen* yang berupa tampilan nama *game* dan gambar wajah karakter dari *game* tersebut yang terlihat pada gambar 3.4. Tampilan ini hanya sekilas setelah itu akan langsung pindah ke antarmuka menu *game*.



Gambar 3.4 Antarmuka *Splashscreen*

3.3.2 Perancangan Antarmuka Menu *Game*



Gambar 3.5 Antarmuka Menu *Game*

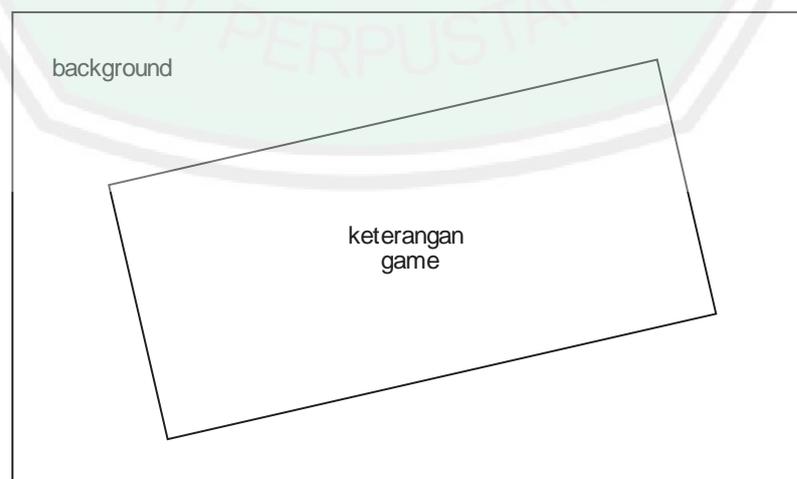
Terdapat beberapa tombol pada menu *game* seperti pada gambar 3.5 ketika tombol ditekan maka akan berbunyi suara efek. Pada antarmuka menu mengeluarkan musik untuk memberikan *game* lebih menarik. Keterangan fungsi tombol yang ada pada tabel 3.2 antara lain:

Tabel 3.2 Keterangan Fungsi Tombol Menu

No.	Menu	Keterangan
1.		Untuk memulai permainan
2.		Untuk mengaktifkan suara efek dan musik
3.		Untuk menonaktifkan suara music
4.		Untuk menampilkan keterangan tentang <i>game</i>
5.		Untuk keluar dari permainan

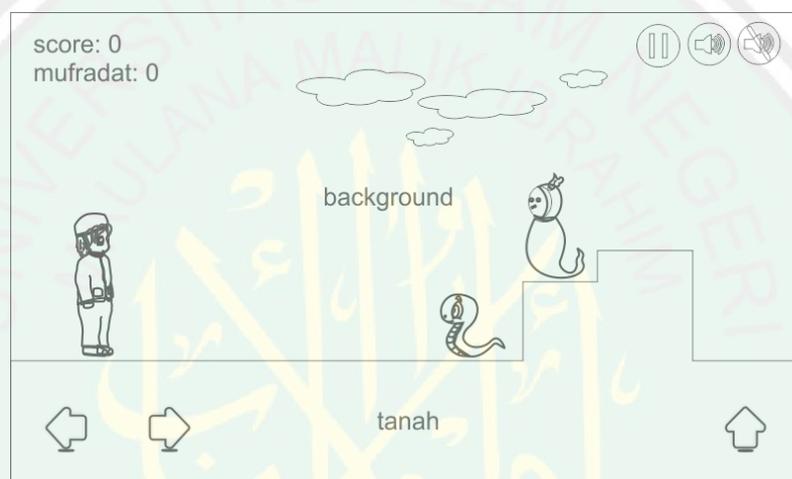
3.3.3 Perancangan Antarmuka Tentang *Game*

Pada gambar 3.6 merupakan perancangan yang menjelaskan tentang cerita dari *game* yang dimainkan dan bagaimana cara menyelesaikan *game* ini.

**Gambar 3.6** Antarmuka Tentang *Game*

3.3.4 Perancangan Antarmuka *Game*

Perancangan antarmuka *game* ini menggambarkan keadaan layar ketika pemain sudah pada posisi memulai permainan. Pada antarmuka *game* ini terdapat tiga skenario yang dibuat yaitu pagi hari, siang hari, dan malam hari. Terdapat beberapa tombol yang ada pada antarmuka ini yang terlihat pada gambar 3.7.



Gambar 3.7 Antarmuka *Game*

3.3.5 Perancangan Level Permainan

Perancangan ini menjelaskan tentang setiap level permainan. Terdapat tiga level pada permainan ini. Pada setiap level pemain harus mengumpulkan mufradat yang tersedia pada permainan.

1. Level/Scene Pertama

Pada level ini, terdapat 8 mufradat, 20 koin dan 2 NPC Pocong. Misi yang harus dilakukan adalah mengumpulkan koin dan mufradat sebanyak-banyaknya dan juga harus melewati NPC tersebut. Untuk mencapai level selanjutnya, player minimal harus mendapatkan 180 poin dan 5 mufradat.

2. Level/Scene Kedua

Pada level kedua ini, terdapat 12 mufradat, 30 koin, 1 NPC ular dan 2 NPC pocong. Untuk mencapai level selanjutnya, player minimal harus mendapatkan 280 poin dan 8 mufradat. Pada saat bermain, player akan diganggu NPC pocong dan NPC ular. Poin yang sudah didapat akan berkurang 10 poin bila menabrak kedua NPC tersebut.

3. Level/Scene Ketiga

Pada level ini, terdapat 15 mufradat, 20 koin, 1 NPC ular, 2 NPC jin dan 2 NPC pocong. Misi yang harus dilakukan adalah mengumpulkan koin dan mufradat sebanyak-banyaknya dan juga harus melewati NPC tersebut. Untuk mencapai level selanjutnya, player minimal harus mendapatkan 380 poin dan 12 mufradat. Pada saat bermain, player akan diganggu NPC pocong, NPC jin, dan NPC ular. Poin yang sudah didapat akan berkurang 10 poin bila menabrak kedua NPC tersebut.

3.3.6 Cara Memainkan *Game*

Cara memainkan *game* ini cukup mudah. Pemain mengoperasikan *game* dengan touch (sentuhan). Terdapat beberapa tombol yang digunakan pada *game* ini. Tombol tersebut yakni tombol arah panah yang digunakan untuk bergerak ke kiri dan ke kanan serta satu tombol yang digunakan untuk melompat.

