

**GAME PENGENALAN TAJWID BERBASIS MOBILE
DENGAN METODE ITERATIVE DEEPENING A*
UNTUK PENENTUAN ARAH GERAK MUSUH**

SKRIPSI

oleh:
MUIZ LIDINILLAH
NIM. 10650005



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

**GAME PENGENALAN TAJWID BERBASIS MOBILE
DENGAN METODE ITERATIVE DEEPENING A*
UNTUK PENENTUAN ARAH GERAK MUSUH**

SKRIPSI

oleh:
MUIZ LIDINILLAH
NIM. 10650005



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

**GAME PENGENALAN TAJWID BERBASIS MOBILE
DENGAN METODE ITERATIVE DEEPENING A*
UNTUK PENENTUAN ARAH GERAK MUSUH**

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:
MUIZ LIDINILLAH
NIM. 10650005**

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

LEMBAR PERSETUJUAN

**GAME PENGENALAN TAJWID BERBASIS MOBILE
DENGAN METODE ITERATIVE DEEPENING A*
UNTUK PENENTUAN ARAH GERAK MUSUH**

SKRIPSI

Oleh :

Nama : Muiz Lidinillah

NIM : 10650005

Jurusan : Teknik Informatika

Fakultas : Sains dan Teknologi

Telah diperiksa dan disetujui untuk diuji :

Tanggal :

Dosen Pembimbing I

Dosen Pembimbing II

Hani Nurhayati, M.T

Dr. M. Faisal, M.T

NIP. 19780625 200801 2 006

NIP. 19740510 200501 1 007

Mengetahui,

Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdiان

NIP. 19740424 200901 1 008

LEMBAR PENGESAHAN

**GAME PENGENALAN TAJWID BERBASIS MOBILE
DENGAN METODE ITERATIVE DEEPENING A*
UNTUK PENENTUAN ARAH GERAK MUSUH**

SKRIPSI

Oleh :

Muiz Lidinillah**NIM. 10650005**

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

Tanggal : 10 April 2014

Susunan Dewan Penguji:**Tanda Tangan**

- | | | | |
|---------------------------|---|----------|----------|
| 1. Penguji Utama | : <u>Yunifa Miftachul Arif, MT</u> | (|) |
| | NIP. 19830616 201101 1 004 | | |
| 2. Ketua Penguji | : <u>Fachrul Kurniawan, M.MT</u> | (|) |
| | NIP. 19771020 200901 1 001 | | |
| 3. Sekretaris | : <u>Hani Nurhayati, M.T</u> | (|) |
| | NIP. 19780625 200801 2 006 | | |
| 4. Anggota Penguji | : <u>M. Faisal, M.T</u> | (|) |
| | NIP. 19740510 200501 1 007 | | |

Mengetahui,
Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdiان**NIP. 19740424200901 1 008**

**HALAMAN PERNYATAAN
KEASLIAN PENELITIAN**

Saya yang bertandatangan di bawah ini:

Nama : Muiz Lidinillah

NIM : 10650005

Fakultas/Jurusan : Sains dan Teknologi / Teknik Informatika

Judul Penelitian : Game Pengenalan Tajwid Berbasis Mobile Dengan Metode Iterative Deepening A* untuk Penentuan Arah Gerak Musuh

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

Malang, 08 Januari 2013
Yang membuat pernyataan,

Muiz Lidinillah
NIM. 10650005

MOTTO

“Membaca Al-Quran
dengan tartil dan
memperhatikan ilmu
tajwid akan
menghasilkan suara yang
enak didengar.”

HALAMAN PERSEMBAHAN

Dengan mengucapkan syukur alhamdulillah, kupersembahkan skripsi ini untuk orang-orang yang kusayangi :

Ayah (Abdul Mutholib), Ibu (Lailatul Badriyah) tercinta yang telah memberikan doa, spirit, tenaga, biaya yang tiada henti-hentinya dalam menuntut ilmu. Karya ini merupakan salah satu balas jasanya, meskipun tiada bandingannya dengan kasih sayang dan pengorbanan yang telah engkau berikan. Semoga Allah memberikan segala rahmat-Nya untuk ayah dan ibu.

Adik – adikku (Dek Feriq, Dek Nizam) yang telah memberikan banyak dukungan, motivasi moral dan doa yang tiada hentinya untuk menyelesaikan studi ini. Semoga Allah selalu memberikan kecerdasan, kemudahan, kesehatan dalam menuntut ilmu.

