

**IMPLEMENTASI METODE A* (A-STAR) UNTUK NPC MUSUH
PADA GAME 3D PEMBELAJARAN
KOSAKATA BAHASA ARAB**

SKRIPSI

Oleh :

BADZROTUL MUFIDA

NIM. 10650116



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2016**

**IMPLEMENTASI METODE A* (A-STAR) UNTUK NPC MUSUH PADA
GAME 3D PEMBELAJARAN KOSAKATA BAHASA ARAB**

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:

BADZROTUL MUFIDA

NIM.10650116

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2016**

HALAMAN PERSETUJUAN

**IMPLEMENTASI METODE A* (A-STAR) UNTUK NPC MUSUH PADA
GAME 3D PEMBELAJARAN KOSAKATA BAHASA ARAB**

SKRIPSI

Oleh :

Nama : Badzrotul Mufida
NIM : 10650116
Jurusan : Teknik Informatika
Fakultas : Sains dan Teknologi

Telah diperiksa dan disetujui untuk diuji:

Dosen Pembimbing I

Dosen Pembimbing II

Dr.M.Faisal,MT.
NIP. 19740510 200501 1 007

Hani Nurhayati,MT.
NIP. 19780625 200801 2 006

Tanggal, 30 Juni 2016

Mengetahui,

Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

HALAMAN PENGESAHAN

**IMPLEMENTASI METODE A* (A-STAR) UNTUK NPC MUSUH PADA
GAME 3D PEMBELAJARAN KOSAKATA BAHASA ARAB**

SKRIPSI

oleh:

BADZROTUL MUFIDA
NIM. 10650116

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

Tanggal, Juni 2016

Susunan Dewan Penguji:

Tanda Tangan

- | | | | |
|--------------------|-----------------------------------------------------------------|---|---|
| 1. Penguji Utama | : <u>Fresy Nugroho, M.T</u>
NIP. 19710722 201101 1 001 | (|) |
| 2. Ketua Penguji | : <u>Fachrul Kurniawan, M.MT</u>
NIP. 19771020 200901 1 001 | (|) |
| 3. Sekretaris | : <u>Dr. Muhammad Faisal, M.T</u>
NIP. 19740510 200501 1 007 | (|) |
| 4. Anggota Penguji | : <u>Hani Nurhayati, MT</u>
NIP. 19780625 200801 2 006 | (|) |

Mengetahui,

Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdiان
NIP. 19740424 200901 1 008

**PERNYATAAN
PERNYATAAN KEASLIAN PENELITIAN**

Saya yang bertandatangan di bawah ini:

Nama : BADZROTUL MUFIDA

NIM : 10650116

Fakultas/Jurusan : Sains dan Teknologi / Teknik Informatika

Judul Penelitian : Implementasi Metode A^* (*A-Star*) Untuk NPC Musuh
Pada *Game* 3D Pembelajaran Kosakata Bahasa Arab

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, Juni 2016

Yang membuat pernyataan,

Badzrotul Mufida

NIM. 10650116

MOTTO

“perangi segala ujian dengan usaha yang keras, karena dengan usaha yang keras maka kita akan merasakan rasa manis dari ujian tersebut”

HALAMAN PERSEMBAHAN

Puji syukur segala rahmat, taufik dan hidayah kepada Allah SWT atas terselesaikannya skripsi ini. Tak lupa ucapan terima kasih sebanyak-banyaknya kepada :

Orang tua yang selalu memberi dukungan dan bersabar kepada saya dalam setiap proses pengerjaan skripsi, terima kasih juga kepada kakak yang selalu mengingatkan dan memberi dorongan.

Dosen-dosen di jurusan teknik informatika yang telah membimbing dan memberikan ilmu selama menjalani studi di jurusan Teknik Informatika, dosen wali sekaligus dosen pembimbing Bapak Faisal dan Ibu Hani Nurhayati yang selalu memberikan bimbingan dan mensupport demi kesuksesan saya. Dan tidak lupa terima kasih kepada Bapak Eko budi minarno yang memberi dukungan kepada saya.

Terima kasih juga kepada arif firmansyah yang juga telah memberikan dukungan kepada saya.

Kepada teman-teman seperjuangan mbak ziah, febrí, agus, alif, syukron, odak, fani yang telah berjuang bersama-sama selama proses pengerjaan skripsi.

Teman-teman yang membantu dan memberi masukan dalam penggarapan skripsi febrí, joehani, alif, dan yutub tutorial terima kasih banyak.

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Alhamdulillah rabbil'Alamin penulis mengucapkan sukur 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 Raharjo, M.Si, selaku rektor UIN Maulana Malik Ibrahim Malang beserta seluruh staf. Dharma Bakti Bapak dan Ibu sekalian terhadap Universitas Islam Negeri Malang turut membesarkan dan mencerdaskan penulis.
2. Dr. Hj. Bayyinatul M., drh., M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang beserta seluruh staf. Bapak dan ibu sekalian sangat berjasa memupuk dan menumbuhkan semangat untuk maju kepada penulis.
3. Bapak Dr. Cahyo Crys dian, selaku Ketua Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang, yang sudah memberi banyak memberi pengetahuan, inspirasi dan pengalaman yang berharga.

4. Bapak Dr.M.Faisal, M.T, selaku dosen pembimbing I yang telah meluangkan waktu untuk membimbing, memotivasi, mengarahkan dan memberi masukan kepada penulis dalam pengerjaan skripsi ini hingga akhir.
5. Ibu Hani Nurhayati, M.T, selaku dosen pembimbing II yang juga senantiasa memberi masukan dan nasihat serta petunjuk dalam penyusunan skripsi ini.
6. Ayah, Ibu, dan Kakak serta keluarga besar saya tercinta yang selalu memberi dukungan yang tak terhingga serta do'a yang senantiasa mengiringi setiap langkah penulis.
7. Segenap Dosen Teknik Informatika yang telah memberikan bimbingan keilmuan kepada penulis selama masa studi.
8. Teman – teman seperjuangan Teknik Informatika 2010
9. 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.*

Wassalamu'alaikum Wr. Wb.

Malang, Juni 2016

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
LEMBAR PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
HALAMAN PERNYATAAN.....	v
MOTTO	vi
PERSEMBAHAN.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL	xv
ABSTRAK	xvi
ABSTRACT	xvii
ملخص.....	xviii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.4 Manfaat Penelitian.....	4
BAB II KAJIAN PUSTAKA	5
2.1 Game (Permainan).....	5
2.2 Pengertian Game	5
2.3 Elemen-elemen Game	6
2.4 Genre Game atau Jenis Game	7
2.5 Artificial Intelligence (AI).....	11
2.6 Algoritma A* (AStar).....	12
2.7 Fungsi Heuristik	15
2.8 Penelitian Terkait	16
2.9 Game Engine Unity3D	18
2.9.1 Unity Software	18
2.9.2 Fitur-fitur.....	19

2.10 Metode Penelitian.....	21
BAB III DESAIN DAN PERANCANGAN GAME	23
3.1 Pengembangan Game.....	23
3.2 Keterangan umum Game.....	23
3.3 Objek Penelitian	24
3.4 Story Board	24
3.5 Deskripsi masing-masing karakter	28
3.6 Deskripsi Gameplay game.....	30
3.7 Scoring.....	31
3.8 Pembelajaran dalam game	31
3.9 Finite State Machine	32
3.10 Perancangan A* (AStar).....	33
3.10.1 Simulasi A* (AStar).....	33
BAB IV HASIL DAN PEMBAHASAN	49
4.1 Implementasi Sistem.....	49
4.1.1 Kebutuhan Perangkat Keras.....	49
4.1.2 Kebutuhan Perangkat Lunak.....	50
4.2 Implementasi Algoritma A* pada Perilaku NPC.....	51
4.2.1 A* Sebagai Metode Perilaku Pencarian.....	51
4.3 Implementasi Aplikasi Game.....	55
4.3.1 Tampilan <i>Splashscrene</i>	55
4.3.2 Tampilan Menu Game	56
4.3.3 Tampilan Pilihan Tema Permainan.....	57
4.3.4 Tampilan Scene Help.....	57
4.3.5 Tampilan Scene About.....	58
4.3.6 Tampilan Permainan	59
4.4 Uji Coba.....	65
4.4.1 Uji Coba Algoritma	66
4.4.2 Uji Coba Game	68
4.5 Game Pembelajaran kosakata Bahasa Arab dalam Pandangan Islam.....	70
BAB V PENUTUP	73
5.1 Kesimpulan.....	73
5.2 Saran	74

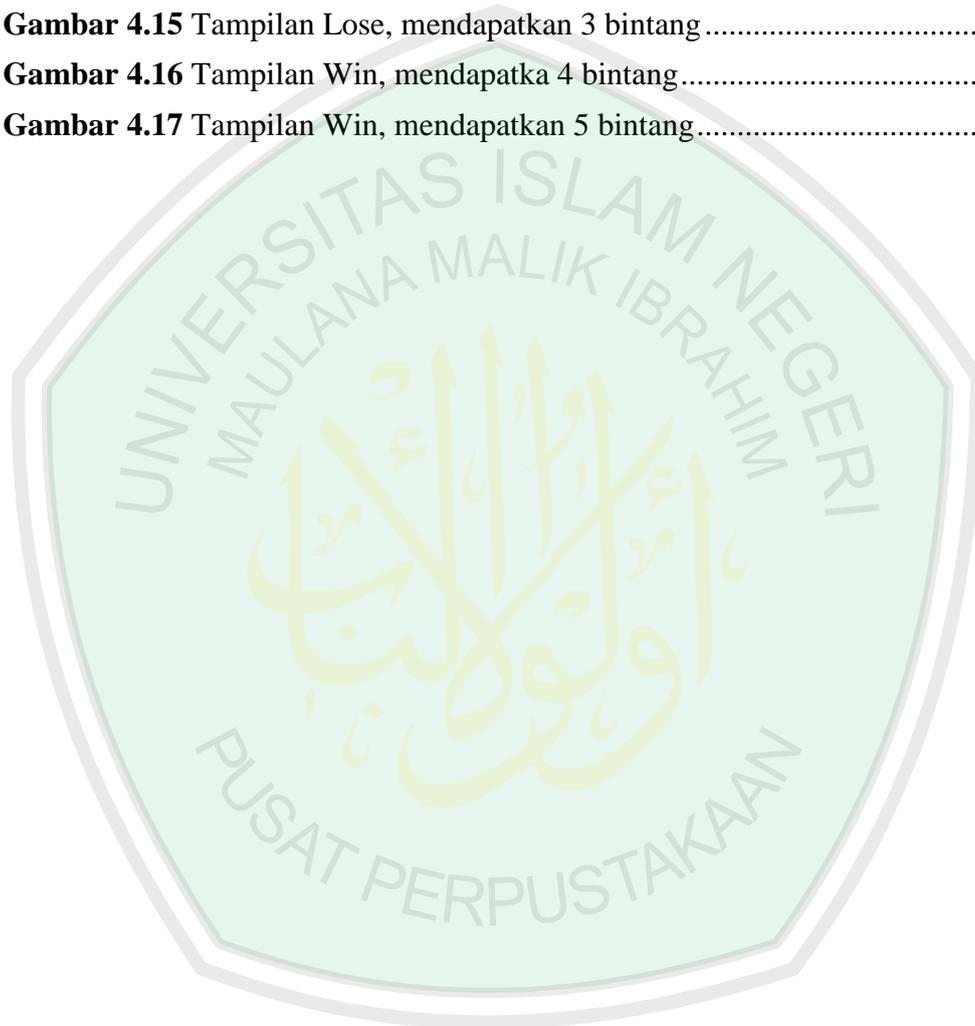
DAFTAR PUSTAKA	75
LAMPIRAN	78



DAFTAR GAMBAR

Gambar 2.1 Tahap metode penelitian.....	21
Gambar 3.1 avatar memasuki labirin.....	25
Gambar 3.2 avatar mencari NPC penduduk	25
Gambar 3.3 avatar mengantar pulang NPC penduduk	26
Gambar 3.4 NPC musuh mendekati dan memakan NPC penduduk	27
Gambar 3.5 avatar lose	27
Gambar 3.6 avatar win.....	27
Gambar 3.7 Najwa (Avatar)	28
Gambar 3.8 FSM penduduk “kampong Arab”	31
Gambar 3.9 FSM NPC musuh (tikus)	32
Gambar 3.10 Tahap awal pencarian algoritma A*	34
Gambar 3.11 Ilustrasi open list dan closed list.....	36
Gambar 3.12 Perhitungan menentukan nilai F	37
Gambar 3.13 Penjelasan A* menggunakan tree pada langkah pertama.....	39
Gambar 3.14 Penjelasan AStar pada langkah kedua	40
Gambar 3.15 Penjelasan AStar dengan tree pada langkah kedua	41
Gambar 3.16 Penjelasan AStar pada langkah ketiga.....	42
Gambar 3.17 PenjelasanAStar dengan tree pada langkah ketiga	43
Gambar 3.18 Penjelasan AStar pada langkah keempat	43
Gambar 3.19 Penjelasan AStar dengan tree pada langkah keempat	44
Gambar 3.20 Penjelasan AStar pada langkah kelima.....	45
Gambar 3.21 Penjelasan AStar dengan tree pada langkah kelima	46
Gambar 3.22 Penjelasan AStar pada langkah keenam	46
Gambar 3.23 Penjelasan AStar dengan tree pada langkah keenam.....	48
Gambar 4.1 Tampilan Splashscrene	55
Gambar 4.2 Tampilan Menu.....	56
Gambar 4.3 Tampilan Tema pilihan permainan	57
Gambar 4.4 Tampilan misi Game.....	58
Gambar 4.5 Tampilan menu About	58
Gambar 4.6 Mulai Permainan.....	59
Gambar 4.7 Penduduk kampong arab mengikuti player.....	59
Gambar 4.8 Player memilih rumah penduduk yang salah.....	60
Gambar 4.9 Player memilih rumah yang benar dan mendapatkan bintang.....	61

Gambar 4.10 Tikus menghampiri NPC penduduk	61
Gambar 4.11 Tikus memakan NPC penduduk	62
Gambar 4.12 Tampilan Lose, tidak mendapatkan bintang	62
Gambar 4.13 Tampilan Lose, mendapatkan 1 bintang	63
Gambar 4.14 Tampilan Lose, mendapatkan 2 bintang	63
Gambar 4.15 Tampilan Lose, mendapatkan 3 bintang	64
Gambar 4.16 Tampilan Win, mendapatka 4 bintang	64
Gambar 4.17 Tampilan Win, mendapatkan 5 bintang	65



DAFTAR TABEL

Tabel 3.1 Jenis NPC beserta karakternya	29
Tabel 3.2 Tabel perhitungan langkah pertama	38
Tabel 3.3 Perhitungan langkah kedua	40
Tabel 3.4 Tabel perhitungan langkah ketiga	42
Tabel 3.5 Perhitungan Langkah keempat	44
Tabel 3.6 Tabel perhitungan langkah kelima	45
Tabel 3.7 Perhitungan langkah keenam	47
Tabel 4.1 Kebutuhan Perangkat keras(1)	49
Tabel 4.2 Kebutuhan Perangkat Keras(2)	50
Tabel 4.3 Kebutuhan perangkat lunak.....	50
Tabel 4.4 Keterangan Metode AStar	52
Tabel 4.5 Hasil Uji coba algoritma AStar	66
Tabel 4.6 Hasil Uji coba Game	68
Tabel 5.1 Hasil presentase.....	74

**IMPLEMENTASI METODE A* (A-STAR) UNTUK NPC
MUSUH PADA GAME 3D PEMBELAJARAN
KOSAKATA BAHASA ARAB**

SKRIPSI

Oleh :

BADZROTUL MUFIDA

NIM. 10650116



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2016**

**IMPLEMENTASI METODE A* (A-STAR) UNTUK NPC MUSUH PADA
GAME 3D PEMBELAJARAN KOSAKATA BAHASA ARAB**

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:

BADZROTUL MUFIDA

NIM.10650116

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2016**

HALAMAN PERSETUJUAN

**IMPLEMENTASI METODE A* (A-STAR) UNTUK NPC MUSUH PADA
GAME 3D PEMBELAJARAN KOSAKATA BAHASA ARAB**

SKRIPSI

Oleh :

Nama : Badzrotul Mufida

NIM : 10650116

Jurusan : Teknik Informatika

Fakultas : Sains dan Teknologi

Telah diperiksa dan disetujui untuk diuji:

Dosen Pembimbing I

Dosen Pembimbing II

Dr.M.Faisal,MT.
NIP. 19740510 200501 1 007

Hani Nurhayati,MT.
NIP. 19780625 200801 2 006

Tanggal, 30 Juni 2016

Mengetahui,

Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

HALAMAN PENGESAHAN

**IMPLEMENTASI METODE A* (A-STAR) UNTUK NPC MUSUH PADA
GAME 3D PEMBELAJARAN KOSAKATA BAHASA ARAB**

SKRIPSI

oleh:

BADZROTUL MUFIDA
NIM. 10650116

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

Tanggal, Juni 2016

Susunan Dewan Penguji:

Tanda Tangan

- | | | | |
|--------------------|-----------------------------------------------------------------|---|---|
| 1. Penguji Utama | : <u>Fresy Nugroho, M.T</u>
NIP. 19710722 201101 1 001 | (|) |
| 2. Ketua Penguji | : <u>Fachrul Kurniawan, M.MT</u>
NIP. 19771020 200901 1 001 | (|) |
| 3. Sekretaris | : <u>Dr. Muhammad Faisal, M.T</u>
NIP. 19740510 200501 1 007 | (|) |
| 4. Anggota Penguji | : <u>Hani Nurhayati, MT</u>
NIP. 19780625 200801 2 006 | (|) |

Mengetahui,

Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdiان
NIP. 19740424 200901 1 008

**PERNYATAAN
PERNYATAAN KEASLIAN PENELITIAN**

Saya yang bertandatangan di bawah ini:

Nama : BADZROTUL MUFIDA

NIM : 10650116

Fakultas/Jurusan : Sains dan Teknologi / Teknik Informatika

Judul Penelitian : Implementasi Metode A^* (*A-Star*) Untuk NPC Musuh
Pada *Game* 3D Pembelajaran Kosakata Bahasa Arab

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, Juni 2016

Yang membuat pernyataan,

Badzrotul Mufida

NIM. 10650116

MOTTO

“perangi segala ujian dengan usaha yang keras, karena dengan usaha yang keras maka kita akan merasakan rasa manis dari ujian tersebut”

HALAMAN PERSEMBAHAN

Puji syukur segala rahmat, taufik dan hidayah kepada Allah SWT atas terselesaikannya skripsi ini. Tak lupa ucapan terima kasih sebanyak-banyaknya kepada :

Orang tua yang selalu memberi dukungan dan bersabar kepada saya dalam setiap proses pengerjaan skripsi, terima kasih juga kepada kakak yang selalu mengingatkan dan memberi dorongan.