3.3.7 Kebutuhan Smartphone Pemain

Berikut ini merupakan tabel daftar spesifikasi smartphone yang harus dimiliki pemain untuk memainkan *game* ini.

Tabel 3.3 Daftar Spesifikasi Smartphone

Kebutuhan	Spesifikasi Minimum	Spesifikasi Rekomendasi
<i>Smartphone</i>	Android versi 4.0 (Ice Cream Sandwich)	Android Versi 4.0 ICS (Ice Cream Sandwich)
RAM	512 MB	512 MB

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

Pada bab ini membahas tentang implementasi dan perancangan terhadap aplikasi yang sudah dibuat. Serta melakukan pengujian terhadap aplikasi yang dibuat untuk mengetahui apakah aplikasi tersebut telah berjalan sesuai dengan yang diharapkan. Sebelum diimplementasikan, terlebih dahulu dipaparkan spesifikasi sistem perangkat keras (*hardware*) dan perangkat lunak (*software*). Untuk pembuatan aplikasi ini diperlukan perangkat keras (*hardware*) dan lunak (*software*), adapun keperluan tersebut adalah:

1. Perangkat Keras (*Hardware*)

Perangkat keras pendukung digunakan adalah:

a. Laptop Acer Aspire 4741

Spesifikasi komputer pendukung yang digunakan:

- Processor Intel® Core™ i5-450M (2.4GHz, 3MB L3 cache)
- RAM 2 GB DDR3
- Intel® HD Graphics
- 14.0" HD LED LCD
- Hardisk 500 GB
- Sistem Operasi Windows 7 Ultimate

b. Smartphone SONY Xperia J (ST26i)

Spesifikasi smartphone pendukung yang digunakan:

- Processor Qualcomm MSM7227A Snapdragon, 1 GHz Cortex-A5 Adreno 200
- RAM 512 MB
- Sistem Operasi Android v4.04 (Ice Cream Sandwich)

2. Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan untuk pembuatan aplikasi ini adalah:

- Android Development Tool (ADT), untuk membangun aplikasi.
- Corel Draw X6, untuk mendesain tampilan.
- Adobe Photoshop CS6, untuk membuat gambar.

4.2 Implementasi Kecerdasan Buatan Perilaku NPC

Implementasi merupakan proses pembangunan komponen-komponen pokok suatu sistem, yang didasarkan pada desain dan rancangan yang telah dibuat sebelumnya. Perceptron digunakan untuk memberikan perilaku NPC. Pada implementasi ini perceptron dibuat dalam kelas sendiri namun memiliki method `Perceptron(int x1, int x2)` sehingga method tersebut dapat dipanggil pada method lain. Berikut adalah *source code* perceptron yang digunakan untuk menentukan perilaku NPC.

```
public class Perceptron {
    public static int Y = 0;
    public Perceptron(int x1, int x2) {
        double teta = 0.6;
        int bias = -1;
        int input1 = x1;
        int input2 = x2;
        double bobot1 = 1;
        double bobot2 = 1;
        double aktivasi = (input1 * bobot1) + (input2 *
```

```

bobot2);
    double yOut = bias + aktivasi;
    if (yOut < -teta) {
        Y = -1;
    } else if ((yOut >= -teta) && (yOut <= teta)) {
        Y = 0;
    } else {
        Y = 1;
    }
}
}
}

```

4.2.1 Kecerdasan Buatan Perilaku NPC Pocong

Pada bagian ini akan dibahas mengenai implementasi Perceptron untuk memberikan perilaku NPC Pocong. Pada saat permainan dimulai perceptron akan bekerja berdasarkan syarat yang sudah ditentukan. Berikut ini adalah *source code* untuk memberikan pergerakan perilaku NPC Pocong.

Source code pergerakan perilaku NPC Pocong

```

public void setRunning() {
    body.setLinearVelocity(new Vector2(1,
body.getLinearVelocity().y));
    final long[] PLAYER_ANIMATE = new long[] { 150, 150,
150 };
    animate(PLAYER_ANIMATE, 1, 3, true);
}
public void setRunning2() {
    body.setLinearVelocity(new Vector2(2,
body.getLinearVelocity().y));
    final long[] PLAYER_ANIMATE = new long[] { 150, 150,
150 };
    animate(PLAYER_ANIMATE, 1, 3, true);
}
public void setRunnging3() {
    body.setLinearVelocity(new Vector2(2,
body.getLinearVelocity().y));
    final long[] PLAYER_ANIMATE = new long[] { 150, 150,
150 };
    animate(PLAYER_ANIMATE, 1, 3, true);
}
}

```

```

public void setRunningBack() {
    body.setLinearVelocity(new Vector2(-1,
body.getLinearVelocity().y));
    final long[] PLAYER_ANIMATE = new long[] { 150, 150,
150 };
    animate(PLAYER_ANIMATE, 5, 7, true);
}
public void setRunningBack2() {
    body.setLinearVelocity(new Vector2(-2,
body.getLinearVelocity().y));
    final long[] PLAYER_ANIMATE = new long[] { 150, 150,
150 };
    animate(PLAYER_ANIMATE, 5, 7, true);
}
public void setRunningBack3() {
    body.setLinearVelocity(new Vector2(-4,
body.getLinearVelocity().y));
    final long[] PLAYER_ANIMATE = new long[] { 150, 150,
150 };
    animate(PLAYER_ANIMATE, 5, 7, true);
}

```

Source code diatas merupakan method-method yang digunakan untuk memberikan pergerakan perilaku NPC Pocong. Terdapat enam method dimana method tersebut akan di panggil pada kelas *GameScene*. Berikut ini *source code* pemberian perilaku NPC yang sudah mengimplementasi method pergerakan NPC.