Special Thanks for :

Ibu Hani Nurhayati, M.T, Bapak Dr. Faisal, M.T, Bapak Fachrul Kurniawan, S.T,M.MT, Bapak Fressy,M.T, Dek Alvin yang telah memberikan banyak bimbingan, motivasi, ajakan kebaikan, memberi warna tersendiri dalam suasana akademik dan kehidupan. Semoga Allah membalas segala kebaikan dan selalu dalam lindungan serta rahmat-Nya.

Teman-Teman Kontrakan Joyosuko :

Syifaul, Agus, Haris, Susanto, dan Amrozi yang telah banyak membagi waktu untuk berdiskusi, mendukung dari awal kuliah hingga menjelang lulus.

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

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

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

1. Prof. Dr. H. Mudjia Rahardjo, M.Si selaku rektor UIN Maulana Malik Ibrahim Malang, yang telah banyak memberikan pengetahuan dan pengalaman yang berharga.
2. Dr. drh. Bayyinatul Muchtaromah, M.Si selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Bapak Dr. Cahyo Crysdiyan selaku ketua Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Ibu Hani Nurhayati, M.T dan Bapak Dr. M. Faisal, M.T selaku dosen pembimbing skripsi, yang telah banyak memberikan pengarahan dan pengalaman yang berharga.
5. Segenap civitas akademika jurusan Teknik Informatika, terutama seluruh dosen, terima kasih atas segenap ilmu dan bimbingannya.
6. Ayahanda Abdul Mutholib dan Ibunda Lailatul Badriyah tercinta yang senantiasa memberikan doa, spirit, tenaga, biaya, dan restunya kepada penulis dalam menuntut ilmu.
7. Adik Rifqul Islam Al Fata dan Badruz Zaman Ash-Sholih yang selalu memberikan semangat, canda tawa kepada peneliti untuk menyelesaikan skripsi ini.
8. Semua pihak yang ikut membantu dalam menyelesaikan skripsi ini baik berupa materiil maupun moril.

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

Wassalamu'alaikum Wr. Wb.

Malang, Januari 2014
Peneliti

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
LEMBAR PERSETUJUAN	iii
LEMBAR PENGESAHAN	iv
HALAMAN PERNYATAAN	v
MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiv
ABSTRAK	xv
ABSTRACT	xvi
مستخلص البحث	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	5
1.3 Batasan Masalah	5
1.4 Tujuan Penelitian	6
1.5 Manfaat Penelitian	6
1.6 Metode Penelitian	6
1.7 Sistematika Penulisan	7
BAB II KAJIAN TEORI	9
2.1 Ilmu Tajwid	9
2.2 Game	14
2.3 NPC (<i>Non-Player Character</i>)	20
2.4 Android	21
2.5 Algoritma A*	25
2.5.1 Istilah – Istilah dalam A*	25

2.5.2 Algoritma A*	26
2.5.3 Iterative Deepening A* (IDA*)	28
BAB III ANALISIS DAN PERANCANGAN SISTEM	30
3.1 Analisis Game Sejenis	30
3.1.1 Game Super Mario Bros.....	30
3.1.2 <i>Game</i> Santri Story.....	31
3.1.3 <i>Game Side Scrolling</i> 2d Petualangan Kabayan	32
3.2 Analisis Game Pengenalan Tajwid.....	33
3.2.1 Keterangan Umum	33
3.2.2 FSM (<i>Finite State Machine</i>).....	35
3.2.3 Storyboard Game	38
3.2.4 Storyline	40
3.2.5 Gameplay.....	41
3.2.6 <i>Scoring</i>	42
3.2.7 Tingkat Kesulitan.....	42
3.2.8 Konten-konten Game	43
3.2.9 Skenario Game.....	44
3.3 Perancangan algoritma Iterative Deepening A* pada musuh	45
3.4 Perancangan Aplikasi Game	58
3.4.1 Perancangan Antarmuka Intro.....	58
3.4.2 Perancangan Antarmuka Menu <i>Game</i>	59
3.4.3 Perancangan Antarmuka <i>Game</i>	59
3.4.4 Kebutuhan sistem	62
3.4.5 Cara Memainkan <i>Game</i>	64
3.4.6 Kebutuhan PC Pemain	64
BAB IV HASIL DAN PEMBAHASAN	65
4.1 Implementasi Aplikasi	65
4.3 Uji Coba	78
4.3.1 Uji Coba Algoritma Iterative Deepening A* (IDA*)	78
4.3.2 Uji Coba Pengguna	101
4.4 Integrasi Game Tajwid dengan Islam.....	106