Dosen-dosen di jurusan teknik informatika yang telah membimbing dan memberikan ilmu selama menjalani studi di jurusan Teknik Informatika, dosen wali sekaligus dosen pembimbing Bapak Faisal dan Ibu Hani Nurhayati yang selalu memberikan bimbingan dan mensupport demi kesuksesan saya. Dan tidak lupa terima kasih kepada Bapak Eko budi minarno yang memberi dukungan kepada saya.

Terima kasih juga kepada arif firmansyah yang juga telah memberikan dukungan kepada saya.

Kepada teman-teman seperjuangan mbak ziah, febrí, agus, alif, syukron, odak, fani yang telah berjuang bersama-sama selama proses pengerjaan skripsi.

Teman-teman yang membantu dan memberi masukan dalam penggarapan skripsi febrí, joehani, alif, dan yutub tutorial terima kasih banyak.

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Alhamdulillah rabbil'Alamin penulis mengucapkan sukur 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 Raharjo, M.Si, selaku rektor UIN Maulana Malik Ibrahim Malang beserta seluruh staf. Dharma Bakti Bapak dan Ibu sekalian terhadap Universitas Islam Negeri Malang turut membesarkan dan mencerdaskan penulis.
2. Dr. Hj. Bayyinatul M., drh., M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang beserta seluruh staf. Bapak dan ibu sekalian sangat berjasa memupuk dan menumbuhkan semangat untuk maju kepada penulis.
3. Bapak Dr. Cahyo Crys dian, selaku Ketua Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang, yang sudah memberi banyak memberi pengetahuan, inspirasi dan pengalaman yang berharga.

4. Bapak Dr.M.Faisal, M.T, selaku dosen pembimbing I yang telah meluangkan waktu untuk membimbing, memotivasi, mengarahkan dan memberi masukan kepada penulis dalam pengerjaan skripsi ini hingga akhir.
5. Ibu Hani Nurhayati, M.T, selaku dosen pembimbing II yang juga senantiasa memberi masukan dan nasihat serta petunjuk dalam penyusunan skripsi ini.
6. Ayah, Ibu, dan Kakak serta keluarga besar saya tercinta yang selalu memberi dukungan yang tak terhingga serta do'a yang senantiasa mengiringi setiap langkah penulis.
7. Segenap Dosen Teknik Informatika yang telah memberikan bimbingan keilmuan kepada penulis selama masa studi.
8. Teman – teman seperjuangan Teknik Informatika 2010
9. 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.*

Wassalamu'alaikum Wr. Wb.

Malang, Juni 2016

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
LEMBAR PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
HALAMAN PERNYATAAN	v
MOTTO	vi
PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xv
ABSTRAK	xvi
ABSTRACT	xvii
ملخص	xviii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian.....	3
1.4 Manfaat Penelitian.....	4
BAB II KAJIAN PUSTAKA	5
2.1 Game (Permainan)	5
2.2 Pengertian Game	5
2.3 Elemen-elemen Game	6
2.4 Genre Game atau Jenis Game	7
2.5 Artificial Intelligence (AI).....	11
2.6 Algoritma A* (AStar).....	12
2.7 Fungsi Heuristik	15
2.8 Penelitian Terkait	16
2.9 Game Engine Unity3D	18
2.9.1 Unity Software	18
2.9.2 Fitur-fitur.....	19

2.10 Metode Penelitian.....	21
BAB III DESAIN DAN PERANCANGAN GAME	23
3.1 Pengembangan Game.....	23
3.2 Keterangan umum Game.....	23
3.3 Objek Penelitian	24
3.4 Story Board	24
3.5 Deskripsi masing-masing karakter	28
3.6 Deskripsi Gameplay game.....	30
3.7 Scoring.....	31
3.8 Pembelajaran dalam game	31
3.9 Finite State Machine	32
3.10 Perancangan A* (AStar).....	33
3.10.1 Simulasi A* (AStar).....	33
BAB IV HASIL DAN PEMBAHASAN	49
4.1 Implementasi Sistem.....	49
4.1.1 Kebutuhan Perangkat Keras.....	49
4.1.2 Kebutuhan Perangkat Lunak.....	50
4.2 Implementasi Algoritma A* pada Perilaku NPC.....	51
4.2.1 A* Sebagai Metode Perilaku Pencarian.....	51
4.3 Implementasi Aplikasi Game.....	55
4.3.1 Tampilan <i>Splashscrene</i>	55
4.3.2 Tampilan Menu Game	56
4.3.3 Tampilan Pilihan Tema Permainan.....	57
4.3.4 Tampilan Scene Help.....	57
4.3.5 Tampilan Scene About.....	58
4.3.6 Tampilan Permainan	59
4.4 Uji Coba.....	65
4.4.1 Uji Coba Algoritma	66
4.4.2 Uji Coba Game	68
4.5 Game Pembelajaran kosakata Bahasa Arab dalam Pandangan Islam.....	70
BAB V PENUTUP	73
5.1 Kesimpulan.....	73
5.2 Saran	74

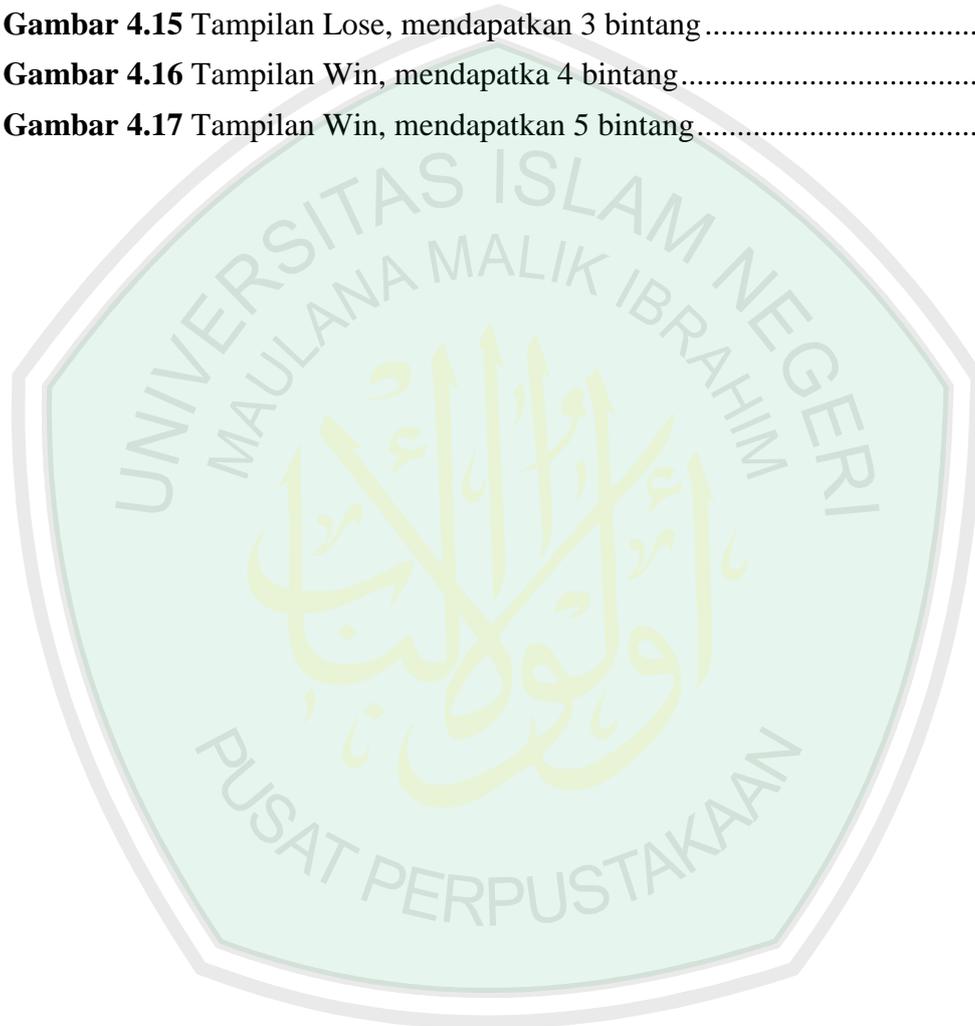
DAFTAR PUSTAKA	75
LAMPIRAN	78



DAFTAR GAMBAR

Gambar 2.1 Tahap metode penelitian.....	21
Gambar 3.1 avatar memasuki labirin.....	25
Gambar 3.2 avatar mencari NPC penduduk	25
Gambar 3.3 avatar mengantar pulang NPC penduduk	26
Gambar 3.4 NPC musuh mendekati dan memakan NPC penduduk	27
Gambar 3.5 avatar lose	27
Gambar 3.6 avatar win.....	27
Gambar 3.7 Najwa (Avatar)	28
Gambar 3.8 FSM penduduk “kampong Arab”	31
Gambar 3.9 FSM NPC musuh (tikus)	32
Gambar 3.10 Tahap awal pencarian algoritma A*	34
Gambar 3.11 Ilustrasi open list dan closed list.....	36
Gambar 3.12 Perhitungan menentukan nilai F	37
Gambar 3.13 Penjelasan A* menggunakan tree pada langkah pertama.....	39
Gambar 3.14 Penjelasan AStar pada langkah kedua	40
Gambar 3.15 Penjelasan AStar dengan tree pada langkah kedua	41
Gambar 3.16 Penjelasan AStar pada langkah ketiga.....	42
Gambar 3.17 PenjelasanAStar dengan tree pada langkah ketiga	43
Gambar 3.18 Penjelasan AStar pada langkah keempat	43
Gambar 3.19 Penjelasan AStar dengan tree pada langkah keempat	44
Gambar 3.20 Penjelasan AStar pada langkah kelima.....	45
Gambar 3.21 Penjelasan AStar dengan tree pada langkah kelima	46
Gambar 3.22 Penjelasan AStar pada langkah keenam	46
Gambar 3.23 Penjelasan AStar dengan tree pada langkah keenam.....	48
Gambar 4.1 Tampilan Splashscrene	55
Gambar 4.2 Tampilan Menu.....	56
Gambar 4.3 Tampilan Tema pilihan permainan	57
Gambar 4.4 Tampilan misi Game.....	58
Gambar 4.5 Tampilan menu About	58
Gambar 4.6 Mulai Permainan.....	59
Gambar 4.7 Penduduk kampong arab mengikuti player.....	59
Gambar 4.8 Player memilih rumah penduduk yang salah.....	60
Gambar 4.9 Player memilih rumah yang benar dan mendapatkan bintang.....	61

Gambar 4.10 Tikus menghampiri NPC penduduk	61
Gambar 4.11 Tikus memakan NPC penduduk	62
Gambar 4.12 Tampilan Lose, tidak mendapatkan bintang	62
Gambar 4.13 Tampilan Lose, mendapatkan 1 bintang	63
Gambar 4.14 Tampilan Lose, mendapatkan 2 bintang	63
Gambar 4.15 Tampilan Lose, mendapatkan 3 bintang	64
Gambar 4.16 Tampilan Win, mendapatka 4 bintang	64
Gambar 4.17 Tampilan Win, mendapatkan 5 bintang	65



DAFTAR TABEL

Tabel 3.1 Jenis NPC beserta karakternya	29
Tabel 3.2 Tabel perhitungan langkah pertama	38
Tabel 3.3 Perhitungan langkah kedua	40
Tabel 3.4 Tabel perhitungan langkah ketiga	42
Tabel 3.5 Perhitungan Langkah keempat	44
Tabel 3.6 Tabel perhitungan langkah kelima	45
Tabel 3.7 Perhitungan langkah keenam	47
Tabel 4.1 Kebutuhan Perangkat keras(1)	49
Tabel 4.2 Kebutuhan Perangkat Keras(2)	50
Tabel 4.3 Kebutuhan perangkat lunak.....	50
Tabel 4.4 Keterangan Metode AStar	52
Tabel 4.5 Hasil Uji coba algoritma AStar	66
Tabel 4.6 Hasil Uji coba Game	68
Tabel 5.1 Hasil presentase.....	74

ABSTRAK

Mufidah, Badzrotul. 2016. Implementasi Metode A* (A-Star) Untuk NPC Musuh Pada Game 3D Pembelajaran Kosakata Bahasa Arab. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Pembimbing: (I) Dr.M.Faisal,M.T. (II) Hani Nurhayati, M.T.

Kata Kunci: *Game 3D, Pathfinding, A-Star*

Bagi umat Islam khususnya, bahasa Arab merupakan bahasa yang sangat istimewa, karena kitab bagi seluruh umat Muslim menggunakan Bahasa Arab. Bahasa Arab wajib dipelajari bagi umat Muslim untuk mengetahui kandungan-kandungan dari kitab Al-Qur'an. Untuk mempelajari bahasa arab kita harus mengetahui tentang Nahwu, Sorof, dan juga kosakata-kosakata dari bahasa arab itu sendiri. Untuk itu dibuatlah sebuah *game* 3D pembelajaran kosakata bahasa arab yang ditujukan untuk pemula. Dengan metode pembelajaran seperti ini diharapkan dapat meningkatkan keinginan belajar bagi orang-orang yang ingin memulai belajar Bahasa Arab. Sehingga ketika kita bermain *game* kita tidak hanya mendapatkan kesenangan, tetapi kita juga mendapatkan Ilmu dari *game* bahasa arab ini.

Dalam *game* ini pemain akan dibawa ke pulau fantasi dimana di dalam pulau tersebut terdapat karakter-karakter penduduk pulau tersebut yang berbentuk angka dan buah-buahan. Dan pemain harus mengantarkan karakter tersebut ke dalam rumah karakter itu masing-masing. Supaya bisa meningkatkan keseruan dalam *game*, diterapkan algoritma *A-Star* pada pergerakan *NPC* musuh yang nantinya juga akan mengejar karakter penduduk, apabila *player* kalah cepat dengan *NPC* musuh maka *player* tidak akan mendapatkan *point*. Pengujian dilakukan pada perangkat *mobile* yang menggunakan *platform android*.

ABSTRACT

Mufidah, Badzrotul. 2016. **Method Implementation of A * (A-Star) For NPC Enemies in 3D Game of Arabic Vocabulary Learning**. Thesis. Department of Informatics Faculty of Science and Technology of the State Islamic University of Maulana Malik Ibrahim Malang.

Supervisor: (I) Dr.M.Faisal, M.T. (II) Hani Nurhayati, M.T.

Keywords: *3D Games, Pathfinding, A-Star*

For Muslims in particular, Arabic is a language that is very special, because the book for all Muslims use the Arabic. Learning Arabic is obligatory for Muslims to know the contents of the book of the Qur'an. To learn Arabic language need to know about *Nahwu, Sorof*, and also new vocabulary words from the Arabic language itself. It made a 3D game Arabic vocabulary learning that is intended for beginners. By learning methods is expected to increase the desire to learn for people who want to start learning Arabic. So when we play the game we do not only have fun, but we also gain knowledge of this Arabic game.

In this game players will be brought to the fantasy island in the island where the characters who shaped the island's population and the number of fruits. And the player must deliver the character into the character of the house respectively. In order to increase the excitement in the game, A-Star algorithm is applied on the movement of enemy NPC that will also pursue the character of the population, if the player is lost quickly with the enemy NPC player would not get the point. Tests conduct on mobile devices that use the Android *platform*.

ملخص

Mufidah, Badzrotul. 2016. **Method Implementation of A * (A-Star) For NPC Enemies in 3D Game of Arabic Vocabulary Learning**. Thesis. Department of Informatics Faculty of Science and Technology of the State Islamic University of Maulana Malik Ibrahim Malang.

Supervisor: (1) Dr.M.Faisal, M.T. (II) Hani Nurhayati, M.T.

Keywords: *3D Games, Pathfinding, A-Star*

للمسلمين على وجه الخصوص، اللغة العربية هي اللغة التي هي خاصة جدا، لأن كتاب لجميع المسلمين استخدام اللغة العربية. تعلم اللغة العربية واجبة على المسلمين لمعرفة محتويات الكتاب من القرآن الكريم. لتعلم اللغة العربية نحتاج إلى معرفته عن النهووالصرف، وكذلك المفردات الجديدة من اللغة العربية نفسها. لذلك جعل لعبة 3D تعلم المفردات العربية التي تهدف للمبتدئين. عن طريق أساليب التعلم حيث من المتوقع أن تزيد من الرغبة في التعلم للأشخاص الذين يريدون للبدء في تعلم اللغة العربية. حتى عندما نلعب لعبة ونحن لم يكن لديك متعة فقط، ولكن نحن أيضا اكتساب المعرفة العربية هذه اللعبة. في هذه اللعبة سوف يتم التعاقد مع لاعبين للجزيرة الخيال في الجزيرة حيث هناك شخصيات شكل سكان الجزيرة وعدد من الفواكه. ويجب على اللاعب تسليم حرف في حرف من البيت على التوالي. من أجل زيادة الإثارة في اللعبة، يتم تطبيق النجوم الخوارزمية على حركة الشخصيات العدو الذي ستسعى أيضا طبيعة السكان، إذا كان لاعب سرعان ما فقدت مع لاعب مجلس الشعب العدو لن تحصل على هذه النقطة. التجارب التي أجريت على الأجهزة المحمولة التي تستخدم منصة *platform* أندرو

BAB I

PENDAHULUAN

1.1. Latar Belakang

Belajar merupakan suatu hal yang sangat penting, terutama dalam belajar Agama, di dalam Agama Islam tentu tidak lepas dari pembelajaran bahasa Arab, hal ini dikarenakan sumber utama dari Agama Islam adalah Al-Qur'an dan Hadist, dan bahasa asli dari Al-Qur'an dan Hadist itu sendiri adalah bahasa Arab. Untuk itu bahasa Arab merupakan salah satu yang harus dipelajari, dikuasai atau minimal dipahami oleh seorang Muslim agar tidak menyimpang dan memahami hukum-hukum Islam.

Tentang pentingnya belajar bahasa Arab Umar Bin Khattab Rahiyallohu'anhun berkata:

“Belajarlah bahasa Arab, karena sesungguhnya bahasa Arab itu adalah sebagian dari agama kalian”.

Dengan berkembangnya zaman, kini kita dapat membuat suatu pembelajaran menjadi lebih menarik. Belajar kini tidak hanya dengan melalui buku saja, meskipun membaca buku masih sangat diperlukan, kini siapa saja bisa mencicipi praktisnya teknologi. Dengan perkembangan teknologi sebuah pembelajaran kini tidak lagi membosankan tetapi bisa sangat menyenangkan. Dengan kemudahan-kemudahan saat ini, diharapkan umat manusia dapat memanfaatkan dan tidak menyia-nyiakannya.