***Source code* pemberian perilaku NPC Pocong scene pertama:**

```

if (score < 100){
    input1 = 0;
} else {
    input1 = 1;
}
if (mufradat < 5){
    input2 = 0;
} else {
    input2 = 1;
}
new Perceptron(input1, input2);
if (Perceptron.Y == -1) {
    if (player.getX() >= getX()) {
        setRunning();
    } else if (player.getX() < getX()) {

```

```

        setRunningBack();
    }
    pocong.setScale(1f);
} else if (Perceptron.Y == 0) {
    if (player.getX() >= getX()) {
        setRunning2();
    } else if (player.getX() < getX()) {
        setRunningBack2();
    }
    pocong.setScale(1f);
} else if (Perceptron.Y == 1) {
    if (player.getX() >= getX()) {
        setRunnging3();
    } else if (player.getX() < getX()) {
        setRunningBack3();
    }
    pocong.setScale(2f);
}

```

Source code di atas mengimplementasikan perceptron yaitu mengatur perilaku NPC Pocong pada scene pertama. Pada *source code* diatas terdapat variabel `score` dan `mufradat`. Variabel tersebut merupakan acuan yang dijadikan sebagai input pada perceptron. Pada scene pertama memiliki aturan `score < 100` dan `mufradat < 5`, aturan tersebut digunakan untuk menentukan input. Jika input sudah didapatkan maka akan dilanjutkan atau diproses pada method `new Perceptron(input1, input2)` untuk didapatkan nilai `Perceptron.Y` (output). Setelah itu nilai output akan diproses kembali untuk mendapatkan perilaku yang sesuai dengan syarat yang sudah ditentukan. Terdapat tiga pilihan perilaku NPC yang sudah sesuai dengan perancangan sebelumnya pada BAB III. Jika output `Perceptron.Y == -1` maka akan menjalankan method `setRunning()` atau `setRunningBack()`, jika output `Perceptron.Y == 0` maka akan menjalankan method `setRunning2()` atau `setRunningBack2()`, jika

`output Perceptron.Y == 1` maka akan menjalankan method `setRunning3()` atau `setRunningBack3()`.

Pada scene kedua dan ketiga memiliki *source code* yang sama, hanya persyaratan `score` dan `mufradat` saja yang dibedakan. Pada scene kedua memiliki persyaratan `score < 180` dan `mufradat < 8` sedangkan scene ketiga memiliki persyaratan `score < 260` dan `mufradat < 12`.

4.2.2 Kecerdasan Buatan Perilaku NPC Jin

Implementasi perceptron perilaku pada NPC Jin tidak berbeda dengan implementasi Perceptron perilaku NPC Pocong, hanya saja NPC Jin ini hanya ada pada scene ketiga saja. Berikut ini *source code* untuk memberikan perilaku pada NPC Jin.

Source code pemberian perilaku NPC Jin scene ketiga:

```

if (score < 260){
    input1 = 0;
} else {
    input1 = 1;
}
if (mufradat < 12){
    input2 = 0;
} else {
    input2 = 1;
}
new Perceptron(input1, input2);
if (Perceptron.Y == -1) {
    if (player.getX() >= getX()) {
        setRunning();
    } else if (player.getX() < getX()) {
        setRunningBack();
    }
    jin.setScale(1f);
} else if (Perceptron.Y == 0) {
    if (player.getX() >= getX()) {

```

```

        setRunning2();
    } else if (player.getX() < getX()) {
        setRunningBack2();
    }
    jin.setScale(1f);
} else if (Perceptron.Y == 1) {
    if (player.getX() >= getX()) {
        setRunnging3();
    } else if (player.getX() < getX()) {
        setRunningBack3();
    }
    jin.setScale(2f);
}
}

```

Source code di atas mengimplementasikan perceptron yaitu mengatur perilaku NPC Jin. Pada *source code* diatas terdapat variabel `score` dan `mufradat`. Variabel tersebut merupakan acuan yang dijadikan sebagai input pada perceptron. Pada scene pertama memiliki aturan `score < 260` dan `mufradat < 12`, aturan tersebut digunakan untuk menentukan input. Jika input sudah didapatkan maka akan dilanjutkan atau diproses pada method `new Perceptron(input1, input2)` untuk didapatkan nilai `Perceptron.Y` (output). Setelah itu nilai output akan diproses kembali untuk mendapatkan perilaku yang sesuai dengan syarat yang sudah ditentukan sama halnya pada perilaku NPC Pocong.

4.2.3 Kecerdasan Buatan Perilaku NPC Ustadz

Pada implementasi ini tidak berbeda dengan implementasi NPC Pocong dan Jin tetapi yang membedakan hanya terdapat perilakunya saja. NPC Ustadz juga menggunakan perceptron untuk memberikan perilakunya. NPC ini terdapat pada

scene pertama sampai ketiga sama seperti NPC Pocong. Berikut *source code* untuk memberikan perilaku NPC Ustadz.

Source code pemberian perilaku NPC Ustadz scene pertama:

```

if (score < 180) {
    input1 = 0;
} else {
    input1 = 1;
}
if (mufradat < 5) {
    input2 = 0;
} else {
    input2 = 1;
}
new Perceptron(input1, input2);
if (Perceptron.Y == -1) {
    if (player.collidesWith(this)) {
        addToScore(20);
        tampilToast("Score anda ditambah 20 poin");
    }
} else if (Perceptron.Y == 0) {
    if (player.collidesWith(this)) {
        addToScore(10);
        tampilToast("Score anda ditambah 10 poin");
    }
} else if (Perceptron.Y == 1) {
    addToScore(0);
}

```

Tidak berbeda dengan NPC Pocong dan Jin, pada variabel `score` dan `mufradat` digunakan sebagai input untuk menghitung atau memproses method `Perceptron(input1, input2);`. Setelah didapat nilai outputnya maka akan di proses lagi pada aturan `if-else` untuk menentukan perilaku yang tepat sesuai dengan syarat yang sudah ditentukan. Ada tiga perilaku yang sudah diberikan pada NPC ini yang sesuai dengan perancangan pada BAB III. Perilaku tersebut diberikan berdasarkan dengan output (`Perceptron.Y`) yang di dapat, jika `Perceptron.Y == -1` maka NPC akan memberikan tambahan `score` sebanyak

20 poin dan jika $\text{Perceptron.Y} == 0$ maka NPC akan memberikan tambahan score sebanyak 10 poin sedangkan $\text{Perceptron.Y} == 1$ maka NPC tidak memberikan poin.

Pada scene kedua dan ketiga memiliki implementasi *source code* yang sama, hanya aturan *score* dan *mufradat* saja yang berbeda. Pada scene kedua memiliki aturan $\text{score} < 280$ dan $\text{mufradat} < 8$ sedangkan scene ketiga memiliki aturan $\text{score} < 380$ dan $\text{mufradat} < 12$.