BAB V PENUTUP	108
5.1 Kesimpulan	108
5.2 Saran	109
DAFTAR PUSTAKA	110
LAMPIRAN-LAMPIRAN	111



DAFTAR GAMBAR

Gambar 2.1 Hukum Nun Sukun dan Tanwin	14
Gambar 2.2 Arsitektur android	22
Gambar 3.1 <i>Interface game</i> super mario bros Robin Baumgarten	31
Gambar 3.2 A* dalam <i>game</i> super mario bros	31
Gambar 3.3 Antarmuka depan <i>game</i> santri story	32
Gambar 3.4 Antarmuka <i>game</i> santry story	32
Gambar 3.5 Antarmuka depan <i>game</i> petualangan kabayan	33
Gambar 3.6 Antarmuka <i>game</i> petualangan Kabayan ketika bermain	33
Gambar 3.7 Desain menu utama <i>game</i> tajwid	38
Gambar 3.8 Desain untuk memilih level permainan	38
Gambar 3.9 Desain misi pada setiap level	38
Gambar 3.10 Desain pemain ketika mengambil item huruf	39
Gambar 3.11 Desain pemain ketika mengambil item pertanyaan	39
Gambar 3.12 Desain misi sukses	40
Gambar 3.13 Desain misi gagal	40
Gambar 3.14 Blok diagram implementasi IDA*	46
Gambar 3.15 Tampilan awal koordinat pemain dan musuh	47
Gambar 3.16 Jalur terbaik untuk pergerakan Musuh	57
Gambar 3.17 Musuh memukul <i>player</i>	58
Gambar 3.18 Desain <i>splashscreen game</i>	58
Gambar 3.19 Desain menu <i>game</i>	59
Gambar 3.20 Desain menu pilihan level	60
Gambar 4.1 Tampilan <i>splashscreen game</i> tajwid	65
Gambar 4.2 Tampilan menu utama	65
Gambar 4.3 Tampilan menu <i>setting</i>	66
Gambar 4.4 Tampilan petunjuk permainan	66
Gambar 4.5 Tampilan info pembuat <i>game</i>	67
Gambar 4.6 Tampilan koleksi bacaan	67
Gambar 4.7 Tampilan koleksi bacaan <i>idgham bighunnah</i>	67
Gambar 4.8 Tampilan koleksi bacaan <i>idgham bilaghunnah</i>	68
Gambar 4.9 Tampilan koleksi bacaan <i>iqlab</i>	68
Gambar 4.10 Tampilan koleksi bacaan <i>izh-har</i>	68
Gambar 4.11 Tampilan koleksi bacaan <i>ikhfa' 1</i>	69
Gambar 4.12 Tampilan koleksi bacaan <i>ikhfa' 2</i>	69
Gambar 4.13 Tampilan koleksi bacaan <i>ikhfa' 3</i>	69
Gambar 4.14 Tampilan memilih level	70
Gambar 4.15 Tampilan misi level satu	70
Gambar 4.16 Tampilan <i>game</i> level pertama (<i>idgham bighunnah</i>)	71