Seperti yang telah kita ketahui *game* merupakan sarana untuk bermain, apapun bentuk dari *game* itu sendiri, mulai dari *game* yang sifatnya sederhana hingga *game* yang paling *modern*. Kita dapat menemukan *game* dalam *PC*, *website* bahkan *handpone*. Banyak anak-anak bahkan orang dewasa yang sangat tertarik dengan *game*. Menurut *john beck* dan *Mitchell wale* “*game* adalah penarik perhatian yang telah terbukti”.

Pengaruh yang ditimbulkan *game* memang tidak selamanya baik, karena memang pada dasarnya *game* adalah sarana untuk hiburan. Ada juga beberapa orang yang berperilaku buruk yang di sebabkan oleh *game*, tetapi itu juga kembali kepada masing-masing orang yang menjalankannya. Ada beberapa *game* yang buruk yang sama sekali tidak mendatangkan manfaat bagi pemain, bahkan dapat merusak moral, akhlak dan syari’at.

Telah disebutkan dalam surat Al-Qashash ayat 77:

وَابْتَغِ فِيمَا آتَاكَ اللَّهُ الدَّارَ الْآخِرَةَ وَلَا تَنْسَ نَصِيبَكَ مِنَ الدُّنْيَا وَأَحْسِنْ كَمَا أَحْسَنَ اللَّهُ إِلَيْكَ
وَلَا تَبْغِ الْفُسَادَ فِي الْأَرْضِ ۚ إِنَّ اللَّهَ لَا يُحِبُّ الْمُفْسِدِينَ {٧٧}

“Dan carilah pada apa yang telah dianugerahkan Allah kepadamu (kebahagiaan) negeri Akhirat, dan janganlah kamu melupakan kebahagiaanmu dari (kenikmatan) duniawi dan berbuat baiklah (kepada orang lain) sebagaimana Allah telah berbuat baik, kepadamu, dan janganlah kamu berbuat kerusakan di (muka) bumi. Sesungguhnya Allah tidak menyukai orang-orang yang berbuat kerusakan”.(Al-Qashash : 77)

Baik dan buruknya sebuah *game* memang berpengaruh bagi penikmat *game*, tinggal bagaimana kita sebagai umat manusia untuk mengimbangi perkembangan

game-game saat ini. Tentunya masih banyak juga *game* yang bermanfaat bagi pemain dan memberikan pengaruh baik bagi pemain.

1.2. Rumusan Masalah

Berikut rumusan masalah pada penelitian ini adalah bagaimana mengimplementasikan Algoritma A^* (*AStar*) pada *game* “jelajah kampoeng Arab” untuk melakukan proses pencarian rute terpendek pada *NPC* musuh?

1.3. Batasan Masalah

Untuk menghindari kemungkinan meluasnya pembahasan, maka dilakukan batasan-batasan masalah sebagai berikut:

- a. Untuk pembelajaran bahasa Arab berupa kosakata dasar bahasa Arab
- b. *Game* yang dibangun merupakan *game single player*
- c. *Game* ini bergenre *game Adventure*
- d. Algoritma A^* (*AStar*) di terapkan pada *NPC* musuh
- e. *Game* ini dibuat dengan *Game Engine Unity3D*

1.4. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah:

- a. Algoritma A^* (*A-Star*) dapat diterapkan dan digunakan untuk perilaku *NPC* musuh

- b. Menerapkan *game* “jelajah kampoeng Arab” dengan menambahkan *NPC* penduduk yang berbentuk buah dan angka sebagai pembelajaran kosakata bahasa Arab.

1.5. Manfaat Penelitian

- 1) Bagi peneliti :

Menambah wawasan dan keterampilan dalam membangun *game* beredukasi, dan memperluas tentang pembelajaran bahasa Arab.

- 2) Bagi Umum :

Sebagai hiburan yang mendidik yang dapat di gunakan untuk membantu dalam mempelajari bahasa Arab.

BAB II

KAJIAN PUSTAKA

2.1 Game (Permainan)

Game merupakan Permainan yang terdiri atas sekumpulan peraturan yang membangun situasi bersaing dari dua sampai beberapa orang atau kelompok dengan memilih strategi yang dibangun dengan untuk memaksimalkan kemampuan sendiri atau pun meminimalkan kemenangan lawan (Neumann, 1953).

Definisi *game* menurut Agustinus Nilwan *Game* merupakan permainan komputer yang dibuat dengan teknik dan metode animasi (Nilwan, 1995).

Game yang di maksud dalam penelitian ini adalah *game* yang diterapkan pada aplikasi *handpone android*.

2.2 Pengertian Game

Permainan (*game*) pertama kali ditemukan oleh sekelompok ahli Matematika pada tahun 1944. Teori itu ditemukan oleh John von Neumann dan Oskar Morgenstern yang berisi:

“Permainan terdiri atas sekumpulan peraturan yang membangun situasi bersaing dari dua sampai beberapa orang atau sekelompok dengan memilih strategi yang dibangun untuk memaksimalkan kemenangan sendiri atau pun untuk meminimalkan kemenangan lawan. Peraturan-peraturan menentukan kemungkinan tindakan untuk setiap pemain, sejumlah keterangan diterima setiap

pemain sebagai kemajuan bermain, dan sejumlah kemenangan atau kekalahan dalam berbagai situasi”.

Menurut Agustinus Nilwan dalam bukunya “Pemrograman Animasi dan *Game Profesional*” terbitan Elex Media Komputindo,

“Game merupakan permainan computer yang dibuat dengan teknik dan metode animasi. Jika ingin mendalami penggunaan animasi haruslah memahami pembuatan game. Atau jika ingin membuat game, maka haruslah memahami teknik dan metode animasi, sebab keduanya saling berkaitan”.

2.3 Elemen-elemen *game*

Menurut Teresa Dillon (futurelab.com, 2005) elemen-elemen dasar sebuah *game* adalah:

1. *Game Rule*

Game rule merupakan aturan perintah, cara menjalankan, fungsi obyek dan karakter di dunia permainan Dunia *Game*. Dunia *game* bisa berupa pulau, dunia khayal, dan tempat-tempat lain yang sejenis yang dipakai sebagai *setting* tempat dalam permainan *game*.

2. *Plot*

Plot biasanya berisi informasi tentang hal-hal yang akan dilakukan oleh *player* dalam *game* dan secara detail, perintah tentang hal yang harus dicapai dalam *game*.

3. *Theme*

Di dalam biasanya ada pesan moral yang akan disampaikan

4. *Character*

Pemain sebagai karakter utama maupun karakter yang lain yang memiliki ciri dan sifat tertentu.

5. *Object*

Merupakan sebuah hal yang penting dan biasanya digunakan pemain untuk memecahkan masalah, adakalanya pemain harus punya keahlian dan pengetahuan untuk bisa memainkannya.

6. *Text, grafik dan sound*

Game biasanya merupakan kombinasi dari media *teks*, grafik maupun suara, walaupun tidak harus semuanya ada dalam permainan *game*.

7. *Animation*

Animation ini selalu melekat pada dunia *game*, khususnya untuk gerakan karakter-karakter yang ada dalam *game*, property dari objek.

8. *User interface*

Merupakan fitur-fitur yang mengkomunikasikan *user* dengan *game*.

2.4 Genre game atau jenis game

Jenis *game* biasa disebut dengan istilah *genre game*. Menurut Henry (2010:112) format sebuah *game* bisa murni sebuah *genre* atau bisa merupakan campuran (*hybrid*) dari beberapa *genre* lain. Jenis-jenis *game* yang ada menurut Henry (2010:111) adalah sebagai berikut:

1. *Maze game*

Jenis *game* ini biasanya menggunakan *maze* sebagai *setting* atau latar *game*.

Jenis *game maze* ini termasuk jenis *game* yang paling awal muncul. Contoh *game* ini adalah *game pacman* dan *digger*

2. *Board game*

Game jenis ini sama dengan *game board* tradisional seperti monopoli.

Hanya saja permainan tradisional ini dimainkan melalui *computer*.

3. *Card game*

Jenis *game* kartu juga tidak jauh berbeda dari *game* tradisional aslinya.

Namun, tampilanya lebih bervariasi dari versi tradisional. *Game* ini juga termasuk *game* yang awal muncul. Contoh *game* ini adalah *solitaire* dan *hearts*.

4. *Battle card game*

Contoh *game* ini yang populer yaitu *battle card pokemon*. *Game* ini jarang di temukan di Indonesia. Film kartun yang bercerita tentang permainan *battle card* ini pernah ditayangkan di stasiun televisi Indonesia.

5. *Quiz game*

Game jenis ini merupakan *game* dengan bentuk kuis. Contoh *quiz game* yang pernah beredar yaitu *game kuis who want to be millionaire*

6. *Shoot them up*

Game jenis ini biasanya musuh berbentuk pesawat atau bentuk lain yang datang dari arah kanan, kiri atau atas yang harus di tembak sebanyak dan

secepat mungkin. Dulu *game* ini berbentuk dua dimensi (2D), tetapi sekarang sudah berkembang dan menggunakan efek tiga dimensi (3D).

7. *Side scroller game*

Saat pertama kali muncul *game* ini berbentuk 2D. Sekarang sudah banyak yang dibuat dengan efek 3D. Pada *game* jenis ini pemain diharuskan bergerak searah di alur yang disediakan. Dia diharuskan untuk berjalan, melompat, merunduk serta menghindari rintangan-rintangan. Contoh *game* ini yang populer yaitu *Mario bros* dan *Prince of Persia*.

8. *Fighting game*

Jenis *game* ini sesuai dengan namanya berisi tentang pertarungan. Contoh *game* ini yaitu *street fighter*, *samurai showdown*, *virtual fighter* dan kungfu

9. *Racing game*

Racing game adalah *game* tentang balapan. Contoh *game* ini yaitu *need for speed underground* dan *toca race driver*.

10. *Turn-based strategy game*

Pemain dalam *game* ini melakukan gerakan setelah pemain lain melakukan gerakan jadi saling bergantian. Contoh *game* yang terkenal adalah *mpir* dan *civilization*.

11. *Real-time strategy game*

Game ini seperti *game turn-based strategy (RTS)*, namun pada *game* ini pemain tidak perlu menunggu pemain lain. Pemain tercepatlah yang akan menang. Contoh *game* ini yaitu *Warcraft*.

12. *SIM*

Game genre ini merupakan bentuk permainan simulasi. Di sini pemain membangun sebuah area, Kota, Negara atau koloni. Contoh *game* ini yaitu *ship simulator*, *train simulator*, dan *crane simulator*.

13. *First person shooter*

Disebut *first person shooter* karena pandangan pemain adalah pandangan orang pertama. Banyak baku tembak dan *game* ini mengutamakan kecepatan gerakan. Contoh *game* ini yaitu *game counterstrike* dan *doom*.

14. *First person shooter 3D vehicle based*

Game ini sama dengan *FPS* hanya saja pandangan pemain bukan dari orang pertama, tetapi dari kendaraan atau mesin yang digunakan. Kendaraan itu biasa berupa *tank* atau kapal.

15. *Third person 3D games*

Game ini juga hampir sama dengan *FPS* hanya sudut pandang pemain merupakan sudut pandang orang ketiga.

16. *Role playing game*

Jenis *game* ini pemainnya memainkan sebuah tokoh atau karakter. Biasanya ada alur cerita yang harus dijalankan. Contoh *game* ini adalah *legacy of kain*, *blade of sword*, dan *beyond divinity*.

17. *Adventure game*

Adventure game merupakan *genre game* petualangan. Di sepanjang perjalanan pemain akan menemukan peralatan yang akan disimpan dan

berguna sebagai petunjuk perjalanan. Contoh game ini yaitu *sam and max* atau *byond and evil*

18. *Educational and edutainment*

Game ini lebih mengacu pada isi dan tujuan dari *game*. *Game* ini bertujuan untuk memancing minat belajar anak sambil bermain. Contoh *game* ini adalah *game* boby bola.

19. *Sports*

Jenis *game* ini memiliki tema olahraga. *Game* yang mengetengahkan *genre* olahraga disebut *sport game*.

2.5 *Artificial Intelligence (AI)*

Kecerdasan buatan sering disebutkan juga dengan *Artificial Intelligence (AI)* merupakan salah satu bagian ilmu *computer* yang mempelajari tentang bagaimana caranya agar *computer* dapat melakukan pekerjaan seperti yang dapat dilakukan oleh manusia. Pada awal diciptakanya, *computer* hanya difungsikan sebagai alat hitung saja. Namun seiring dengan berkembangnya zaman, teknologi *computer* semakin ditingkatkan dan peran *computer* semakin mendominasi kehidupan umat manusia. *Computer* tidak lagi hanya digunakan sebagai alat hitung, melainkan *computer* diharapkan dapat diberdayakan untuk mengerjakan hal-hal yang dapat memudahkan pekerjaan manusia.

Manusia bisa menjadi pandai dalam menyelesaikan segala permasalahan di dunia ini karena manusia mempunyai pengetahuan dan pengalaman. Pengetahuan dapat diperoleh dari proses belajar. Semakin banyak bekal pengetahuan yang

dimiliki oleh seseorang tentu saja diharapkan akan lebih baik dalam menyelesaikan permasalahan. Namun bekal pengetahuan saja tidak cukup, manusia juga diberi akal untuk melakukan penalaran, mengambil kesimpulan berdasarkan pengetahuan dan pengalaman yang mereka miliki. Tanpa memiliki kemampuan menalar yang baik, manusia dengan segudang pengalaman dan pengetahuan tidak akan dapat menyelesaikan masalah dengan baik, demikian pula dengan kemampuan menalar yang sangat baik, namun tanpa bekal pengetahuan dan pengalaman yang memadai, manusia juga tidak akan bias menyelesaikan masalah dengan baik.

Agar *computer* bisa bertindak seperti apa yang dapat dilakukan manusia, maka *computer* juga harus diberi bekal pengetahuan dan mempunyai kemampuan untuk menalar. Untuk itu *AI* akan mencoba untuk memberikan beberapa metode untuk membekali *computer* dengan kedua komponen tersebut agar *computer* bisa menjadi mesin yang pintar.

2.6 Algoritma A^* (*AStar*)

Algoritma ini pertama kali diperkenalkan pada 1968 oleh Peter Hart, Nils Nilsson, dan Bertram Raphael. Dalam ilmu *computer*, A^* (yang diucapkan dengan “*A Star*”) merupakan salah satu algoritma pencarian *graph* terbaik yang mampu menemukan jalur dengan biaya pengeluaran paling sedikit dari titik permulaan yang diberikan sampai ke titik tujuan yang diharapkan (dari satu atau lebih mungkin tujuan).

Algoritma A^* merupakan perbaikan dari metode *BFS* (*Best First Search*) dengan memodifikasi fungsi *heuristiknya*. A^* akan meminimumkan total biaya

lintasan yang terdapat di metode *BFS*. Pada kondisi yang tepat, *A** akan memberikan solusi yang terbaik dalam waktu yang optimal. Pada pencarian *rute* kasus sederhana, dimana tidak terdapat halangan pada peta, *AStar* bekerja secepat dan seefisien *BFS*. Pada kasus peta dengan halangan, *AStar* dapat menemukan solusi *rute* tanpa terjebak oleh halangan yang ada.

Beberapa *terminology* dasar yang terdapat pada algoritma *AStar* adalah *starting point*, simpul (*nodes*), *A*, *open list*, *closed list*, harga (*cost*), halangan (*unwalkable*). *Starting point* adalah sebuah *terminology* untuk posisi awal sebuah benda. *A* adalah *node* yang sedang dijalankan dalam algoritma pencarian jalan terpendek. *Node* adalah petak-petak kecil sebagai representasi dari area *path finding*. Bentuknya dapat berupa persegi, lingkaran, maupun segitiga. *Open list* adalah tempat menyimpan data *node* yang mungkin diakses dari *starting point* maupun simpul yang sedang dijalankan. *Closed list* adalah tempat menyimpan data simpul sebelum *A* yang juga merupakan sebagai dari jalur terpendek yang telah berhasil didapatkan.

Harga (*F*) adalah nilai yang diperbolehkan dari penjumlahan nilai *G*, jumlah nilai tiap simpul dalam jalur terpendek dari *starting point* ke *A*, dan *H*, jumlah nilai perkiraan dari sebuah simpul ke simpul tujuan. Simpul tujuan adalah simpul yang dituju. Rintangan adalah sebuah atribut yang menyatakan bahwa sebuah simpul tidak dapat dilalui oleh *A*.

AStar memiliki 2 fungsi utama dalam menentukan solusi terbaik. Fungsi pertama disebut sebagai $g(n)$ merupakan fungsi yang digunakan untuk menghitung total *cost* yang dibutuhkan dari *starting point* menuju *node* tertentu. Fungsi kedua

yang biasa disebut sebagai $h(n)$ merupakan fungsi perkiraan total *cost* yang diperkirakan dari suatu *node* ke *node* akhir.

Pada *A Star*, setiap *node* dari *node* awal ditelusuri kemudian dihitung *cost* dari tiap-tiap *node* dan dimasukkan ke tabel prioritas. *Node* dengan *cost* paling rendah akan diberikan tingkat prioritas paling tinggi. Kemudian pencarian dilanjutkan pada *node* dengan nilai prioritas tertinggi pada *table*.

$$F(n) = g(n) + h(n)$$

Dengan :

n = posisi koordinat *node*

$f(n)$ = fungsi evaluasi

$g(n)$ = biaya (*cost*) yang sudah dikeluarkan dari keadaan sampai keadaan n

$h(n)$ = estimasi biaya untuk sampai pada suatu tujuan mulai dari n

nilai F adalah *cost* perkiraan suatu *node* yang teridentifikasi. Nilai F merupakan hasil dari $f(n)$. Nilai G hasil dari fungsi $g(n)$, adalah banyaknya langkah yang diperlukan untuk menuju ke *node* sekarang. Setiap *node* (*node*) harus memiliki informasi nilai $h(n)$, yaitu estimasi harga *node* tersebut dihitung dari *node* tujuan yang hasilnya menjadi nilai H .

Node dengan nilai terendah merupakan solusi terbaik untuk diperiksa pertama kali pada $g(n) + h(n)$. Dengan fungsi *heuristic* yang memenuhi kondisi tersebut, maka pencarian dengan algoritma *A Star* dapat optimal.

2.7 Fungsi *Heuristik*

Heuristic adalah sebuah teknik yang mengembangkan efisiensi dalam proses pencarian, namun dengan kemungkinan mengorbankan kelengkapan

(*completeness*). Fungsi *heuristic* digunakan untuk mengevaluasi keadaan-keadaan problema individual dan menentukan seberapa jauh hal tersebut dapat digunakan untuk mendapatkan solusi yang diinginkan.