4.3 Implementasi Aplikasi *Game*

Pada pembahasan ini merupakan hasil implementasi aplikasi *game* mencari *mufradat*. Berikut ini adalah beberapa hasil implementasi aplikasi *game* tersebut.

4.3.1 Antarmuka *Splashscreen*

Pada saat *game* dijalankan akan menampilkan tampilan seperti pada gambar

4.1. *Splashscreen* menampilkan berupa tampilan wajah karakter dan nama aplikasi.



Gambar 4.1 Tampilan *Splashscreen*

4.3.2 Antarmuka Menu *Game*

Menu dalam *game* ini terdapat lima menu yang memiliki fungsi masing-masing, antara lain: mulai, suara aktif, suara nonaktif, informasi dan keluar. Pada tabel 4.1 merupakan keterangan dari fungsi tombol yang ada pada tampilan menu.

Tabel 4.1 Keterangan Fungsi Tombol Menu Permainan

No.	Menu	Keterangan
1.		Untuk memulai permainan
2.		Untuk mengaktifkan suara efek dan musik
3.		Untuk menonaktifkan suara music
4.		Untuk menampilkan informasi tentang <i>game</i>
5.		Untuk keluar dari permainan

Pada saat pemain menjalankan aplikasi makan akan menampilkan tampilan seperti gambar 4.2. Apabila pemain menekan tombol suara aktif maka suara musik aktif seperti pada gambar 4.3.



Gambar 4.2 Tampilan Menu Utama

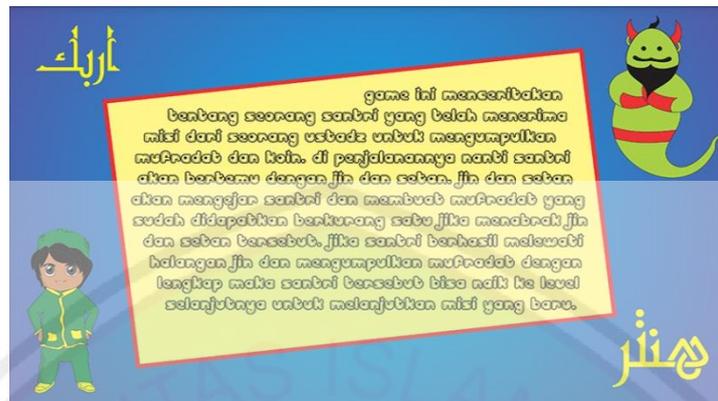


Gambar 4.3 Tampilan Menu Suara Aktif

Pada tombol suara nonaktif, akan membuat suara musik pada menu di nonaktifkan. Jika pemain menekan tombol info maka akan menuju pada halaman informasi permainan. Kedua tampilan terlihat pada gambar 4.4 dan 4.5.



Gambar 4.4 Tampilan Menu Suara Nonaktif



Gambar 4.5 Tampilan Informasi Permainan

Pada gambar 4.6 merupakan tampilan *loading*. Tampilan tersebut akan tampil setiap pemain ingin berpindah halaman.



Gambar 4.6 Tampilan *Loading*

4.3.3 Scene *Game* Pertama

Setelah menekan tombol mulai pada tampilan cara bermain, maka pemain akan memasuki halaman cara bermain scene pertama seperti pada gambar 4.7 dan memulai permainan seperti pada gambar 4.8.



Gambar 4.7 Cara Bermian Scene Pertama

Pada scene pertama ini memiliki latar belakang dengan keadaan di pagi hari. Pada scenen ini, pemain mengumpulkan score sebanyak 200 dan mufradat sebanyak 8 gambar.



Gambar 4.8 Tampilan Scene Pertama

Pada gambar 4.9 merupakan tampilan pemain dikejar oleh NPC pocong. Pada status NPC jahat terlihat bahwa musuh mengejar pemain dengan pelan dikarenakan score dan mufradat yang didapat oleh pemain sudah terkumpul lumayan banyak. Sedangkan pada gambar 4.10 merupakan tampilan pemain mendapatkan bantuan tambahan poin dari NPC Baik.



Gambar 4.9 Tampilan NPC Mengejar dengan Pelan



Gambar 4.10 Tampilan NPC Baik Menolong Player

4.3.4 Scene Game Kedua

Pemain akan memasuki scene kedua jika pemain sudah mengumpulkan poin dan mufradat dengan lengkap. Sebelum memasuki scene kedua, terdapat tampilan perintah bermain seperti pada gambar 4.11. Pada scene ini memiliki latar belakang siang hari. Gambar 4.12 merupakan tampilan awal pada scene kedua. Pemain harus mengumpulkan sekurang-kurangnya 280 poin dan mufradat 8 gambar. Terdapat dua NPC pada scene ini, yaitu NPC ular dan NPC pocong. Pemain harus

menghindari kedua NPC tersebut agar poin yang didapat tidak berkurang. Jika pemain menabrak kedua NPC

tersebut maka poin pemain akan berkurang 10 poin seperti pada gambar

4.13.



Gambar 4.11 Cara bermain Scene Kedua



Gambar 4.12 Tampilan Scene Kedua



Gambar 4.13 Tampilan Poin Berkurang

Pada gambar 4.14 terlihat pemain dikejar oleh NPC jahat dan NPC tersebut bertambah besar karena score dan mufradat yang didapat oleh pemain telah terpenuhi sedangkan pada gambar 4.15 terlihat NPC baik memberikan poin sebesar 10 poin dikarenakan poin yang didapat pemain untuk melanjutkan ke scene selanjutnya kurang.



Gambar 4.14 Tampilan NPC Jahat Bertambah Besar



Gambar 4.15 Tampilan NPC Baik Menolong Pemain

4.3.5 Scene *Game* Ketiga

Pada scene ketiga ini memiliki latar belakang waktu malam hari. Sebelum memainkan permainan scene ketiga, terdapat tampilan perintah bermain yang harus dilakukan seperti gambar 4.16. Pada scene ketiga ini terdapat dua scenario diatas tanah dan dibawah tanah. Tampilan scene seperti pada gambar 4.17. pada scene ini, pemain harus mengumpulkan poin dan mufradat lebih banyak lagi sekurang-kurangnya 380 poin dan 12 mufradat. Ada tiga musuh NPC di scene ketiga ini, antara lain NPC pocong, NPC jin dan ular. Seperti halnya pada scene pertama dan kedua, pemain harus menghindari ketiga NPC musuh tersebut jika tidak poin yang didapat akan berkurang 10 poin.