Gambar 4.17 Tampilan <i>game</i> level kedua (<i>idgham bilaghunnah</i>)	71
Gambar 4.18 Tampilan <i>game</i> level tiga (<i>iqlab</i>).....	71
Gambar 4.19 Tampilan <i>game</i> level empat (<i>izh-har</i>).....	72
Gambar 4.20 Tampilan <i>game</i> level lima (<i>ikhfa'</i>)	72
Gambar 4.21 Tampilan misi sukses.....	72
Gambar 4.22 Tampilan misi gagal.....	73
Gambar 4.23 Tampilan aplikasi yang akan diuji.....	79
Gambar 4.24 Posisi pemain yang memenuhi persyaratan algoritma	80
Gambar 4.25 Hasil uji coba <i>IDA</i> * 1.....	82
Gambar 4.26 Musuh sampai ke node <i>goal</i> (tujuan).....	82
Gambar 4.27 Grafik <i>path</i> (jalur) terpilih yang dilalui musuh	87
Gambar 4.28 Tampilan aplikasi yang akan diuji.....	88
Gambar 4.29 Hasil uji coba <i>IDA</i> * 2.....	90
Gambar 4.30 Musuh sampai ke node <i>goal</i> (tujuan).....	90
Gambar 4.31 Grafik <i>path</i> (jalur) terpilih yang dilalui musuh	94
Gambar 4.32 Tampilan aplikasi yang akan diuji.....	95
Gambar 4.33 Hasil uji coba <i>IDA</i> * 3.....	97
Gambar 4.34 Musuh sampai ke node <i>goal</i> (tujuan).....	97
Gambar 4.35 Grafik <i>path</i> (jalur) terpilih yang dilalui musuh	101

DAFTAR TABEL

Tabel 3.1 Deskripsi musuh dalam <i>game</i>	44
Tabel 3.2 Daftar spesifikasi komputer	64
Tabel 4.1 Percobaan pertama pada game tajwid	83
Tabel 4.2 Percobaan kedua pada game tajwid.....	91
Tabel 4.3 Percobaan ketiga pada game tajwid	98
Tabel 4.4 Rekapitulasi hasil uji coba responden	103
Tabel 4.5 Analisis Pertanyaan Kuisisioner	104



ABSTRAK

Lidinillah, Muiz. 2014. **Game Pengenalan Tajwid Berbasis Mobile Dengan Metode Iterative Deepening A* untuk Penentuan Arah Gerak Musuh.**

Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.

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

Kata Kunci: *Ilmu Tajwid, Game Side Scrolling, Iterative Deepening A**

Dalam membaca Al-Quran, terdapat ilmu untuk mempelajari tata cara membaca Al-Quran dengan baik dan benar yang dikenal dengan tajwid. Berbagai metode dan media belajar dibuat untuk memudahkan dalam pembelajaran. Salah satunya adalah media *game*. *Game* pengenalan tajwid ini berbentuk *side scrolling 2D* dimana pemain berpetualang untuk belajar ilmu tajwid dasar. Ilmu tajwid tersebut yakni hukum nun sukun dan tanwin. Tujuan dari permainan adalah mengumpulkan item-item huruf bacaan ilmu tajwid dan permata untuk menjawab pertanyaan yang tersebar di area permainan. Contoh-contoh bacaan tajwid akan diperoleh bila misi berhasil.

Dalam *game* ini, musuh akan diterapkan kecerdasan buatan untuk menghadang *player*. Algoritma yang digunakan adalah Iterative Deepening A*. Algoritma ini digunakan untuk menentukan jalur terpendek bagi musuh dalam menemukan *player*. Pengujian dilakukan dengan memperhitungkan waktu penyelesaian dan hasil dengan menggunakan perangkat *mobile* Android. Dari uji coba yang dilakukan, hasil penelusuran jalur menggunakan algoritma ini ditemukan solusi dengan waktu tempuh rata-rata 0.31 detik.

ABSTRACT

Lidinillah, Muiz. 2014. **Mobile-Based Game Introduction to Tajweed With Iterative Deepening A* Method to Determine the Direction of Enemy Motion**. Thesis. Informatics Department of Faculty of Science and Technology. Maulana Malik Ibrahim State Islamic University, Malang.
Adviser: (I) Hani Nurhayati, M.T (II) Dr. M. Faisal, M.T

Keywords: *Science tajweed, Game Side Scrolling, Iterative Deepening A**

In reciting the Quran, there is a science to learn how to read the Quran properly known as Tajweed. A variety of learning methods and media were made to facilitate the learning. One of this media is game. The game to introduce the form of tajweed was 2D side scrolling adventure in which players learn the basic of tajweed. The subject of tajweed included in this game was the “*nun sukun dan tanwin*” law. The objective of the game was to collect items of tajweed typefaces and gems to answer questions spreading over the game area. Examples of recitation readings will be obtained when the mission completed successfully.

In this game, the enemies will be applied artificial intelligence to block the player. The algorithm used is Iterative Deepening A*. This algorithm is used to determine the shortest path to the enemy in finding players. Testing was conducted by taking into account the time of completion and results by using Android mobile devices. According the experiments conducted, the results of the search paths using the algorithm found a solution with an average travel time of 0,31 seconds.