A *Star* sebagai algoritma pencarian yang menggunakan fungsi *heuristic* untuk ‘menuntun’ pencarian *route*, khususnya dalam hal pengembangan dan pemeriksaan *node-node* pada peta (Stuart dan Peter, 2003). terdapat beberapa fungsi *heuristic* umum yang bisa dipakai untuk algoritma *A** ini. Salah satunya adalah yang dikenal dengan istilah ‘*manhattan distance*’. Fungsi *heuristic* ini digunakan untuk kasus dimana pergerakan pada peta hanya lurus (horizontal atau vertikal), tidak diperbolehkan pergerakan diagonal (Arnold Nugroho, 2009).

Perhitungan nilai *heuristic* untuk node ke-*n* menggunakan *manhattan distance* adalah sebagai berikut:

$$h(n) = (\text{abs}(n.x - \text{goal}.x) + \text{abs}(n.y - \text{goal}.y))$$

dimana $h(n)$ adalah nilai *heuristic* untuk *node* *n*, dan *goal* adalah *node* tujuan. Jika pergerakan diagonal pada peta diperbolehkan, maka digunakan fungsi *heuristic* selain *manhattan distance*.

Untuk mendekati kenyataan, *cost* untuk perpindahan *node* secara diagonal dan orthogonal dibedakan *Cost* diagonal adalah 1,4 kali *cost* perpindahan secara orthogonal, maka fungsi *heuristic* yang digunakan adalah sebagai berikut:

$$h_{\text{diagonal}}(n) = \min(\text{abs}(n.x - \text{goal}.x) + \text{abs}(n.y - \text{goal}.y))$$

$$h_{\text{orthogonal}}(n) = (\text{abs}(n.x - \text{goal}.x) + \text{abs}(n.y - \text{goal}.y))$$

$$h(n) = h_{\text{diagonal}}(n) + (h_{\text{orthogonal}}(n) - (2 * h_{\text{diagonal}}(n)))$$

Dimana $h_{diagonal}(n)$ adalah banyaknya langkah diagonal yang bisa diambil untuk mencapai *goal* dari *node* n . $h_{orthogonal}$ adalah banyaknya langkah lurus yang bisa diambil untuk mencapai *goal* dari *node* n . Nilai *heuristic* kemudian diperoleh dari $h_{diagonal}(n)$ ditambah dengan selisih $h_{orthogonal}(n)$ dengan dua kali $h_{diagonal}(n)$. Dengan kata lain, jumlah langkah diagonal kali *cost* diagonal ditambah jumlah langkah lurus yang masih bisa diambil dikali *cost* pergerakan lurus (Zou, dkk).

2.8 Penelitian Terkait

Galih Bonifatius dkk, 2013, Skripsi tentang analisis implementasi algoritma A^* (*AStar*) pada *game RPG (Role Player Game)* 3D sebagai dasar pergerakan *NPC (Non Player Karakter)* mendekati *player* untuk meningkatkan realitas *game word*. Dalam penelitian ini digunakan dua uji coba untuk mengukur kinerja system. Pada uji coba pertama didapatkan bahwa A^* dapat diimplementasikan pada *game* 3D sebagai algoritma pencarian jejak karakter musuh saat mengejar karakter *player* dengan memberikan pemicu. Pemicu digunakan untuk menggerakkan agar karakter musuh memanggil fungsi A^* dan menggunakan *waypoint* sebagai pengganti *node* dari algoritma A^* ini. Berdasarkan uji coba kedua, dapat disimpulkan bahwa selama *game* dengan implementasi A^* ini selalu dapat memberikan solusi jalur untuk karakter tersebut berada selama karakter musuh masih terpicu, dan 75% dari jalur yang dihasilkan merupakan jalur optimal (memiliki jarak terpendek).

Winanti Wina dkk, Desember 2013, Skripsi tentang analisis pengaruh penggunaan nilai *heuristic* terhadap performansi algoritma A^* pada *game*

pathfinding. Berdasarkan hasil pengujian guna mengetahui performansi waktu pencarian, jarak dan simpul yang diperiksa dari titik awal menuju titik tujuan dengan algoritma A^* (*AStar*) yang diterapkan dalam pencarian jalan terpendek pada *game pathfinding*, maka jalan yang dihasilkan merupakan jalan terpendek karena simpul yang diperiksa *relative* banyak dan memerlukan waktu pencarian yang relatif lama pula.

Parameter dengan nilai heuristik paling kecil akan menghasilkan waktu dan simpul yang diperiksa besar sebaliknya nilai heuristik paling besar akan menghasilkan waktu dan simpul yang diperiksa kecil. Jadi, lamanya proses pencarian dan banyaknya simpul yang diperiksa untuk pencarian jalan tergantung pada jarak antara titik awal dan titik tujuan. Hasil algoritma A^* memberikan hasil pencarian jalan yang optimal.

Ramadhani Aristama, 2008, menggerakkan karakter *game* menggunakan algoritma *Breadth-First Search* (BFS) dan algoritma A^* (*AStar*). Sekripsi ini menjelaskan bahwa algoritma A^* merupakan algoritma yang lebih baik dari pada algoritma BFS karena bisa menemukan jalan terpendek (*shortest path*) dalam masalah ini. Selain itu, algoritma A^* juga merupakan algoritma terbaik dalam ilmu computer (*computer science*) dalam pencarian *graf* untuk mencari jalan dalam *cost* terkecil.

Joehani Abdillah Aslami, 2015, sekripsi tentang Implementasi Algoritma A^* (*AStar*) sebagai dasar pergerakan NPC (*Non Player Character*) mendekati *Player* untuk meningkatkan realitas *game* pembelajaran kosakata bahasa arab. Dalam kesimpulan ini mengambil kesimpulan supaya bisa meningkatkan realitas *game*,

diterapkan algoritma A* (*Astar*) pada pergerakan NPC (*Non Player Character*) agar bisa mengejar pemain. NPC akan selalu mengejar pemain melalui jalur terpendek dan apabila telah mencapai pemain maka *Health point* akan berkurang. *Game* yang dibuat berbasis desktop yang bisa dijalankan di beberapa sistem operasi.

Dari beberapa pustaka yang telah dijelaskan di atas dapat disimpulkan bahwa algoritma A* merupakan algoritma yang cukup baik dalam menentukan jalan terpendek. Maka, apabila diterapkan pada NPC (*Non Player Character*) sebuah *game* akan lebih efektif dalam mencari jalan seperti yang kita inginkan melalui jalan terpendek tersebut.

2.9 Game Engine Unity3D

2.9.1 Unity Software

Unity merupakan suatu aplikasi yang digunakan untuk mengembangkan *game multi platform* yang didesain untuk mudah digunakan. *Unity* itu bagus dan penuh perpaduan dengan aplikasi yang profesional. Editor pada *Unity* dibuat dengan *user interface* yang sederhana. *Unity* cocok dengan versi 64-bit dan dapat beroperasi pada Mac OS x dan windows dan dapat menghasilkan *game* untuk Mac, Windows, Wii, iPhone, iPad dan Android. *Unity* secara rinci dapat digunakan untuk membuat *video game 3D*, *real time animasi 3D* dan visualisasi arsitektur dan ini serupa yang interaktif lainnya. *Server asset* dari *Unity* dapat digunakan semua *scripts* dan *asset game* sebagai solusi dari versi control dan dapat

mendukung proyek yang terdiri atas banyak giga bytes dan ribuan dari file *multy-mega byte*.

2.9.2 Fitur-fitur

a. *Rendering*

Graphics engine yang digunakan adalah *Direct3D* (*Windows, Xbox 360*), *OpenGL* (*Mac, Windows, Linux, PS3*), *OpenGL ES* (*Android, iOS*), dan *proprietary APIs* (*Wii*). Adapun kemungkinan untuk *bump mapping*, *reflection mapping*, *parallax mapping*, *screen space ambient occlusion (SSAO)*, *dynamic shadows using shadow maps*, *render-to-texture* and *full-screen post-proccesing effects*.

Unity dapat mengambil format desain dari 3ds Max, Maya, Softimage, Blender, modo, ZBrush, Cinema 4D, Cheetah3D, adobe Photoshop, adobe Fireworks and Allegorithmic Substanc.

b. *Scripting*

Script game engine dibuat dengan Mono 2.6, sebuah implementasi *open-source* dari .NET Framework. Programmer dapat menggunakan Unity Script (bahasa terkustomisasi yang terinspirasi dari syntax ECMAScript, dalam bentuk JavaCsript), C#, atau Boo (terinspirasi dari Syntax bahasa pemrograman phyton).

c. *Asset Tracking*

Unity juga menyertakan Server Unity Asset – sebuah solusi terkontrol untuk *developer game asset* dan *script*.

d. *Platforms*

Unity support pengembangan ke berbagai platform. Dimana *project, developer* memiliki control untuk mengirim perangkat mobile, web browser, desktop, and consol. Utity juga mengijinkan spesifikasi kompresi texture dan pengaturan resolusi di setiap *platform* yang didukung.

Saat ini *platform* yang didukung oleh BlackBerry 10, windows 8, windows Phone 8, Windows, Mac, Linux, Android, iOS, Unity Wb Player, Adob Flash, PlayStation 3, Xbox 360, Wii U and Wii.

e. *Asset Store*

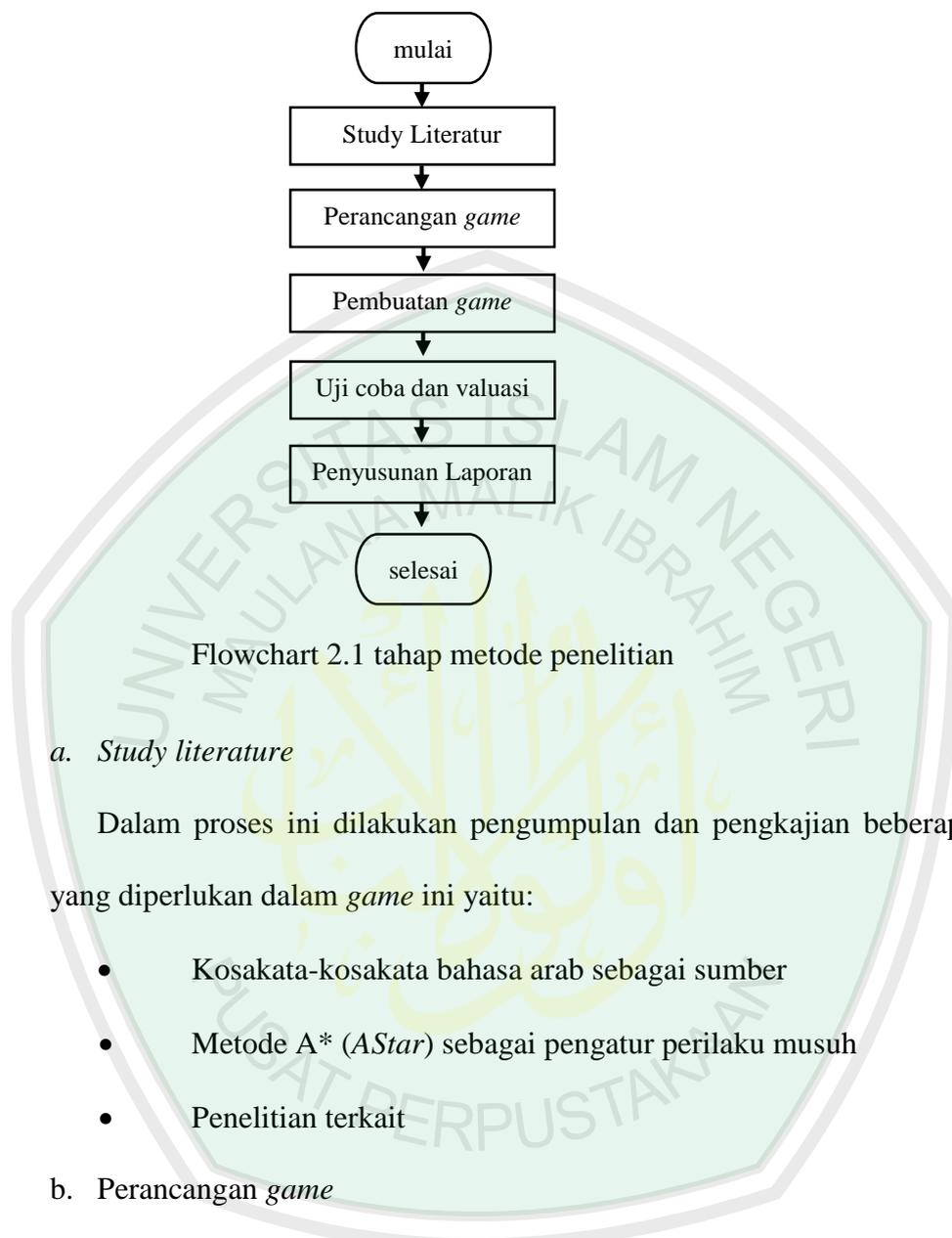
Diluncurkan November 2010, *Unity Asset Store* adalah *resource* yang hadir di Unity editor. *Asset store* terdiri dari koleksi lebih dari 4,400 *asset packages*, beserta 3D models, *textures* dan materials, *system partikel, music* dan efek suara, tutorial dan *project, scripting package*, editor xtensions dan servis online.

f. *Physics*

Unity juga memiliki *support built-in* untuk *PhysX physics engine* (sejak Unity 3.0) dari Nvidia (sebelumnya Ageia) dengan menambahkan kemampuan untuk simulasi *real-time cloth* pada arbitrary dan *skinned meshes, thick ray cast*, dan *collision layers*.

2.10 Metode Penelitian

Terdapat beberapa tahap dalam metode penelitian, yaitu:



a. *Study literature*

Dalam proses ini dilakukan pengumpulan dan pengkajian beberapa data yang diperlukan dalam *game* ini yaitu:

- Kosakata-kosakata bahasa arab sebagai sumber
- Metode A* (*AStar*) sebagai pengatur perilaku musuh
- Penelitian terkait

b. *Perancangan game*

Dalam proses ini akan dilakukan perancangan baik dari *story board*, bahasa pemrograman, pengumpulan dan pembuatan kebutuhan audio visual, perancangan kebutuhan *game* (status, barang dalam *game*, dan sebagainya), dan segala yang akan dibutuhkan dalam *game* ini.

c. *Pembuatan game*

Dalam proses ini akan dibuat *game* menggunakan *Unity3D engine* dengan difokuskan menggunakan bahasa *C#* dan *JavaScript*

d. Uji coba dan evaluasi

Dalam proses ini akan dilakukan serangkaian uji coba baik sebelum *game* jadi maupun setelah, untuk mencegah terjadinya kesalahan dalam *game* ini.

e. Penyusunan laporan

Dalam proses ini dilakukan dokumentasi berupa laporan skripsi sebagai bukti penelitian telah berhasil menyelesaikan *game* ini dan diharapkan bisa digunakan untuk penelitian yang lainnya.



BAB III

DESAIN DAN PERANCANGAN GAME

3.1. Pengembangan game

Game adventure atau yang biasa disebut dengan *game* petualangan berbasis *game Artificial Intelligence* untuk pembelajaran bahasa arab ini adalah sebuah *game* edukasi bahasa arab yang diimplementasikan pada tiap-tiap tantangan yang diberikan sehingga puncak dari permainan ini di harapkan pemain mendapatkan tambahan atau bekal kosakata-kosakata dasar dalam pembelajaran bahasa arab.

3.2. Keterangan umum Game

Game ini merupakan sebuah permainan yang dirancang untuk para pemain yang ingin memulai belajar bahasa arab, karena didalam permainan ini pemain di kenalkan beberapa kosakata dasar bahasa arab, seperti angka-angka dan buah-buahan. *Game* ini mempunyai *setting* di sebuah pulau fantasi, dan pemain akan berpetualang di sebuah pulau tersebut, pemain akan bertemu dengan musuh dan karakter-karakter yang berbeda dan pemain harus mendapatkan karakter-karakter tersebut. Dan dalam permainan ini terdapat dua level yang didalamnya mencakup karakter buah-buahan, sayur-sayuran dan angka.

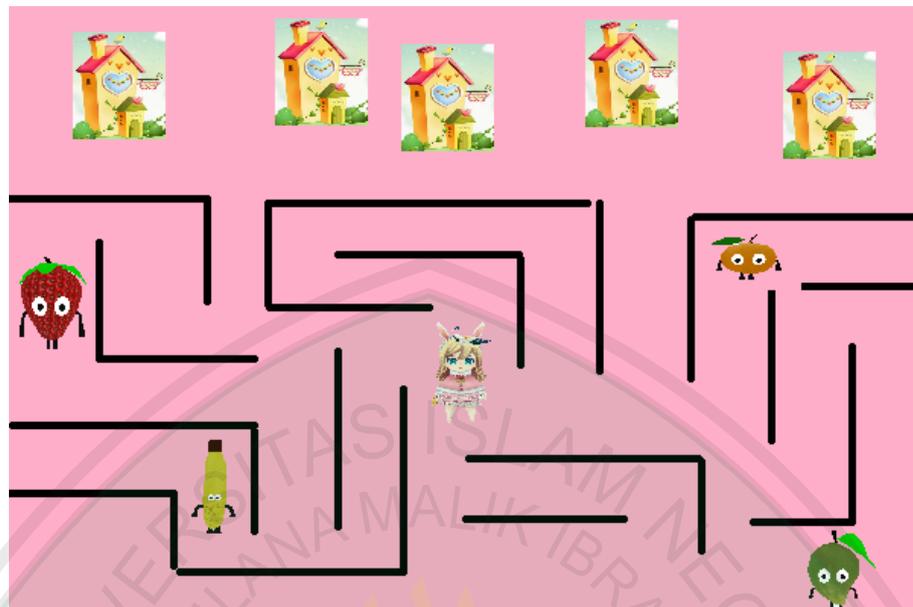
3.3. Objek Penelitian

Objek penelitian yang diambil pada *game* ini adalah implementasi Algoritma A* pada NPC musuh untuk pencarian rute terpendek.