Gambar 4.16 Cara Bermain Scene Ketiga



Gambar 4.17 Tampilan Scene Ketiga

Jika poin yang didapat pemain kurang untuk menyelesaikan permainan maka pemain dapat meminta bantuan tambahan poin kepada NPC baik seperti pada gambar 4.18 tetapi, jika poin dan mufradat yang didapat oleh pemain sudah memenuhi untuk menyelesaikan permainan maka pemain dapat langsung menuju tanda panah untuk menyelesaikan permainan yang terlihat pada gambar 4.19.



Gambar 4.18 Tampilan NPC Baik Menolong Pemain



Gambar 4.19 Tampilan Permainan Selesai

4.4 Uji Coba

Pada subbab ini membahas tentang uji coba yang telah dilakukan. Ada dua uji coba yang telah dilakukan yaitu uji coba algoritma perceptron dan uji coba aplikasi. Berikut pembahasan uji coba tersebut.

4.4.1 Uji Coba Algoritma Perceptron

Uji coba algoritma perceptron ini dilakukan untuk mengetahui pola data yang akan dikenali. Proses ini melakukan uji coba pada *learning rate*, *threshold*,

dan mencari nilai epoch yang dicapai. Data-data tersebut dapat dilihat pada tabel

4.2.

Tabel 4.2 Hasil Uji Coba Nilai Epoch

Learning rate	Threshold	Max epoch	Epoch yang dicapai
0.1	0.1	1000	15
0.2	0.1	1000	8
0.3	0.1	1000	6
0.4	0.1	1000	1000
0.5	0.1	1000	5
0.6	0.1	1000	1000
0.7	0.1	1000	1000
0.8	0.1	1000	1000
0.9	0.1	1000	4
1	0.1	1000	4
0.1	0.2	1000	17
0.2	0.2	1000	10
0.3	0.2	1000	8
0.4	0.2	1000	5
0.5	0.2	1000	5
0.6	0.2	1000	5
0.7	0.2	1000	1000
0.8	0.2	1000	4
0.9	0.2	1000	4
1	0.2	1000	4
0.1	0.3	1000	19
0.2	0.3	1000	10
0.3	0.3	1000	8
0.4	0.3	1000	7
0.5	0.3	1000	5
0.6	0.3	1000	5
0.7	0.3	1000	1000
0.8	0.3	1000	4
0.9	0.3	1000	4
1	0.3	1000	4
0.1	0.4	1000	21
0.2	0.4	1000	12
0.3	0.4	1000	8
0.4	0.4	1000	7
0.5	0.4	1000	5
0.6	0.4	1000	6
0.7	0.4	1000	6

Learning rate	Threshold	Max epoch	Epoch yang dicapai
0.8	0.4	1000	4
0.9	0.4	1000	4
1	0.4	1000	4
0.1	0.5	1000	23
0.2	0.5	1000	12
0.3	0.5	1000	10
0.4	0.5	1000	7
0.5	0.5	1000	7
0.6	0.5	1000	6
0.7	0.5	1000	6
0.8	0.5	1000	4
0.9	0.5	1000	4
1	0.5	1000	4
0.1	0.6	1000	25
0.2	0.6	1000	14
0.3	0.6	1000	10
0.4	0.6	1000	7
0.5	0.6	1000	7
0.6	0.6	1000	6
0.7	0.6	1000	6
0.8	0.6	1000	4
0.9	0.6	1000	4
1	0.6	1000	4
0.1	0.7	1000	27
0.2	0.7	1000	14
0.3	0.7	1000	10
0.4	0.7	1000	9
0.5	0.7	1000	7
0.6	0.7	1000	6
0.7	0.7	1000	6
0.8	0.7	1000	6
0.9	0.7	1000	4
1	0.7	1000	4
0.1	0.8	1000	29
0.2	0.8	1000	16
0.3	0.8	1000	12
0.4	0.8	1000	9
0.5	0.8	1000	7
0.6	0.8	1000	7
0.7	0.8	1000	6
0.8	0.8	1000	6
0.9	0.8	1000	1000
1	0.8	1000	4

Learning rate	Threshold	Max epoch	Epoch yang dicapai
0.1	0.9	1000	31
0.2	0.9	1000	16
0.3	0.9	1000	12
0.4	0.9	1000	9
0.5	0.9	1000	7
0.6	0.9	1000	7
0.7	0.9	1000	6
0.8	0.9	1000	6
0.9	0.9	1000	6
1	0.9	1000	4
0.1	1	1000	33
0.2	1	1000	18
0.3	1	1000	12
0.4	1	1000	10
0.5	1	1000	9
0.6	1	1000	8
0.7	1	1000	6
0.8	1	1000	6
0.9	1	1000	6
1	1	1000	6

Percobaan dilakukan untuk mendapatkan nilai *learning rate* dan nilai *threshold* terbaik. Pada tabel 4.2 dapat diketahui nilai *learning rate* dan *threshold* berpengaruh pada performansi jaringan. Semakin besar nilai *learning rate*, maka semakin sedikit iterasi yang diperlukan untuk mendapatkan nilai bobot dan bias mencapai stabil. Proses pembelajaran dilakukan secara maksimal, yaitu dilakukan sampai 1000 iterasi. Pada saat uji coba, dihasilkan 93% keberhasilan dalam melakukan proses *learning* dan terdapat 7% yang gagal. Kegagalan ini terjadi karena nilai error yang dihasilkan kurang begitu baik karena tidak sama dengan nilai target.

Setelah diketahui nilai epoch maksimal maka dilakukan uji coba kembali untuk mengetahui waktu eksekusi. Hasil uji coba waktu eksekusi dapat dilihat pada tabel 4.3.

Tabel 4.3 Hasil Uji Coba Waktu Eksekusi

Learning rate	Threshold	Max epoch	Waktu eksekusi (detik)
1	0.1	4	1.497
1	0.2	4	1.665
1	0.3	4	1.474
1	0.4	4	1.566
1	0.5	4	1.802
1	0.6	4	1.433
1	0.7	4	1.582
1	0.8	4	1.695
1	0.9	4	1.573
1	1	6	1.628

Data yang digunakan pada tahap ini berjumlah empat data. Empat data tersebut akan di lakukan pengenalan dengan menggunakan *learning rate* = 1, *threshold* = 0.6, nilai bias = -1, waktu eksekusi = 1.433 detik, dan maksimum epoch = 100 setelah melakukan uji coba. Data uji coba tersebut digunakan untuk mencari nilai bobot dan nilai bias yang terbaik. Berikut hasil pencarian nilai bobot dan nilai bias.