مستخلص البحث

معز لدين الله. ٢٠١٤. المحمول القائم على لعبة مقدمة في التجويد مع *Iterative Deepening A** طريقة لتحديد اتجاه العدو الحركة . البحث . شعبة المعلوماتية من كلية العلوم والتكنولوجيا . في جامعة مولانا مالك إبراهيم الإسلامية الحكومية بمالانج . المشرف : (١) هاني نورحياتي، الماجستير (٢) الدكتور محمد فيصل، الماجستير

المصطلحات: التجويد العلوم ، لعبة الجانب التمرير، *Iterative Deepening A**

في تلاوة القرآن الكريم ، وهناك العلم لتعلم كيفية قراءة القرآن الكريم بشكل صحيح كما هو معروف التجويد . وقدمت مجموعة متنوعة من أساليب التعلم ووسائل الإعلام لتسهيل التعلم. واحدة من هذه الوسائط هي لعبة . كانت اللعبة لتقديم شكل التجويد 2D الجانب التمرير المغامرة التي لاعبين في تعلم التجويد الأساسية . كان موضوع التجويد المدرجة في هذه اللعبة الاحكام "نون سكون و تنوين" . كان الهدف من اللعبة لجمع المواد من المحارف التجويد و الأحجار الكريمة للإجابة على الأسئلة تمتد على مساحة اللعبة. وسيتم الحصول على أمثلة من القراءات تلاوة عند الانتهاء من المهمة بنجاح.

في هذه اللعبة ، سيتم تطبيق أعداء الذكاء الاصطناعي لمنع لاعب. الخوارزمية المستخدمة هي *Iterative Deepening A** . يتم استخدام هذه الخوارزمية لتحديد أقصر طريق للعدو في العثور على لاعبين. وأجري الاختبار مع مراعاة وقت الانتهاء والنتائج باستخدام الأجهزة النقلة الروبوت. وفقا تجارب التي أجريت ، وجدت نتائج مسارات البحث باستخدام خوارزمية الحل مع الوقت الذي يستغرقه السفر معدله ٣١،٠ ثانية.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Al-Quran merupakan firman Allah yang bernilai pahala bagi yang pembacanya. Dalam membaca Al-Quran, dituntut agar dibaca dengan tartil. Artinya, seseorang yang membaca Al-Quran diwajibkan untuk mempelajari ilmu tentang tata cara membaca Al-Quran yang dikenal dengan ilmu tajwid. Imam Jalaluddin memberikan pengertian tentang tajwid yakni dengan mengembalikan huruf-huruf hijaiyah kepada *makhraj* dan sifatnya, menghaluskan pelafalan dengan sempurna tanpa berlebih-lebihan, tergesa-gesa maupun dipaksakan. Dalam ilmu qira'ah, tajwid adalah mengeluarkan huruf dari tempatnya disertai sifat-sifat yang dimilikinya. Dengan adanya ilmu tajwid, dapat memberikan tuntunan bagaimana cara pengucapan ayat Al-Quran yang tepat, sehingga lafal dan maknanya terpelihara.

Menurut Syekh Ibnu Jazary (dalam Wahyudi,2007), wajib hukumnya membaca Al-Quran dengan menggunakan ilmu tajwid. Sesungguhnya Allah menurunkan Al-Quran beserta tajwidnya. Oleh karena itu, hukumnya berdosa bagi siapa saja yang membaca Al-Quran tanpa memakai tajwid. Dasar hukum wajibnya membaca Al-Quran dengan tajwid terdapat dalam surat Al-Muzzammil: 4

﴿ وَرَتَّلِ الْقُرْآنَ تَرْتِيلاً ﴾

dan bacalah Al-Quran itu dengan perlahan-lahan (bertajwid).

Ayat ini memerintahkan kepada setiap muslim agar selalu membaca Al-Quran dengan tidak tergesa-gesa sehingga membantu dalam memahami dan merenungi kandungan isi Al-Quran (Wahyudi,2007).