3.4. Story Board

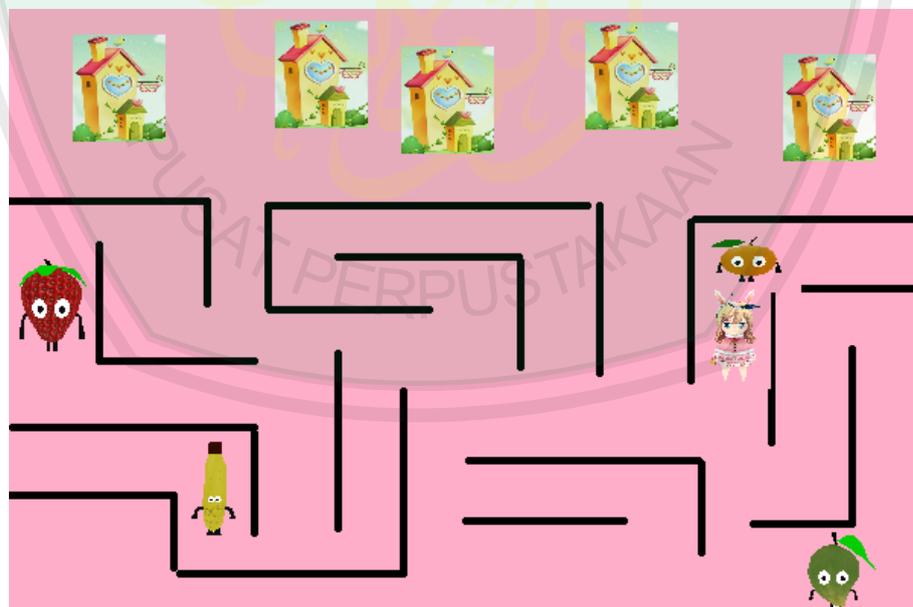
Dalam *game* “Jelajah Kampoeng Arab” ini adalah sebuah petualangan seorang anak bernama Najwa (*Avatar*) yang ingin membantu penduduk “Kampoeng Arab” yaitu yang berupa buah-buahan dan angka-angka. Dalam *game* ini terdapat 2 level (*chapter*) yang bisa dilewati, level pertama berisi karakter buah-buahan, dan di level dua berisi karakter angka-angka. Najwa (*Avatar*) harus segera mencari karakter-karakter tersebut dan mengantarkan mereka kerumah mereka masing-masing, jika najwa berhasil mengantarkan 1 karakter, maka najwa akan mendapatkan satu bintang. Namun apabila najwa terlambat mendapatkan karakter atau penduduk “Kampoeng Arab” maka karakter tersebut akan di ambil oleh tikus jahat (*Enemy*), dan najwa tidak akan mendapatkan bintang.

Agar bisa menang, najwa harus bisa mendapatkan karakter atau penduduk “kampoeng arab” lebih banyak dari pada tikus (*Enemy*).



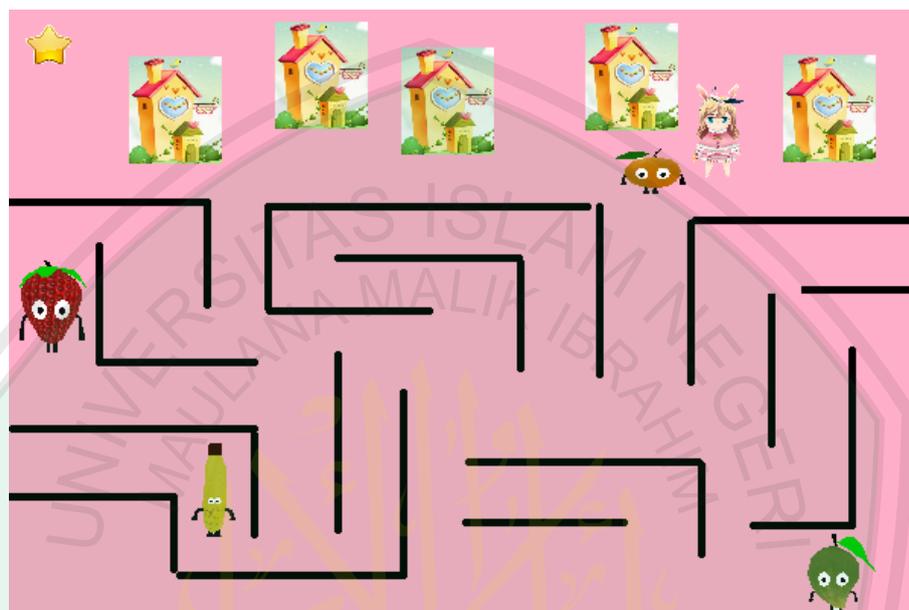
Gambar 3.1 Avatar memasuki labirin

Dari gambar 3.1 menjelaskan bahwa awal dari permainan dimana Avatar akan memasuki sebuah pulau labirin, dan disana terdapat penduduk dari pulau tersebut



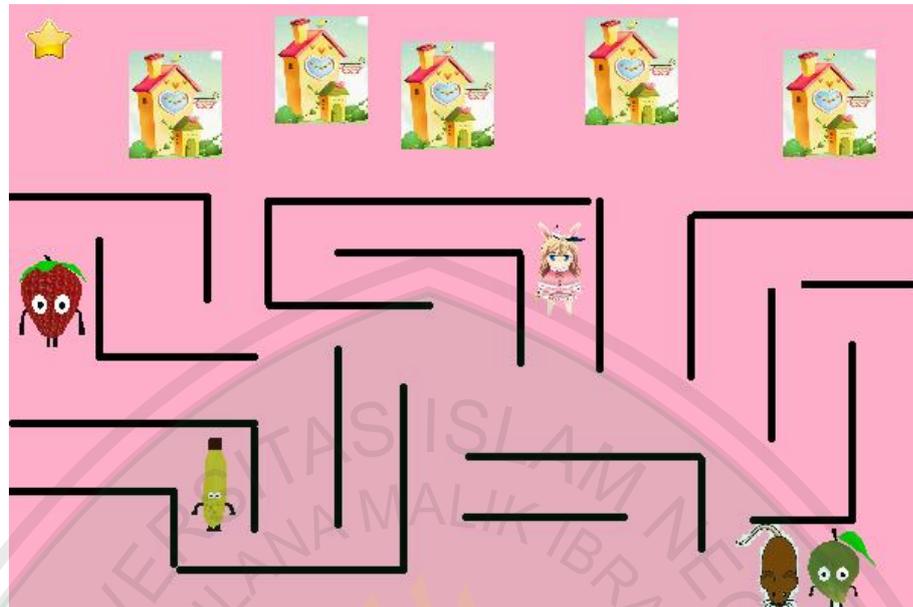
Gambar 3.2 Avatar mencari NPC penduduk

Pada gambar 3.2 menjelaskan bahwa Avatar harus berkeliling pulau untuk bertemu dengan penduduk dan mengantarkanya kembali ke rumah mereka masing-masing



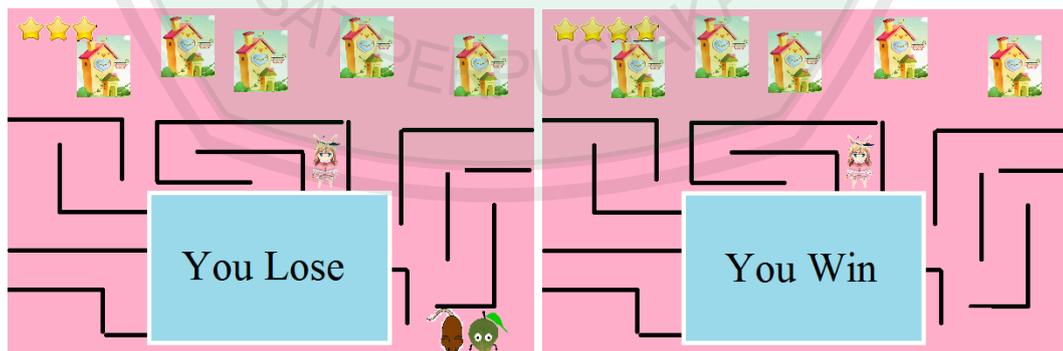
Gambar 3.3 Avatar mengantar pulang NPC penduduk

Setelah *Avatar* berkeliling pulau dan bertemu dengan NPC penduduk maka *Avatar* harus mengantarkan NPC tersebut ke rumah mereka masing-masing, dan jika berhasil memilih rumah yang benar maka *Avatar* akan mendapatkan 1 bintang dari setiap NPC penduduk



Gambar 3.4 NPC musuh mendekati dan memakan NPC penduduk

Ketika *player* memulai permainan tidak hanya *Avatar* yang mencari NPC penduduk, NPC musuh pun juga akan mencari NPC penduduk, dan ketika NPC musuh sudah bertemu dengan NPC penduduk maka NPC penduduk akan di makan oleh NPC musuh, dengan kata lain maka bintang yang akan diperoleh *avatar* akan berkurang.



Gambar 3.5 *avatar lose*

Gambar 3.6 *avatar win*

Ketika *avatar* kalah cepat dengan NPC musuh dan *avatar* hanya mendapatkan 0 sampai 3 bintang saja, maka *avatar* dinyatakan kalah dalam permainan. Tetapi

apabila *avatar* berhasil memperoleh minimal 4 bintang, maka *avatar* dinyatakan menang.

3.5. Deskripsi masing-masing karakter

1. *avatar*

Pada *avatar* dalam *game* ini yang nantinya akan di jalankan oleh pemain adalah seorang anak kecil yang lucu yang bernama Najwa, ciri-ciri fisik dari *avatar* ini adalah perempuan balita yang menggemaskan. *Avatar* dalam ini memiliki semangat untuk menyelesaikan misi dalam *game* ini yaitu menyusuri setiap bagian pulau yang berupa labirin dan mendapatkan sebanyak-banyaknya karakter untuk dibantu pulang.

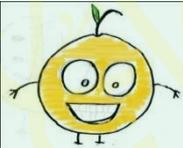
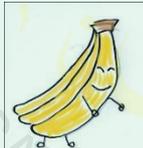
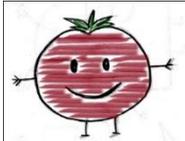


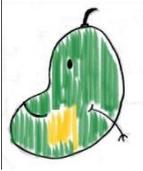
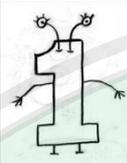
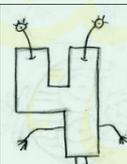
Gambar 3.7 Najwa (*Avatar*)

2. NPC

Non Player characters (NPC) adalah karakter selain yang di perankan oleh pemain. Ada beberapa jenis NPC dalam *game* ini, berikut adalah ulasan dari *avatar* dan karakter NPC

Tabel 3.1 Jenis NPC beserta karakternya

Jenis	Karakter	Nama	Keterangan
NPC Karakter musuh		فارة	Tikus merupakan musuh penduduk dari pulau Kampoeng Arab. Tikus akan mendekati target, dan setelah bertemu dengan target atau penduduk “kampoeng arab” maka tikus akan memakanya.
NPC penduduk kampoeng arab		بُرْتُقَالٌ	Karakter ini merupakan karakter dari penduduk “Kampoeng Arab” yang sedang tersesat, dan menunggu hingga ada bantuan yang mengantarkan pulang.
		مَوْزٌ	
		فَرَاوِلَةٌ	
		طَمَاطِمٌ	

	مَنْجَةٌ	
	وَاحِدٌ	
	إِثْنَانِ	
	ثَلَاثٌ	
	أَرْبَعٌ	
	خَمْسٌ	

3.6. Deskripsi *GamePlay* game

Game ini merupakan *game singlePlayer* yang bergenre *game adventure* atau *game* petualangan. Ketika memulai permainan, pemain akan di bawah di sebuah pulau Kampong Arab di mana pulau itu merupakan pulau yang terdiri dari

beberapa karakter penduduk pulau kampoeng Arab yang harus di temukan oleh *avatar*.

Tugas pemain selama dalam misi adalah membantu memulangkan penduduk atau karakter-karakter yang ada di dalam pulau tersebut agar pemain mendapatkan bintang dari masing-masing karakter.

Selain membantu memulangkan karakter-karakter kembali kerumahnya, pemain akan di hadapkan oleh musuh yang akan memakan karakter-karakter di pulau tersebut.

3.7. *Scoring*

untuk *scoring* dalam *game* ini adalah berupa berapa banyak bintang yang di dapatkan oleh pemain.

- Bintang 1 = *Lose* / kalah
- Bintang 2 = *lose* / kalah
- Bintang 3 = *lose* / kalah
- Bintang 4 = *win* / menang
- Bintang 5 = *win* / menang

3.8. *Pembelajaran dalam game*

Pembelajaran yang di tanamkan dalam *game* ini adalah berupa karakter dan juga rumah masing-masing karakter, di depan rumah masing-masing karakter terdapat nama dari karakter tersebut dalam bahasa arab, sehingga pemain dapat memulangkan karakter ke rumah yang benar. Pemain harus mengerti bahasa arab

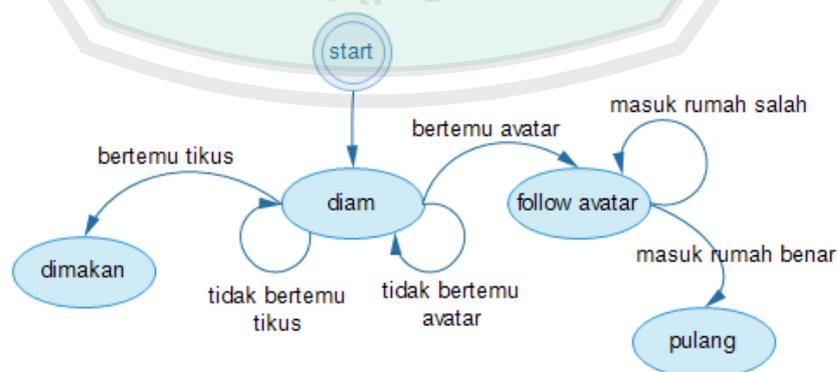
dari masing-masing karakter, secara tidak langsung pemain akan di berikan edukasi berupa kosakata (mufrodat) dasar-dasar kosakata bahasa arab.

Skenario pembelajaran yang di tanamkan yaitu, sebelum permainan di mulai pemain harus menghafalkan kosakata yang di berikan, lalu menempatkan karakter kembali ke rumah yang tepat dengan bekal kosakata yang telah di hafalkan. Jika misi ini bisa di selesaikan maka pemain akan mendapatkan bintang dari masing-masing karakter.

3.9. *Finite State Machine*

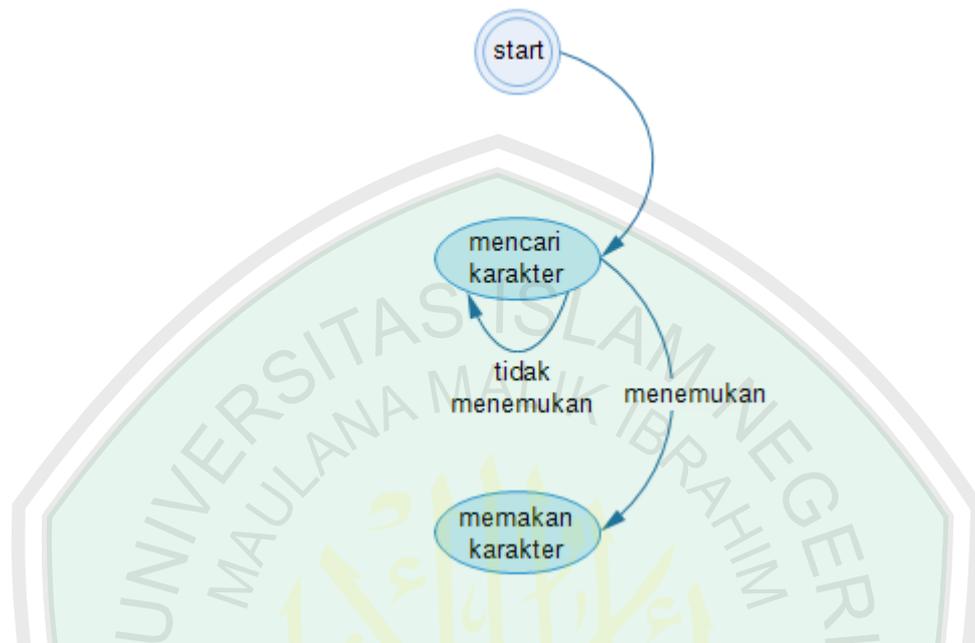
Menurut Iwan Setiawan (2006 : 1), *Finite State Machines* (FSM) adalah sebuah metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan tiga hal berikut: *State* (keadaan), *Event* (kejadian), *Action* (aksi). FSM cukup banyak dipakai sebagai basis perancangan aplikasi yang mempunyai kontinuitas seperti *Game*.

- FSM pada penduduk kampoeng arab



Gambar 3.8 FSM penduduk “kampoeng arab”

- FSM pada tikus atau Musuh



Gambar 3.9 FSM NPC musuh (Tikus)

3.10. Perancangan A* (AStar)

Dalam *game* ini A* (AStar) digunakan menentukan kondisi perilaku yang dilakukan oleh *Non Player Character* (NPC). Dengan adanya A* yang diterapkan pada NPC musuh dapat menentukan perilaku musuh pada *game* ini. Dan dalam *Game* ini Metode A* di terapkan pada tikus atau *enemy*, sehingga tikus akan mencari karakter penduduk pulau Kampoenng Arab dan mencari jalan terpendek untuk menuju karakter tersebut.

3.10.1. Simulasi A* (AStar)

Berikut adalah simulasi dari perhitungan manual pencarian rute terpendek menggunakan A* (AStar).

$$F = G + H$$

Dimana :

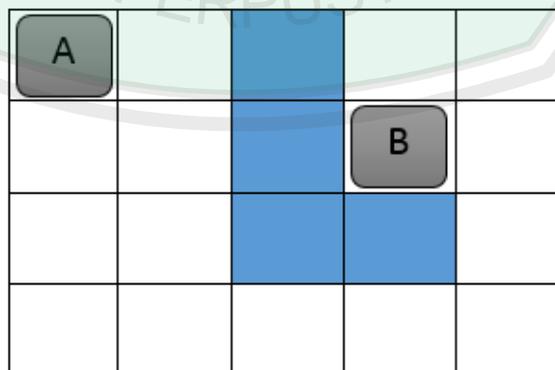
F = harga untuk simpul n

G = harga untuk mencapai simpul n dari akar. G digunakan nilai 10 untuk gerakan vertikal atau horizontal. Angka 10 digunakan untuk memudahkan perhitungan.

Sedangkan pada gerakan diagonal digunakan perhitungan $10 \times \sqrt{2} = 14,14$. Demi memudahkan perhitungan angka dibulatkan menjadi 14.

H = jumlah nilai perkiraan dari sebuah simpul ke simpul tujuan. Untuk perhitungan nilai H digunakan fungsi *heuristic*, metode yang digunakan di dalam contoh ini adalah metode *Manhattan* dimana perhitungan jumlah node hanya yang bergerak secara vertical dan horizontal menuju tujuannya serta mengabaikan penghalang, yang kemudian nilainya dikalikan dengan 10. Atau dirumuskan dengan:

$$H = 10 * (\text{abs}(\text{currentX} - \text{targetX}) + \text{abs}(\text{currentY} - \text{targetY}))$$



Gambar 3.10 Tahap awal pencarian algoritma A*

Pada gambar 3.6 pencarian wilayah dibagi ke dalam *node-node* yang terdiri dari *node* berwarna hijau merupakan *node* awal (*node A*), *node* berwarna merah merupakan *node* akhir (*node B*) dan yang berwarna biru merupakan penghalang (*unwalkable node*).