Hasil pencarian bobot dan nilai bias akhir

```

epoch 1
nilai bobot: 0.0 nilai input: 0.0
nilai bobot: 0.0 nilai input: 0.0
Nilai (bobot+input): 0.0
hasil aktivasi: 1.0
Nilai Output: 1.0
nilai error: -1.0
nilai bobot baru: 0.0
nilai bobot baru: 0.0
nilai bias: 0.0

```

```
nilai bobot: 0.0 nilai input: 0.0
nilai bobot: 0.0 nilai input: 1.0
Nilai (bobot+input): 0.0
hasil aktivasi: 0.0
Nilai Output: 0.0
nilai error: 0.0
```

```
nilai bobot: 0.0 nilai input: 1.0
nilai bobot: 0.0 nilai input: 0.0
Nilai (bobot+input): 0.0
hasil aktivasi: 0.0
Nilai Output: 0.0
nilai error: 0.0
```

```
nilai bobot: 0.0 nilai input: 1.0
nilai bobot: 0.0 nilai input: 1.0
Nilai (bobot+input): 0.0
hasil aktivasi: 0.0
Nilai Output: 0.0
nilai error: 1.0
nilai bobot baru: 1.0
nilai bobot baru: 1.0
nilai bias: 1.0
```

```
epoh 2
nilai bobot: 1.0 nilai input: 0.0
nilai bobot: 1.0 nilai input: 0.0
Nilai (bobot+input): 0.0
hasil aktivasi: 1.0
Nilai Output: 1.0
nilai error: -1.0
nilai bobot baru: 1.0
nilai bobot baru: 1.0
nilai bias: 0.0
```

```
nilai bobot: 1.0 nilai input: 0.0
nilai bobot: 1.0 nilai input: 1.0
Nilai (bobot+input): 1.0
hasil aktivasi: 1.0
Nilai Output: 1.0
nilai error: 0.0
nilai bobot baru: 1.0
nilai bobot baru: 1.0
nilai bias: 0.0
```

```
nilai bobot: 1.0 nilai input: 1.0
nilai bobot: 1.0 nilai input: 0.0
Nilai (bobot+input): 1.0
hasil aktivasi: 1.0
Nilai Output: 1.0
nilai error: 0.0
nilai bobot baru: 1.0
nilai bobot baru: 1.0
```

```
nilai bias: 0.0

nilai bobot: 1.0 nilai input: 1.0
nilai bobot: 1.0 nilai input: 1.0
Nilai (bobot+input): 2.0
hasil aktivasi: 2.0
Nilai Output: 1.0
nilai error: 1.0
```

epoh 3

```
nilai bobot: 1.0 nilai input: 0.0
nilai bobot: 1.0 nilai input: 0.0
Nilai (bobot+input): 0.0
hasil aktivasi: 0.0
Nilai Output: 0.0
nilai error: -1.0
nilai bobot baru: 1.0
nilai bobot baru: 1.0
nilai bias: -1.0
```

```
nilai bobot: 1.0 nilai input: 0.0
nilai bobot: 1.0 nilai input: 1.0
Nilai (bobot+input): 1.0
hasil aktivasi: 0.0
Nilai Output: 0.0
nilai error: 0.0
```

```
nilai bobot: 1.0 nilai input: 1.0
nilai bobot: 1.0 nilai input: 0.0
Nilai (bobot+input): 1.0
hasil aktivasi: 0.0
Nilai Output: 0.0
nilai error: 0.0
```

```
nilai bobot: 1.0 nilai input: 1.0
nilai bobot: 1.0 nilai input: 1.0
Nilai (bobot+input): 2.0
hasil aktivasi: 1.0
Nilai Output: 1.0
nilai error: 1.0
```

epoh 4

```
nilai bobot: 1.0 nilai input: 0.0
nilai bobot: 1.0 nilai input: 0.0
Nilai (bobot+input): 0.0
hasil aktivasi: -1.0
Nilai Output: -1.0
nilai error: -1.0
```

```
nilai bobot: 1.0 nilai input: 0.0
nilai bobot: 1.0 nilai input: 1.0
Nilai (bobot+input): 1.0
hasil aktivasi: 0.0
Nilai Output: 0.0
```

```

nilai error: 0.0

nilai bobot: 1.0 nilai input: 1.0
nilai bobot: 1.0 nilai input: 0.0
Nilai (bobot+input): 1.0
hasil aktivasi: 0.0
Nilai Output: 0.0
nilai error: 0.0

nilai bobot: 1.0 nilai input: 1.0
nilai bobot: 1.0 nilai input: 1.0
Nilai (bobot+input): 2.0
hasil aktivasi: 1.0
Nilai Output: 1.0
nilai error: 1.0

1 1 adalah 1
1 0 adalah 0
0 1 adalah 0
0 0 adalah -1

```

Hasil pencarian bobot dan nilai bias akhir maka dihasilkan nilai bobot = 1 dan nilai bias = -1. Hasil tersebut digunakan untuk data NPC jahat dan NPC baik. Data dapat dilihat pada tabel 4.4 dan 4.5.

Tabel 4.4 Data NPC Jahat

No.	x1	x2	target
1	0	0	-1
2	0	1	0
3	1	0	0
4	1	1	1

Tabel 4.5 Data NPC Baik

No.	x1	x2	target
1	0	0	-1
2	0	1	0
3	1	0	0
4	1	1	1

Setelah didapatkan nilai *learning rate*, *threshold*, nilai bobot dan nilai bias yang stabil, maka dilakukan pengenalan terhadap aplikasi. Hasil pengenalan terlihat pada gambar dibawah ini:



(a)



(b)



(c)

Gambar 4.20 Hasil Uji Coba NPC Jahat

Gambar 4.20 merupakan hasil pengenalan terhadap NPC jahat. Pada tampilan terlihat tulisan bahwa target yang diinginkan sudah tercapai dengan akurat.