Selanjutnya adalah, melakukan pencarian untuk menemukan jalan terpendek dengan cara memeriksa *node-node* yang berdekatan dengan *node* awal untuk bisa sampai *node B*. berikut adalah langkah pencarian yang dilakukan oleh *node* awal:

1. Langkah pertama, dimulai dari *node* awal yang ditambahkan ke dalam *Open List* yang merupakan *list* untuk *node-node* yang harus diperiksa.
2. Periksa semua *node* yang berdekatan dengan *node* awal apakah bisa dilalui atau tidak, *node* penghalang yang berwarna biru diabaikan lalu tambahkan semua *node* tersebut ke dalam *open list*, untuk setiap *node* yang baru ditambahkan, simpan *node* awal sebagai "*parent*" yang nantinya akan digunakan untuk menelusuri jalan.
3. Hapus *node* awal dari *open list*, dan tambahkan ke dalam *closed list*.

Closed list merupakan *list* untuk *node* yang perlu diperiksa atau dalam artian merupakan bagian dari jalur terpendek yang sudah didapatkan.

Untuk lebih jelasnya *open list* dan *closed list* diilustrasikan pada gambar 3.7

0 1 8 NPC	8 1 2 6	3	4	5
2 6 8	6 3 7	8	9 Player	10
11	12	13	14	15
16	17	18	19	20

Gambar 3.12 perhitungan menentukan nilai F

Gambar 3.8 menunjukan pada *open list* dari *node* awal yaitu NPC nilai F yang dihasilkan adalah 4 yang didapatkan dari menambahkan nilai $G = 2$ dan nilai $H = 2$. Di dalam contoh ini G akan diberi nilai 2 apabila *node* bergerak secara *vertical* dan *horizontal* dan diberi nilai 3 apabila *node* bergerak secara *diagonal*.

Perhitungan nilai H yaitu nilai estimasi jalur terpendek dari *node* awal ke *node* tujuan atau target dirumuskan dengan:

$$H = 2 * (\text{abs}(n.x - \text{tujuan}.x) + \text{abs}(n.y - \text{tujuan}.y))$$

Sebagai contoh *node* yang terletak disebelah kanan dari *node* awal diberikan nilai $H = 6$ yang didapatkan dari perhitungan 6 *node* menuju *node* tujuan dan dikalikan 2, $2 * 3 = 6$

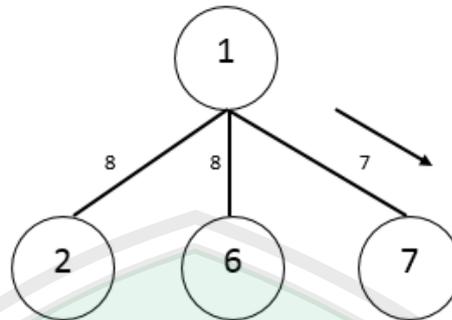
Setelah didapatkan nilai F dari setiap *node* yang ada di *open list*, maka akan dipilih *node* yang memiliki nilai F paling rendah yang kemudian disebut dengan *current node*, langkah selanjutnya adalah sebagai berikut.

1. Hapus *current node* dari *open list* dan masukan ke dalam *closed list*.
2. Periksa semua *node* yang berdekatan dengan *current node*, abaikan *node* penghalang. Jika tidak ada pada *open list*, tambahkan kedalam *open list*, lalu tambahkan "*parent*" untuk *node* tersebut.
3. Jika *node* sudah ada pada *open list*, lalu bandingkan dengan *node* awal apakah jalur yang di ambil lebih baik dari *node* awal, dengan memeriksa nilai G-nya apakah lebih rendah jika kita menggunakan *node* ini, jika lebih rendah maka hitung ulang nilai F dan G serta merubah arah *pointer*, jika tidak maka jangan lakukan apa-apa.

Table 3.2 table Perhitungan Langkah Pertama

<i>Open List</i>	2, 6, 7
<i>Closed List</i>	1
<i>Parent</i>	1
<i>Current Node</i>	7

Dibawah ini adalah penjelasan langkah pertama algoritma *astar* dengan menggunakan *tree*



Gambar 3.13 Penjelasan A* menggunakan *tree* pada langkah pertama

Proses yang dijelaskan di atas, dilakukan berulang kali atau *looping*, dimulai dari *open list*, dimana *open list* juga berperan sebagai *parent* dan menemukan *node* dengan nilai F yang paling rendah yang selanjutnya akan dijadikan *current*

Proses yang dijelaskan di atas, dilakukan berulang kali atau *looping*, dimulai dari *open list*, dimana *open list* juga berperan sebagai *parent* dan menemukan *node* dengan nilai F yang paling rendah yang selanjutnya akan dijadikan *current node* (1,1), langkah-langkahnya sebagai berikut

1. Hapus *current node* dari *open list* atau *parent* lalu masukkan *current node* tersebut ke dalam *closed list*.
2. Periksa semua *node* yang berdekatan dengan *current node*, abaikan penghalang atau *unwalkable node*. Jika tidak ada pada *open list*, tambahkan ke dalam *open list* dan tambahkan *parent* untuk *node* tersebut

3. Jika *node* sudah ada pada *open list*, periksa *open list* yang mana yang memiliki nilai F terendah dengan rumus $F = G + H$, *node* dengan nilai F terendah akan dijadikan *current node* selanjutnya.

Perhitungan pada langkah kedua yang berada pada *node* dengan koordinat dapat dilihat pada gambar 3.7

0 1 8	5 2 6	3	4	5
8 NPC	11	8	9 Player	10
5 6 6	3 7 4	7 closed list	14	15
6 11 8	5 12 6	13	14	15
14	11	18 6	19	20
8 16 10	7 17 8	8 18 6	19	20
18	15	14		

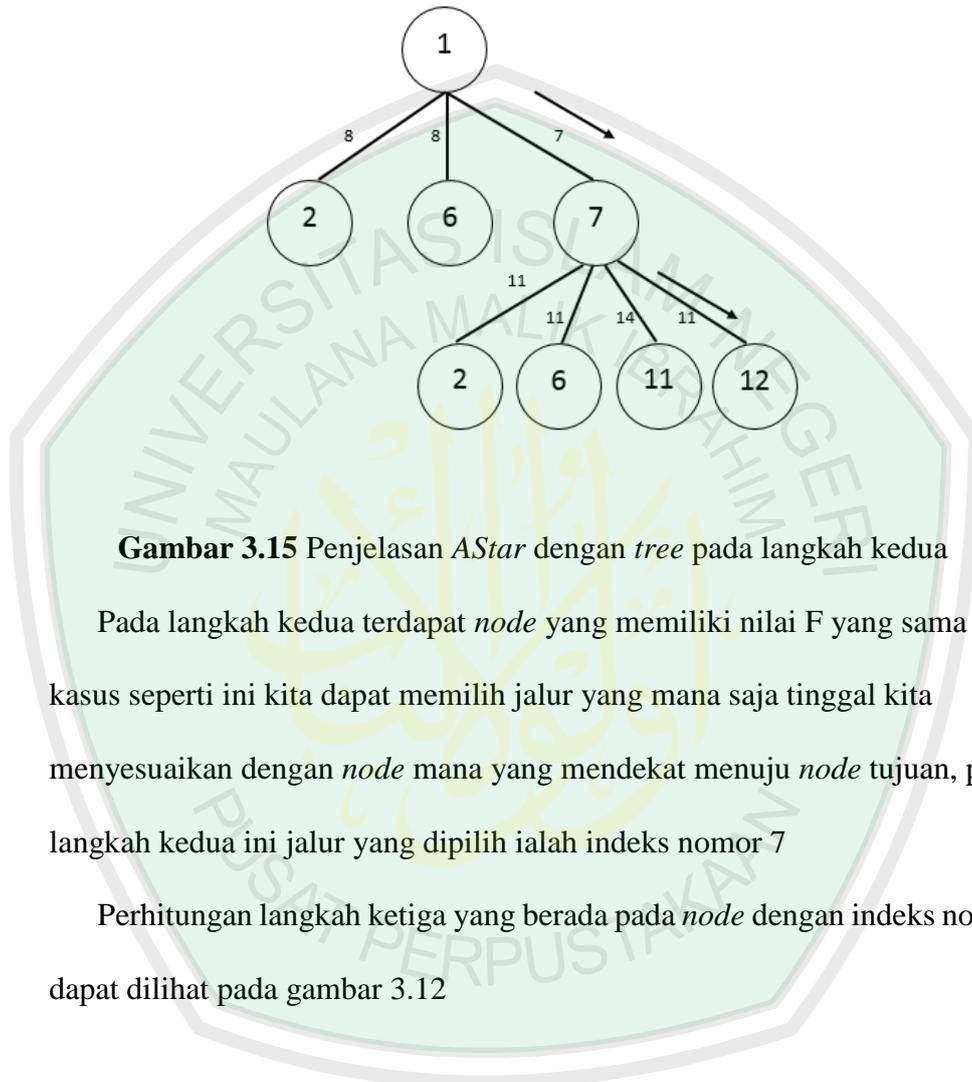
Gambar 3.14 Penjelasan AStar pada langkah kedua

Tabel 3.3 menunjukkan penjelasan tentang pencarian jalur terpendek untuk langkah kedua dapat dilihat pada tabel 3.3

Tabel 3.3 Perhitungan Langkah Kedua

<i>Open List</i>	2, 6, 11, 12
<i>Closed List</i>	1, 3, 7, 8, 13
<i>Parent</i>	7
<i>Current Node</i>	12

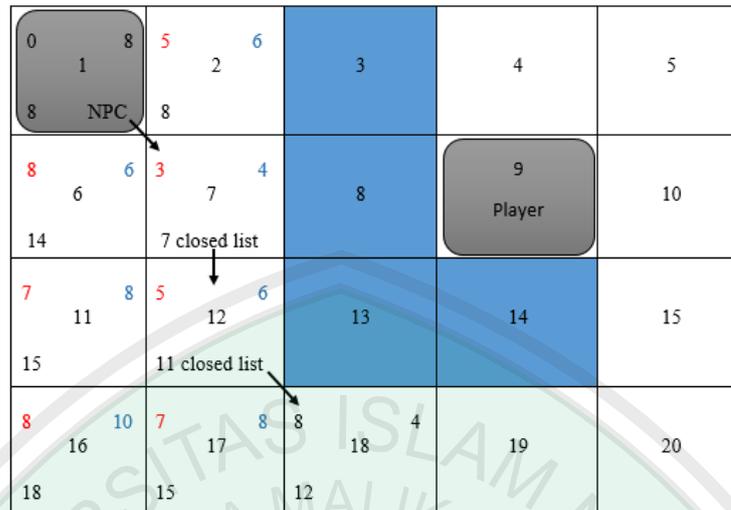
Gambar 3.11 menunjukkan penjelasan melalui *tree* untuk pencarian jalan terpendek pada langkah kedua



Gambar 3.15 Penjelasan AStar dengan *tree* pada langkah kedua

Pada langkah kedua terdapat *node* yang memiliki nilai F yang sama pada kasus seperti ini kita dapat memilih jalur yang mana saja tinggal kita menyesuaikan dengan *node* mana yang mendekati menuju *node* tujuan, pada langkah kedua ini jalur yang dipilih ialah indeks nomor 7

Perhitungan langkah ketiga yang berada pada *node* dengan indeks nomor 12 dapat dilihat pada gambar 3.12



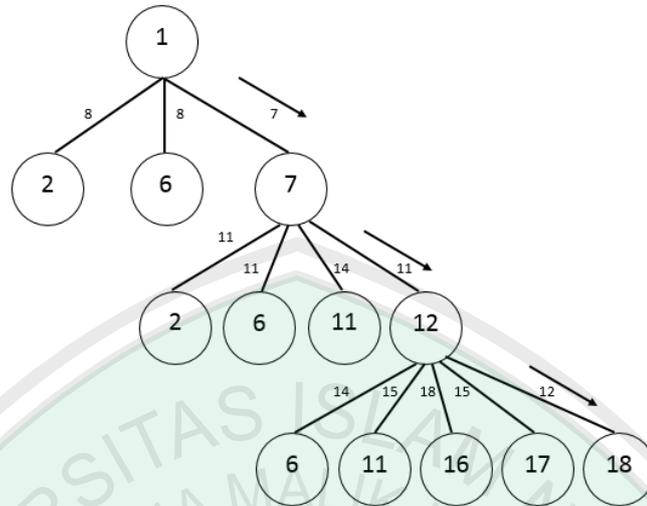
Gambar 3.16 Penjelasan AStar pada langkah ketiga

Tabel 3.4 menunjukkan penjelasan tentang pencarian jalur terpendek untuk langkah kedua dapat dilihat pada tabel 3.4

Tabel 3.4 Tabel Perhitungan langkah ketiga

<i>Open List</i>	6, 11, 16, 17,18
<i>Closed List</i>	12, 13, 8, 7
<i>Parent</i>	12
<i>Current Node</i>	18

Gambar 3.13 menunjukkan penjelasan melalui *tree* untuk pencarian jalan terpendek pada langkah ketiga.



Gambar 3.17 Penjelasan AStar dengan tree pada langkah ketiga
 Perhitungan langkah keempat yang berada pada node dengan indeks 18
 dapat dilihat pada gambar 3.14

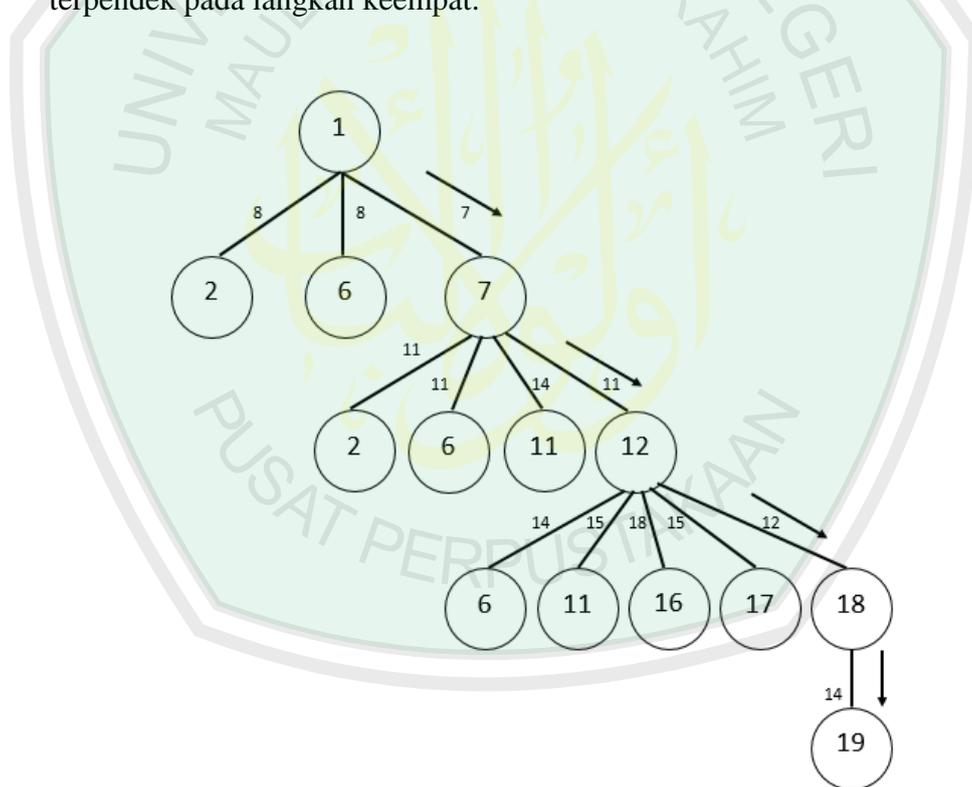
0 1 8	2 2 6	3	4	5
8 NPC	8	8	9 Player	10
2 6 6	3 7 4	8	14	15
8	7	13	14	15
6 11 8	5 12 6	11 closed list	Closed list	Closed list
14	11 closed list	Closed list	Closed list	15
8 16 10	10 17 8	8 18 6	10 19 4	20
18	18	14 closed list	14	20

Gambar 3.18 penjelasan AStar pada langkah keempat
 Tabel 3.5 menunjukkan penjelasan tentang pencarian jalur terpendek untuk
 langkah kedua dapat dilihat pada tabel 3.5

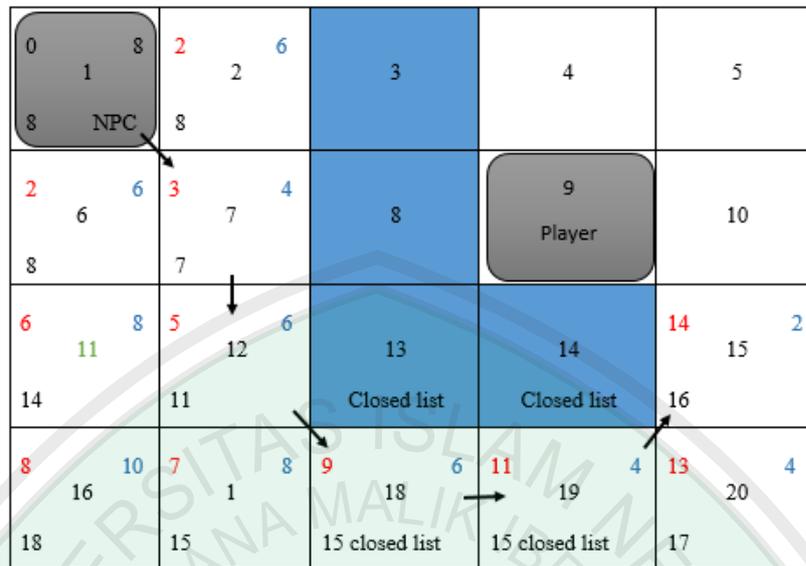
Tabel 3.5 Perhitungan langkah keempat

<i>Open List</i>	17, 19
<i>Closed List</i>	12, 13, 14, 18
<i>Parent</i>	18
<i>Current Node</i>	19

Gambar 3.15 menunjukkan penjelasan melalui *tree* untuk pencarian jalan terpendek pada langkah keempat.

**Gambar 3.19** Penjelasan AStar dengan *tree* pada langkah keempat

Perhitungan pada langkah kelima yang berada pada *node* dengan indeks 15 yang memiliki nilai $F = 16$ bisa dilihat pada gambar 3.16



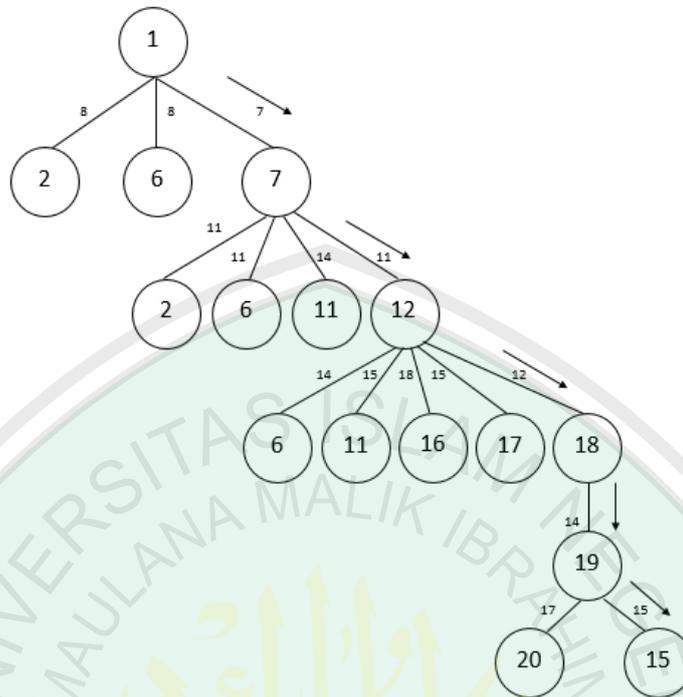
Gambar 3.20 Penjelasan AStar pada langkah kelima

Pada tabel 3.6 penjelasan melalui tabel untuk pencarian jalur terpendek pada langkah keenam.

Tabel 3.6 Tabel perhitungan langkah kelima

<i>Open List</i>	15, 20
<i>Closed List</i>	13, 14, 18
<i>Parent</i>	19
<i>Current Node</i>	15

Gambar 3.17 menunjukkan penjelasan melalui *tree* untuk pencarian jalan terpendek pada langkah kelima.



Gambar 3.21 Penjelasan Astar dengan tree pada langkah kelima

Perhitungan pada langkah keenam yang berada pada node dengan indeks ke 15 yang memiliki nilai F = 16 bisa dilihat pada gambar 3.18

0 1 8 NPC	2 2 8	6 3 8	3 8 13	19 4 14	2 2 18	19 5 16	3 3 2
8	6 7	6 7	8	17 17 player	9 0 18	16 10 14	2 2 2
6 11 14	5 12 11	6 12 11	13	14	14 15 16 closed list	15	2
8 16 18	7 17 15	8 17 15	9 18 15	6 11 15 closed list	4 19 17	13 20	4

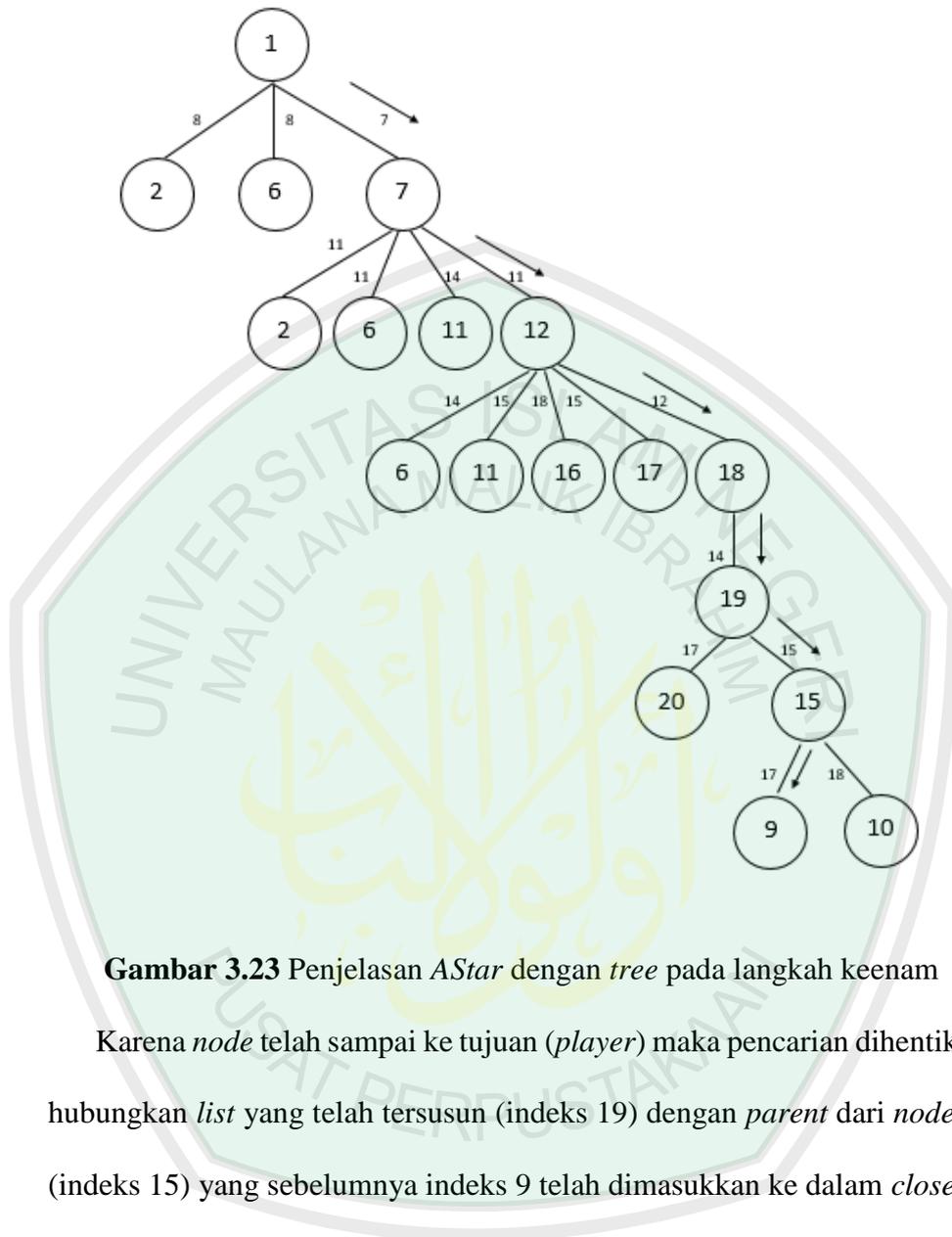
Gambar 3.22 Penjelasan AStar pada langkah keenam

Pada tabel 3.7 penjelasan melalui tabel untuk pencarian jalur terpendek pada langkah keenam.

Tabel 3.7 Perhitungan langkah keenam

<i>Open List</i>	9, 10
<i>Closed List</i>	19, 14
<i>Parent</i>	15
<i>Current Node</i>	9

Gambar 3.19 menunjukkan penjelasan melalui *tree* untuk pencarian jalan terpendek pada langkah keenam.



Gambar 3.23 Penjelasan AStar dengan *tree* pada langkah keenam

Karena *node* telah sampai ke tujuan (*player*) maka pencarian dihentikan lalu hubungkan *list* yang telah tersusun (indeks 19) dengan *parent* dari *node* tujuan (indeks 15) yang sebelumnya indeks 9 telah dimasukkan ke dalam *closed list*.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

Dalam bab ini membahas mengenai implementasi metode terhadap aplikasi yang sudah dibuat dan juga pengujian metode yang di terapkan. Serta melakukan uji coba pada aplikasi yang telah dibangun, apakah telah sesuai dengan perancangan dan hasil yang diharapkan.

4.1.1. Kebutuhan Perangkat Keras

Sebelum diimplementasikan, terlebih dahulu dipaparkan spesifikasi sistem perangkat keras (*hardware*) dan perangkat lunak (*software*). Untuk pembuatan dan melakukan uji coba aplikasi ini diperlukan perangkat keras (*hardware*) dan lunak (*software*), adapun keperluan tersebut adalah:

Tabel 4.1 Kebutuhan Perangkat Keras(1)

No.	Perangkat Keras	Spesifikasi
1.	Processor	Intel(R) Celeron(R) CPU N2840 @ 216Hz 2.16 Ghz
2.	RAM	2 GB
3.	HDD	500 GB

4.	Monitor	14'
5.	Speaker	On
6.	Mouse & Keyboard	On

Tabel 4.2 Kebutuhan Perangkat Keras(2)

No.	Perangkat Keras	Spesifikasi
1.	Processor	Intel atom z2580
2.	RAM	2 GB
3.	System Operasi	Android version 4.4.2 (kitkat)

4.1.2. Kebutuhan Perangkat Lunak

Perangkat lunak (*Software*) yang digunakan untuk pembuatan dan juga mendesain aplikasi ini yaitu:

Tabel 4.3 Kebutuhan Perangkat Lunak

No.	Perangkat Lunak	Spesifikasi
1.	Sistem Operasi	Windows 8.1 Pro
2.	<i>Game Engine</i>	Unity 3D 4.6.1

3.	Desain 3D	Blender 2.7
4.	<i>Script Writer</i>	Mono Develop

4.2. Implementasi Algoritma A* pada Perilaku NPC

Proses implementasi adalah proses penerapan komponen system utama yang dibangun berdasarkan rancangan yang telah dibuat dan diajukan sebelumnya. Implementasi kecerdasan buatan pada penelitian ini diterapkan pada perilaku pencarian pada NPC dengan memanfaatkan metode A*. Metode A* digunakan sebagai pembangkit perilaku pencarian NPC. Posisi NPC dijadikan sebagai *starting point* dan *karakter penduduk* diinisiasi sebagai tujuan.

4.2.1. A* Sebagai Metode Perilaku Pencarian

pada bagian ini membahas tentang penerapan metode pencarian A* pada NPC. Algoritma A* pada NPC musuh bekerja pada saat *game* pertama dimulai. NPC memiliki kecerdasan untuk menemukan posisi pemain. Ketika NPC sampai di posisi pemain maka NPC musuh otomatis memakan karakter penduduk. Pemain bertugas membantu penduduk dari serangan NPC musuh. Algoritma A* dalam *game* ini akan diimplementasikan menggunakan bahasa C# dengan *source code* sebagai berikut :

Tabel 4.4 Keterangan Metode A Star

No.	Method / Fungsi	Keterangan
1.	<pre>private void SetStartAndEndNode(Vector3 start, Vector3 end) { startNode = FindClosestNode(start); endNode = FindClosestNode(end); } private Node FindClosestNode(Vector3 pos) { int x = (MapStartPosition.x < 0F) ? Mathf.FloorToInt(((pos.x + Mathf.Abs(MapStartPosition.x)) / TileSize)) : Mathf.FloorToInt((pos.x - MapStartPosition.x) / TileSize); int z = (MapStartPosition.y < 0F) ? Mathf.FloorToInt(((pos.z + Mathf.Abs(MapStartPosition.y)) / TileSize)) : Mathf.FloorToInt((pos.z - MapStartPosition.y) / TileSize); if (x < 0 z < 0 x > Map.GetLength(0) z > Map.GetLength(1)) return null; private void FindEndNode(Vector3 pos) { int x = (MapStartPosition.x < 0F) ? Mathf.FloorToInt(((pos.x + Mathf.Abs(MapStartPosition.x)) / TileSize)) : Mathf.FloorToInt((pos.x - MapStartPosition.x) / TileSize); int z = (MapStartPosition.y < 0F) ? Mathf.FloorToInt(((pos.z + Mathf.Abs(MapStartPosition.y)) / TileSize)) : Mathf.FloorToInt((pos.z - MapStartPosition.y) / TileSize); Node closestNode = Map[x, z]; List<Node> walkableNodes = new List<Node>(); int turns = 1;</pre>	<p><i>Method</i> memberikan <i>Map</i> <i>Start Position</i> dan <i>End</i> <i>Position</i></p>
2.	<pre>void DrawMapLines() { if (DrawMapInEditor == true && Map != null) { for (int i = 0; i < Map.GetLength(1); i++) { for (int j = 0; j < Map.GetLength(0); j++) {</pre>	<p>Memberikan <i>line</i> di dalam <i>map start position</i> dan <i>end</i> <i>position</i></p>

	<pre> if (!Map[j, i].walkable) continue; for (int y = i - 1; y < i + 2; y++){ for (int x = j - 1; x < j + 2; x++){ if (y < 0 x < 0 y >= Map.GetLength(1) x >= Map.GetLength(0)) continue; if (!Map[x, y].walkable) continue; if (Map[j, i].yCoord > Map[x, y].yCoord && Mathf.Abs(Map[j, i].yCoord - Map[x, y].yCoord) > MaxFalldownHeight) continue; if (Map[j, i].yCoord < Map[x, y].yCoord && Mathf.Abs(Map[x, y].yCoord - Map[j, i].yCoord) > ClimbLimit) continue; Vector3 start = new Vector3(Map[j, i].xCoord, Map[j, i].yCoord + 0.1f, Map[j, i].zCoord); Vector3 end = new Vector3(Map[x, y].xCoord, Map[x, y].yCoord + 0.1f, Map[x, y].zCoord); UnityEngine.Debug.DrawLine(start, end, Color.green); </pre>	
3.	<pre> private WaypointListNode FindClosestNode(Vector3 pos){ Stopwatch a = new Stopwatch(); a.Start(); int ID = -1; float lowestDist = Mathf.Infinity; foreach (WaypointNode m in Map){ float d = Vector3.Distance(m.position, pos); if (d < lowestDist){ ID = m.ID; lowestDist = d; }} if (ID > -1){ WaypointListNode wp = new WaypointListNode(Map[ID].position, Map[ID].ID, null, Map[ID].neighbors); return wp; } else{ return null;}} </pre>	Cari <i>start</i> dan <i>End node</i>
4.	<pre> openList[startNode.ID] = startNode; BHInsertNode(new NodeSearch(startNode.ID, startNode.F)); </pre>	sisipkan <i>start node</i>

	<pre>bool endLoop = false;</pre>	
5.	<pre>int id = BHGetLowest(); closedList[currentNode.ID] = currentNode; openList[id] = null; if (currentNode.ID == endNode.ID){ endLoop = true; continue;</pre>	<p>jika sudah mendapatkan simpul terkecil maka masukan kedalam <i>closed list</i></p>
6.	<pre>NeighbourCheck();} while (true){ returnPath.Add(currentNode.position); if (currentNode.parent != null){ currentNode = currentNode.parent;} else{ break;}} returnPath.Reverse();</pre>	<p>Cek <i>node</i> tetangga</p>
7.	<pre>WaypointListNode n = GetNodeFromOpenList(wp.ID); if (currentNode.G + GetMovementCost(currentNode.position, wp.position) < openList[wp.ID].G){ n.parent = currentNode; n.G = currentNode.G + GetMovementCost(currentNode.position, wp.position); n.F = n.G + n.H; BHSortNode(n.ID, n.F);}</pre>	<p>Memeriksa ketika <i>new path</i> lebih kecil</p>
8.	<pre>private bool OnOpenList(int id){ return (openList[id] != null) ? true : false; }</pre>	<p>Periksa apakah <i>node</i> sudah berada di <i>openList</i></p>
9.	<pre>private bool OnClosedList(int id){ return (closedList[id] != null) ? true : false;} private float GetHeuristics(Vector3 p1, Vector3 p2){ return Vector3.Distance(p1, p2);} private float GetMovementCost(Vector3 p1, Vector3 p2){ return Vector3.Distance(p1, p2);} private WaypointListNode GetNodeFromOpenList(int id){ return (openList[id] != null) ? openList[id] : null;}</pre>	<p>Periksa apakah <i>node</i> sudah berada di <i>closedList</i></p>

4.3. Implementasi Aplikasi *Game*

Pada bagian ini akan dipaparkan tentang implementasi dari rancangan “*Jelajah Kampoeng Arab*” ke dalam aplikasi perangkat *mobile* berbasis *android* dan penjelasan tiap-tiap *scene*. Berikut adalah tampilan *game* yang telah selesai dibuat.

4.3.1. Tampilan *Splashscrene*

Tampilan pembuka halaman akan tampil pada saat pertama kali *game* di jalankan sebelum ke tampilan menu utama. Pada tampilan ini akan tampil beberapa 3 detik, kemudian akan berganti pada halaman menu utama.



Gambar 4.1 Tampilan *Splashscrene*

4.3.2. Tampilan Menu *Game*

Bagian ini menampilkan menu-menu pilihan. Pada tampilan menu terdapat 4 menu yaitu *Play*, *Help*, *About* dan *Quit* yang memiliki fungsi masing – masing yaitu.

a. *Play*

Berfungsi untuk memulai *game* pada saat pertama kali memainkan *game*.

b. *Help*

Berfungsi untuk menampilkan aturan atau cara bermain.

c. *About*

Berfungsi untuk menampilkan informasi tentang *game*.

d. *Quit*

Berfungsi untuk keluar dari permainan.



Gambar 4.2 Tampilan Menu

4.3.3. Tampilan Pilihan Tema Permainan

Bagian ini yaitu tampilan yang akan muncul setelah kita memilih menu *Play*. Pilihan ini berfungsi untuk memilih tema permainan. Tema pada *game* ini terdiri dari 2 kategori, yaitu buah-buahan (الفواكه) dan angka-angka (الاعداد). Untuk kembali ke menu utama yaitu menekan tombol *Back*.



Gambar 4.3 tampilan Tema Pilihan permainan

4.3.4. Tampilan *Scene Help*

Bagian ini merupakan pengenalan karakterter dari “ penduduk kampoeng Arab” dan juga menjelaskan misi-misi yang harus di lakukan oleh *Player* agar dapat membantu karakter “penduduk kampoeng Arab”. Untuk kembali ke menu utama yaitu menekan tombol *Back*.



Gambar 4.4 Tampilan Misi Game

4.3.5. Tampilan Scene About

Scene ini menampilkan informasi tentang *game* ini. Informasi yang ditampilkan yaitu identitas pembuat dan *game* dan informasi *credit* properti yang dipakai dalam *game*. Untuk kembali ke menu utama yaitu menekan tombol *Back*.



Gambar 4.5 Tampilan Menu About

4.3.6. Tampilan Permainan

Setelah memilih menu *Play*, maka pemain akan memasuki arena permainan yang telah di pilih seperti pada gambar 4.4.



Gambar 4.6 Mulai permainan

Pemain harus mencari penduduk kampoeng arab yang tersesat disetiap sudut jalan atau labirin. Jika sudah bertemu dengan penduduk kampoeng arab maka penduduk kampoeng arab akan mengikuti *player* seperti gambar 4.5



Gambar 4.7 penduduk kampoeng arab mengikuti *player*

Setelah *player* bertemu dengan salah satu penduduk kampoeng arab maka *player* harus segera mencari rumah dari salah satu penduduk tersebut. Untuk mengetahui dimana rumah dari penduduk tersebut maka *player* terlebih dahulu harus mengetahui apa bahasa arab dari salah satu penduduk kampoeng arab tersebut. Jika *player* tidak mengetahui dan memasuki rumah yang salah maka penduduk kampoeng arab tidak akan memasuki rumah tersebut seperti pada gambar 4.6



Gambar 4.8 *Player* memilih rumah penduduk yang salah

Akan tetapi jika *player* mengetahui bahasa arab dari karakter penduduk kampoeng arab dan dapat memilih rumah penduduk yang benar maka penduduk kampoeng arab tersebut akan memasuki rumahnya dan *player* akan mendapatkan satu bintang yang tertera di pojok atas sebelah kiri layar, seperti gambar 4.7



Gambar 4.9 *Player* memilih rumah yang benar dan mendapatkan bintang.

player akan mendapatkan satu bintang dari setiap penduduk yang telah berhasil *player* pulangkan ke dalam rumah yang benar.