(a)



(b)



(c)

Gambar 4.21 Hasil Uji Coba NPC Baik

Gambar 4.21 merupakan hasil hasil pengenalan terhadap NPC baik. Pada tampilan terlihat tulisan bahwa target yang diinginkan sudah tercapai dengan akurat.

4.4.2 Uji Coba Aplikasi

Uji coba ini dilakukan untuk mengetahui apakah aplikasi yang telah dibuat dapat diimplementasikan terhadap *device* atau *smartphone* yang lain. Berikut hasil pengujian dari aplikasi pada beberapa *smartphone* dalam bentuk tabel 4.6.

Tabel 4.6 Hasil Pengujian Aplikasi

No.	Versi OS	Layar	CPU	RAM	Keterangan
1.	v2.3.5 (Gingerbread)	3"	832 MHz Processor	290 MB	Tampilan menu berjalan dengan baik. Efek musik dan tombol berfungsi dengan baik. Tampilan <i>game</i> tidak dapat tampil.
2.	v4.0 (Ice Cream Sandwich)	3.2"	Qualcomm MSM8255 Snapdragon, Adreno 205, 1 GHz Scorpion	512 MB	Seluruh tampilan berjalan dengan baik. Efek musik dan tombol berfungsi dengan baik. <i>Game</i> berjalan dengan baik.
3.	v4.0 (Ice Cream Sandwich)	3.5"	Qualcomm MSM7225A, Adreno 200, 800 MHz Cortex-A5	512 MB	Seluruh tampilan berjalan dengan baik. Efek musik dan tombol berfungsi dengan baik. <i>Game</i> berjalan dengan baik.
4.	v4.0.4 (Ice Cream Sandwich)	4"	Qualcomm MSM7227A Snapdragon, 1 GHz Cortex-A5 Adreno 200	512 MB	Seluruh tampilan berjalan dengan baik. Efek musik dan tombol berfungsi dengan baik. <i>Game</i> berjalan dengan baik.
5.	v4.1 (Jelly Bean)	4.6"	Qualcomm MSM8960Pro Snapdragon, Dual-core 1.7 GHz, Adreno 320	1 GB	Seluruh tampilan berjalan dengan baik. Efek musik dan tombol berfungsi dengan baik. <i>Game</i> berjalan dengan baik.

No.	Versi OS	Layar	CPU	RAM	Keterangan
6.	v4.2.2 (Jelly Bean)	5"	PowerVR SGX544, MTK Quad Core 1,2 GHz	1 GB	Seluruh tampilan berjalan dengan baik. Efek musik dan tombol berfungsi dengan baik. <i>Game</i> berjalan dengan baik.

4.5 Integrasi *Game* Arabic Hunter dengan Islam

Allah SWT menciptakan bumi beserta isinya untuk dipelajari oleh makhluknya yang memiliki akal dan pikiran yaitu manusia. Manusia mempelajari seluruh segala sesuatu yang diciptakan oleh Allah SWT dengan tujuan untuk mengetahui tanda-tanda kebesaran Allah SWT maka manusia akan mendapatkan ilmu pengetahuan yang melimpah. Manusia yang memiliki pengetahuan akan di angkat derajatnya dan dimuliakan dirinya dengan ilmu oleh Allah SWT. Dalam surat Al-Mujadilah ayat 11 Allah berfirman:

يَأْتِيهَا الَّذِينَ ءَامَنُوا إِذَا قِيلَ لَكُمْ تَفَسَّحُوا فِي الْمَجَالِسِ فَافْسَحُوا يَفْسَحِ
 اللَّهُ لَكُمْ وَإِذَا قِيلَ أَنْشُرُوا فَأَنْشُرُوا يَرْفَعُ اللَّهُ الَّذِينَ ءَامَنُوا مِنْكُمْ وَالَّذِينَ
 أُوتُوا الْعِلْمَ دَرَجَاتٍ وَاللَّهُ بِمَا تَعْمَلُونَ خَبِيرٌ ﴿١١﴾

Artinya:

“Hai orang-orang beriman apabila dikatakan kepadamu: "Berlapang-lapanglah dalam majlis", maka lapangkanlah niscaya Allah akan memberi kelapangan untukmu. Dan apabila dikatakan: "Berdirilah kamu", maka berdirilah, niscaya Allah akan meninggikan orang-orang yang beriman di antaramu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat. Dan Allah Maha Mengetahui apa yang kamu kerjakan.”

Ibnu katsir dalam tafsirnya mengatakan: *“Allah akan meninggikan orang-orang yang beriman di antaramu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat. Dan Allah Maha Mengetahui apa yang kamu kerjakan.”* Maksudnya, janganlah kalian berkeyakinan bahwa jika salah seorang di antara kalian memberi kelapangan kepada saudaranya, baik yang datang maupun yang akan pergi lalu dia keluar, maka akan mengurangi hak-nya. Bahkan hal itu merupakan ketinggian dan perolehan martabat di sisi Allah. Dan Allah tidak menyia-nyiakan hal tersebut, bahkan Dia akan memberikan balasan kepadanya di dunia dan di akhirat. Sesungguhnya orang yang merendahkan diri karena Allah, maka Allah akan mengangkat derajatnya dan akan memasyhurkan namanya. Allah SWT memuliakan orang yang beriman dan berilmu pengetahuan, karena dengan bekal iman mengantarkan manusia kepada ketinggian di akhirat, dan ilmu pengetahuan membawa manusia kepada ketinggian hidup di dunia.

Dalam islam, bukan hanya ilmu pengetahuan yang patut dipelajari tetapi juga bahasa arab. Bahasa arab merupakan bahasa yang digunakan dalam kitab suci al-Qur’an. Allah juga memerintahkan kepada umatnya untuk mempelajarinya agar termasuk kedalam golongan yang mengetahui. Hal ini sudah dijelaskan dalam firman-Nya dalam surat al-Fushilat ayat 3 yang berbunyi:

كِتَابٌ فُصِّلَتْ آيَاتُهُ قُرْءَانًا عَرَبِيًّا لِقَوْمٍ يَعْلَمُونَ ﴿٣﴾

Artinya:

“Kitab yang dijelaskan ayat-ayatnya, yakni bacaan dalam bahasa Arab, untuk kaum yang mengetahui.”

al-Quran adalah kitab suci yang mengandung ilmu-ilmu yang tak terbatas, dan takkan bisa digali secara tuntas. Ilmu yang dikandungnya takkan pernah habis

walau terus digali dan dikuras sepanjang masa, sepanjang kehidupan dunia masih ada. al-Quran adalah sumber ilmu yang kaya dan abadi.

Banyak cara untuk mendapatkan ilmu pengetahuan dan mempelajari bahasa arab, salah satunya adalah media permainan. *Game* Arabic Hunter dibangun untuk memperkenalkan anak-anak mempelajari bahasa arab dengan mudah. Dalam kandungan ayat 3 surat al-Fushilat sudah dijelaskan bahwa dengan mempelajari bahasa arab dapat membuat umatnya mengetahui ilmu yang ada di dalam al-Qur'an. *Game* ini menggunakan mufradat untuk memperkenalkan bahasa arab terhadap anak-anak. Walaupun hanya memperkenalkan mufradat setidaknya anak-anak sudah mengetahui bahasa arab.

BAB V

PENUTUP

5.1 Kesimpulan

Game Arabic Hunter ini memiliki genre *side scrolling* yang didalamnya terdapat edukasi yaitu memperkenalkan mufradat. Konten edukasi ini mengenalkan mufradat dengan cara menampilkan gambar beserta huruf bahasa arabnya. Cara ini digunakan untuk menghasilkan *game* yang menarik. *Game* ini berhasil diimplementasikan pada jenis *smartphone* yang berbeda agar mengetahui apakah *game* tersebut dapat di jalankan atau tidak.

Algoritma *Perceptron* merupakan algoritma pelatihan terbimbing yang dapat melakukan *learning* terhadap bobot dan menghasilkan keluaran yang sesuai. Algoritma ini diterapkan untuk menentukan perilaku dari NPC. Dari hasil implementasi dan uji coba, algoritma ini dapat digunakan untuk menentukan perilaku NPC sesuai dengan keahlian pemain dalam bermain *game* ini. Pada proses *learning* besarnya nilai *learnin rate* berpengaruh terhadap nilai epoch yang dicapai. Semakin besar nilai *learning rate* semakin kecil nilai epoch yang dicapai. Hasil uji coba dihasilkan *learning rate* = 1, *threshold* = 0.6, nilai bias = -1, waktu eksekusi = 1.433 detik. Pada proses *learning* dapat diketahui 93% yang berhasil dan 7% yang gagal dalam melakukan proses. Dari rata-rata hasil uji waktu eksekusi, untuk menyelesaikan proses *learning* menggunakan algoritma ini membutuhkan waktu 1.591 detik.

5.2 Saran

Dalam pembuatan *game* ini tentu masih banyak kekurangan yang masih perlu dilakukan perbaikan dan pengembangan untuk menjadikan aplikasi ini semakin bagus dan diminati banyak orang. Oleh karena itu penulis menyarankan beberapa hal untuk bahan pengembangan selanjutnya, diantaranya:

1. Mengembangkan *game* ini agar memiliki tampilan yang lebih menarik lagi.
2. Adanya pengacakan penempatan mufradat agar pemain tidak bosan jika memainkan *game* ini terus-menerus.
3. Adanya tambahan metode untuk NPC Jahat dalam memilih jalur mengejar pemain.
4. Adanya pengembangan *multiplatform* agar dapat dimainkan diberbagai sistem operasi *mobile* tidak hanya di android.

DAFTAR PUSTAKA

- al-Qur'anul Karim
- Arif, Yunifa Miftachul. 2010. *Strategi Menyerang pada Game FPS Menggunakan Hierarchy Finite State Machine Dan Logika Fuzzy*. Tesis telah dipublikasi. Surabaya: ITS Surabaya.
- Aryulina, Diah., Choirul Muslim, Syalfinaf Manaf, dan Endang W. Winarni. 2004. *Biologi 2*. Jakarta: Erlangga.
- comScore. 2010. *U.S. Mobile Subscriber Market Share*, http://www.comscore.com/Insights/Press_Releases/2010/12/comScore_Reports_October_2010_US_Mobile_Subscriber_Market_Share, (diakses pada tanggal 15 April 2014).
- Google. 2014. *Android-Discover Android*, <http://www.android.com/about/>, (diakses pada tanggal 6 Februari 2014).
- Google. 2014. *Dashboard-Developer Android*, <http://developer.android.com/about/dashboards/index.html>, (diakses pada tanggal 22 Desember 2013).
- Hermawan, Arif. 2006. *Jaringan Saraf Tiruan, Teori dan Aplikasinya*. Yogyakarta: Andi.
- Hong, JinHyuk dan Sung-Bae Cho. 2005. *Evolving Reactive NPCs for the Real-Time Simulation Game*. Korea: Yonsei University.
- Katsir, Ibnu. *Tafsirul Qur`anil 'Azhim*.
- National Writing Project. 2011. *Book Review: Reality is Broken: Why Games Make Us Better and How They Change the World*, <http://www.nwp.org/cs/public/print/resource/3654>, (diakses pada tanggal 6 Mei 2014).
- Nielsenwire. 2010. *The State of Mobile Apps*, <http://www.nielsen.com/us/en/newswire/2010/the-state-of-mobile-apps.html>, (diakses pada tanggal 15 April 2014).
- Puspitaningrum, Diyah. 2006. *Pengantar Jaringan Saraf Tiruan*. Yogyakarta: Andi.

Rogers, Rick. 2012. *Learning Android Game Programming*. United States: Pearson Education, Inc.

Safaat H, Nazruddin. 2011. *Android, Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Informatika.



The logo is a shield-shaped emblem with a light green background and a white border. It features the text "UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM" in a circular arrangement at the top and "PUSAT PERPUSTAKAAN" at the bottom. In the center, there is a large, stylized yellow calligraphic symbol. Overlaid on the center of the logo is the text "LAMPIRAN-LAMPIRAN" in a bold, black, serif font.

LAMPIRAN-LAMPIRAN



1. Tampilan menu pada layar 3"



2. Tampilan game yang tidak berhasil pada layar 3"



3. Tampilan menu pada layar 3.2"



4. Tampilan *game* pada layar 3.2"



5. Tampilan menu pada layar 3.5"



6. Tampilan *game* pada layar 3.5"



7. Tampilan menu pada layar 4''



8. Tampilan game pada layar 4''



9. Tampilan game pada layar 4.6''



10. Tampilan *game* pada layar 5"