Akan tetapi *player* harus bergerak cepat sebelum karakter penduduk di hampiri oleh tikus, seperti pada gambar 4.8



Gambar 4.10 tikus menghampiri NPC penduduk

Setelah tikus menghampiri dan mendapatkan karakter penduduk maka tikus tersebut akan memakan karakter penduduk tersebut, seperti gambar 7.9

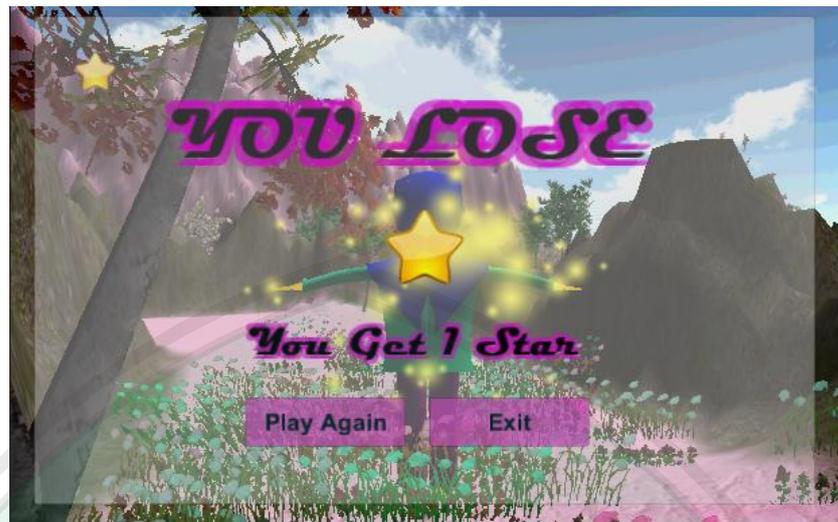


Gambar 4.11 Tikus memakan NPC Penduduk

Untuk itu, agar *player* dapat memenangkan permainan maka *player* harus berhasil lebih cepat dari pada tikus. Semakin sedikit karakter yang dapat di antarkan pulang oleh *player*, maka semakin sedikit pula bintang yang di dapatkan oleh *player*, dan kemenangan *player* tergantung dari bintang yang di dapatkan, seperti gambar berikut



Gambar 4.12 Tampilan *Lose*, tidak mendapatkan bintang



Gambar 4.13 Tampilan *Lose*, mendapatkan 1 bintang



Gambar 4.14 Tampilan *Lose*, mendapatkan 2 bintang



Gambar 4.15 Tampilan *Lose*, mendapatkan 3 bintang



Gambar 4.16 Tampilan *Win*, mendapatkan 4 bintang



Gambar 4.17 Tampilan *Win*, mendapatkan 5 bintang

4.4. Uji coba

Sebelum diimplementasikan pada *game* yang akan dibuat, algoritma *A-Star* akan di uji coba terlebih dahulu. Pada dasarnya algoritma *A-Star* adalah algoritma *pathfinding* dengan cara melakukan pencarian nilai heuristik terkecil dari setiap *node* yang ada disekitar.

Pada subbab ini membahas tentang uji coba yang telah dilakukan. Terdapat dua uji coba yakni uji coba imlementasi algoritma *A Star* dan uji coba pada *game*. Berikut pembahasan uji coba tersebut.

4.4.1. Uji Coba Algoritma

Uji coba A* dilakukan untuk mengetahui harga F , G dan H serta kinerja algoritma dalam menemukan rute. Proses ini melakukan uji coba dengan mengambil nilai koordinat musuh dan pemain.

Posisi Target:

X: 1347; Y: 3; Z: 782

Posisi NPC:

X: 1752; Y: 6; X: 1103

Tabel 4.5 Hasil Uji Coba Algoritma A Star

	Nilai G minimum	Nilai H minimum	Nilai F minimum	Current X, Current Y
1	2057.606619351717	-702	1355. 606619351717	(1752, 1079)
2	2043.577255696491	-675	1368. 577255696491	(1752, 1052)
3	2029.81008963893	-648	1381. 81008963893	(1752, 1025)
4	2016.310490962981	-621	1395. 310490962981	(1752, 998)
5	2003.083872432705	-594	1409. 083872432705	(1752, 971)
6	1821.676425713414	-567	1254. 676425713414	(1752, 944)

7	1977.471365152982	-540	1437.471365152982	(1752, 917)
8	1917.108238989129	-459	1458.108238989129	(1698, 890)
9	1893.235590200015	-432	1461.235590200015	(1671, 890)
10	1869.448046884427	-405	1464.448046884427	(1644, 890)
11	1845.748899498521	-378	1467.748899498521	(1617, 890)
12	1822.141597132341	-351	1471.141597132341	(1590, 890)
13	1798.629756231115	-324	1474.629756231115	(1563, 890)
14	1775.217169813316	-297	1478.217169813316	(1536, 890)
15	1765.777449170761	-297	1468.777449170761	(1509, 917)
16	1757.116956835828	-297	1460.116956835828	(1482, 944)
17	1719.857552240883	-243	1476.857552240883	(1455, 917)
18	1697.077782542686	-216	1481.077782542686	(1428, 917)
19	1674.42228843264	-891	1485.42228843264	(1401, 917)
20	1637.063224191418	-135	1502.063224191418	(1374, 890)
21	1599.74310437645	-81	1518.74310437645	(1347, 863)
22	1585.340657398277	-54	1531.340657398277	(1347, 836)

23	1571.270186823387	-27	1544. 270186823387	(1347, 809)
24	1557.540689677159	0	1557. 540689677159	(1347, 782)

4.4.2 Uji Coba Game

Uji coba ini dilakukan untuk mengetahui apakah *game* yang telah dibuat dapat diimplementasikan di PC. Berikut hasil pengujian yang disajikan dalam tabel

Tabel 4.6 Uji Coba Game

No.	Versi Android	Sudah Diuji	Belum Diuji	Keterangan
1.	Android 2.3-2.3.2 Gingerbread (API level 9)	√		Game tidak berjalan
2.	Android 2.3.3-2.3.7 Gingerbread (API level 10)	√		Game tidak berjalan
3.	Android 4.0-4.0.2 Ice Cream Sandwich (API level 14)	√		Tampilan menu berjalan namun <i>game</i> tidak bisa dimainkan

4.	Android 4.0.3-4.0.4 Ice Cream Sandwich (API level 15)	√		Tampilan menu berjalan namun <i>game</i> berjalan lambat
5.	Android 4.1 Jelly Bean (API level 16)	√		<i>Game</i> berjalan dengan baik
6.	Android 4.2 Jelly Bean (API level 17)	√		<i>Game</i> berjalan dengan baik
7.	Android 4.3 Jelly Bean (API level 18)	√		<i>Game</i> berjalan dengan baik
8.	Android 4.4 Kitkat (API level 19)	√		<i>Game</i> berjalan dengan baik
9.	Android 5.0 Lollipop	√		<i>Game</i> berjalan dengan baik

4.5. Game Pembelajaran Kosakata Bahasa Arab Dalam Pandangan Islam

Sebagai umat Islam, Al-qur'an adalah pedoman utama yang didalamnya berisi petunjuk bagi umatnya. Akan tetapi Al-qur'an ditulis dengan menggunakan bahasa Arab. Oleh karena itu, sebagai umat Islam harus mempelajari bahasa arab supaya bisa memahami apa yang terkandung didalamnya. Dalam surat Yusuf ayat 2 disebutkan :

إِنَّا أَنْزَلْنَاهُ قُرْآنًا عَرَبِيًّا لَعَلَّكُمْ تَعْقِلُونَ

“Sesungguhnya Kami menurunkan berupa Al-Qur'an dengan bahasa Arab, agar kamu memahaminya.”(yusuf : 2).

Ibnu Katsir berkata ketika menafsirkan Surat Yusuf ayat 2 di atas:

“Yang demikian itu (bahwa Al-Qur'an diturunkan dalam bahasa Arab) karena bahasa Arab adalah bahasa yang paling fasih, jelas, luas, dan maknanya lebih mengena lagi cocok untuk jiwa manusia. Oleh karena itu kitab yang paling mulia (yaitu Al-Qur'an) diturunkan kepada Rasul yang paling mulia (yaitu: Rosulullah), dengan bahasa yang termulia (yaitu bahasa Arab), melalui perantara malaikat yang paling mulia (yaitu malaikat Jibril), ditambah dengan kitab inipun diturunkan pada dataran yang paling mulia diatas muka bumi (yaitu tanah Arab), serta awal turunnya pun pada bulan yang paling mulia (yaitu romadhan), sehingga Al-Qur'an menjadi sempurna dari segala sisi.” (Tafsir Ibnu Katsir, Tafsir Surat Yusuf).

Ayat ini menyinggung dua poin penting : pertama, Al-qur'an merupakan sebuah kitab pencerah. Kitab yang menunjukkan jalan kebenaran dan pelita jalan kehidupan,

sehingga manusia dapat berjalan di bawah naungan sinarnya untuk sampai ke tempat tujuannya. Kedua, semua muslim ditekankan menekuni makna dan kandungan ayat-ayat Al-qur'an, serta menfaatkannya guna mengembangkan akal dan pikirannya. Al-qur'an tidak diturunkan dengan tujuan sekedar supaya manusia membacanya dan memperoleh pahala akhirat akan tetapi Al-qur'an diturunkan agar manusia menegakkan kehidupan pribadi dan sosialnya berdasarkan ajaran-ajarannya, dan menjadikan petunjuk Al-qur'an sebagai pelita hidupnya (Effendy, Fuad. 2005).

Dari poin penting yang sudah dijelaskan di atas, bisa dipahami seberapa besar peran Al-qur'an dalam kehidupan manusia. Oleh karena itu diharuskan bagi semua muslim mempelajarinya supaya nantinya tidak tersesat. Belajar bahasa arab adalah cara utama untuk mempelajari Al-qur'an, dengan menguasai bahasa arab maka akan mudah untuk memahami makna penting yang terkandung dalam Al-qur'an. Akan tetapi belajar bahasa arab itu tidaklah mudah, banyak hal yang membuat orang berhenti untuk belajar bahasa arab, salah satu faktornya yaitu bosan dengan metode pembelajaran yang diterapkan dan kesulitan dalam menghafal kosakata.

Salah satu belajar efektif adalah dengan cara membuat orang yang sedang belajar menjadi senang dan menyukai apa yang sedang dipelajarinya. *Game* edukasi adalah salah satu contoh yang termasuk dalam pembelajaran yang cukup efektif, karena bisa membuat orang senang bermain dan sekaligus secara tidak langsung mereka belajar.

Dalam tugas akhir ini dibuat sebuah *game* edukasi yang didalamnya menggunakan konten Islam, yaitu mempelajari beberapa kosakata (Mufrodat) dasar dalam bahasa arab. Dari *game* edukasi ini semoga nantinya bisa membuat pemain secara tidak langsung belajar menghafal kosakata bahasa arab.



BAB V

PENUTUP

5.1 Kesimpulan

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

Penggunaan Algoritma A* berhasil diterapkan sebagai pembangkit perilaku pencarian pada NPC. Ditunjukkan dengan uji coba pada Tabel 4.6. Dalam proses pencarian, NPC mampu melewati halangan yang ada dan berhasil menemukan keberadaan lokasi NPC penduduk. *Game* “Jelajah kampoeng arab” telah diuji cobakan ke 9 sistem operasi berbeda pada platform android, pada berbagai *device* tersebut *game* ini menunjukkan tingkat keberhasilan yang berbeda pada uji coba sistem dan tampilan.

Dari pengujian yang dilakukan sebanyak 9 kali dan dilakukan di berbagai platform android yang berbeda spesifikasi, dapat diketahui prosentase pengujian sebagai berikut :

Tabel 5.1 Hasil presentase

No	Keterangan	Baik	
		Jumlah	%
1	<i>Game</i> tidak berjalan	2	$(2/9) \times 100 = 22.222\%$
2	Tampilan menu berjalan namun <i>game</i> tidak bisa dimainkan	1	$(1/9) \times 100 = 11.111\%$
3	Tampilan menu berjalan namun <i>game</i> berjalan lambat	1	$(1/9) \times 100 = 11.111\%$
4	<i>Game</i> berjalan dengan baik	5	$(5/9) \times 100 = 55.555\%$

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 khususnya anak-anak untuk mengasah pengetahuan pelajaran, oleh karena itu penulis menyarankan beberapa hal untuk bahan pengembangan selanjutnya, diantaranya:

1. Mengembangkan *game* ini agar memiliki tampilan yang lebih menarik lagi
2. Menambah karakter penduduk (kosakata) agar lebih memperbanyak bekal kosakata dalam pembelajaran.
3. Menambah rintangan agar lebih tertantang untuk memainkan *game* ini.



DAFTAR PUSTAKA

- Aslami, Joehani Abdillah. 2015. *Implementasi Algoritma A* (Astar) Sebagai Dasar Pergerakan NPC (Non Player Character) Mendekati Player Untuk Meningkatkan Realitas Game Pembelajaran Kosakata Bahasa Arab*. Malang: UIN Maulana Malik Ibrahim Malang.
- Alif ,Ifa. 2015. *3D Wayang Adventure Game Untuk Pengenalan Budaya Wayang Nusantara Menggunakan A* Pathfinding Algorithm Sebagai Pembangkit Perilaku Pencarian Pada NPC*. Malang: UIN Maulana Malik Ibrahim Malang.
- Harianja, Firman. 2013. *Penerapan Algoritma A* Pada Permasalahan Optimalisasi Pencarian Solusi Dynamic Water Jug*. Jurnal Teknik Informatika STMIK Budidarma Medan.
- Setyawa, Harhaendro Bayu, dkk. *Optimasi Rute Perjalanan Ambulance Menggunakan Algoritma A-Star*. Jurnal Teknik Elektro FTI – ITS Surabaya.
- Galih Bonifatius dkk. 2013. *Analisis Implementasi Algoritma A* (AStar) pada game RPG (Role Playing Game) 3D sebagai dasar pergerakan NPC (Non Player Character)mendekati player untuk meningkatkan realitas game word*. Jurnal dari <http://ti.ukdw.ac.id/ojs/index.php/informatika/article/view/314>.
- Wiyanti, Wina dkk. 2013. *Analisis pengaruh penggunaan nilai heuristik terhadap performansi algoritma A* pada game pathfinding*. Jurnal Universitas Jenderal Achmad Yani.
- Ramadhani, Aristama. 2008. *Menggerakan karakter game menggunakan Algoritma Breadth-First Search (BFS) dan algoritma A*(A Star)*. Jurnal Institut Teknologi Bandung.

Tilawah, Hapsari. 2011. *Penerapan Algoritma Astar (A*) untuk menyelesaikan masalah maze*. Jurnal Institut Teknologi Bandung.

<http://www.hermantolle.com/class/docs/unity-3d-game-engine/> diakses tanggal 21 April 2015

Hart, P.E; Nilsson, N.J.; Raphael, B. 1968. *Aformal basis for the heuristic determination of minimum cost path*. IEEE Transactions on system science and cybernetics SSC4 4 (2): 100-107.

Pearl, Judea. 1984. *Heuristic: Intelligent search strategies for computer problem solving*. Addison-Wesley.

Effendy, Fuad. 2005. Metodologi pembelajaran bahasa arab, Misykat: Malang.

Beck. C. John & Wade Mitchell. 2007. *Gamer Juga Bisa Sukses*. Jakarta. Penerbit PT. Grasindo.

John von Neumann and Oskar Morgenstern, 3d ed. 1953. *Theory of Games and Economic Behavior*.

Nilwan Agustinus. 1995, *Pemograman Animasi dan Game Profesional*, Jakarta: Penerbit Elek Media Komputindo.

Dillon, Teresa (2005) : *Adventure Games for Learning and Storytelling*. UK, Futurelab Prototype Context Paper, Adventure Author

Henry, samuel. (2010). *Cerdas dengan game : panduan praktis Bgi orangtua dalam mendampingi anak bermain game*. Jakarta: PT. Gramedia pustaka utama.

Beck, John C dan Mitchell Wade. 2007. *Gamer Juga Bisa Sukses*. Jakarta : PT Grasindo.

Russel, Stuart J and Peter Novig, 2003. New Jersey: Prentice Hall. *Artificial Intelligence*.

Sutanto, Arnold Nugroho.(2009), Penerapan Algoritma A* dalam Pencarian Jalan untuk Permainan

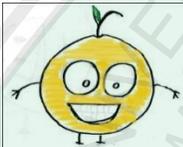
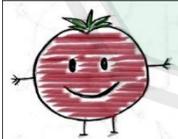
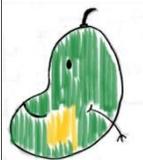
Lose Your Marble, MakalahIF3051-059, Program Studi Teknik Informatika ITB, Bandung.

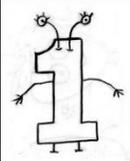
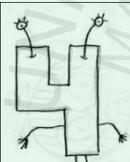
Zou, Huilai., Qu, Zening., Qu, Youtian. *Optimized Application and Practice of A* Algorithm in Game Map Path-Finding*.

Iwan Setiawan, 2006. **Perancangan Software Embedded System Berbasis FSM**,

<http://www.elektro.undip.ac.id/iwan/Perancangan%20Software%20>

[Embedded%20System%20Berbasis%20FSM.pdf](http://www.elektro.undip.ac.id/iwan/Perancangan%20Software%20Embedded%20System%20Berbasis%20FSM.pdf), diakses 11 April 2016.

karakter	Bahasa arab	Bahasa indonesia
	<p>فارة</p>	<p>Tikus</p>
	<p>بُرْتُقَالٌ</p>	<p>Jeruk</p>
	<p>مَوْزٌ</p>	<p>Pisang</p>
	<p>فَرَاوَلَةٌ</p>	<p>Strobery</p>
	<p>طَمَاطِيمٌ</p>	<p>Tomat</p>
	<p>مَنْجَةٌ</p>	<p>Mangga</p>

	<p>وَاحِدٌ</p>	<p>Satu</p>
	<p>اِثْنَانِ</p>	<p>dua</p>
	<p>ثَلَاثٌ</p>	<p>Tiga</p>
	<p>أَرْبَعٌ</p>	<p>Empat</p>
	<p>خَمْسٌ</p>	<p>lima</p>