Dalam mempelajari ilmu tajwid, banyak sekali macam-macam bacaan-bacaan yang harus dikuasai. Berbagai metode dan media belajar dibuat untuk memudahkan dalam pembelajaran. Teknologi *mobile device* pun dikembangkan untuk dijadikan salah satu media dalam hal pembelajaran. *Smartphone* merupakan teknologi *mobile device* yang sedang berkembang. *Smartphone* dijalankan pada sistem operasi yang beragam, seperti *BlackBerry*, *Windows Mobile*, *IOS*, *Symbian OS* dan *Android*. Pengguna sistem operasi *Android* di Indonesia mengalami peningkatan melebihi sistem operasi lainnya(StatCounter,2014). Kondisi ini dimungkinkan karena mudahnya dalam mengembangkan sistem operasi *android* dikarenakan bersifat *open source*. Pihak-pihak *developer* mempunyai hak penuh untuk mengembangkan berbagai aplikasi untuk menyalurkan kreativitas maupun bakatnya. Tidak dapat dipungkiri, berbagai macam aplikasi terkait *game* juga tak lepas untuk dikembangkan.

Game sendiri memiliki banyak sekali *genre*, diantaranya yang sedang berkembang adalah *game* edukasi. *Genre* ini memiliki ciri yang fokus pada isi dan tujuan pembuatan *game*, bukan *genre* yang sesungguhnya (Aeni,2009). Untuk membuat *game* edukasi yang menarik, perlu adanya penggabungan dari berbagai *genre*. Salah satu *genre* yang menarik adalah *side scrolling game*.

Game ini memiliki sudut pandang permainan dari samping. Karakter utama dapat bergerak dari kiri ke kanan dan *background* pada *game* seolah-olah bergeser mengikuti pergerakan karakter utama. Dalam membangun *game side scrolling*, terdapat beberapa komponen yakni grafik (*visual*), suara (*audio*), kecerdasan buatan (*Artificial Intelligence*), dan skenario cerita. Keempat komponen tersebut harus dapat didesain dengan baik agar *game* yang akan dibuat menjadi menarik untuk dimainkan.

Salah satu komponen terpenting agar *game* menjadi lebih hidup adalah dengan disisipkannya kecerdasan buatan. Dalam membuat *game*, kecerdasan buatan dapat diterapkan disalah satu, sebagian atau seluruh elemen-elemen *game* yang meliputi karakter, skenario, level, penentuan skor, atau *obstacle* (objek lain). Dalam *game side scrolling*, karakter musuh menjadi salah satu elemen penting. Karakter musuh bertindak sebagai penghalang bergerak agar pemain tidak mudah dalam mencapai misi. Berbagai teknik kecerdasan buatan disisipkan dalam karakter musuh. Salah satunya adalah dengan menentukan jalur terpendek dalam mengejar *player*. Terdapat berbagai macam algoritma untuk menentukan jalur terpendek, diantaranya: *Generate-and-Test*, *Hill Climbing*, *Simulated Annealing*, *Best-First Search*, *Greedy Best-First Search*, dan *A**. Menurut Suyanto (2011), dari beberapa algoritma tersebut, *A** merupakan algoritma *complete* dan *optimal* karena memperhitungkan biaya sebenarnya dan biaya perkiraan. Sedangkan algoritma lainnya hanya memperhitungkan biaya perkiraan sehingga tidak *optimal*. Meskipun algoritma ini *complete* dan *optimal*, hal ini akan menjadi masalah jika

diterapkan dalam *mobile device* yang memiliki keterbatasan memori. Hal ini terjadi karena algoritma ini akan membangkitkan dan menyimpan simpul ke dalam memori. Sehingga jika simpul yang dibangkitkan berjumlah besar, maka memori yang dibutuhkan juga besar. Oleh karena itu, diusulkan berbagai variasi dari algoritma ini. Salah satunya yaitu Iterative Deepening A* (IDA*). Pada algoritma ini, setiap simpul yang dibangkitkan dan disimpan dalam memori akan dihapus dan diganti dengan simpul baru sebagai batasan untuk mencari simpul berikutnya. Hal ini akan meminimalisir penggunaan memori.

Hasil survei *online* yang dilakukan oleh peneliti kepada 59 responden usia 15 – 23 tahun, ditemukan bahwa 66 % seseorang belajar membaca Al-Quran melalui TPQ (Taman Pendidikan Al-Quran), 15 % belajar di Musholla (tempat ibadah), 14 % belajar dari orang tua sendiri dan 5% belajar di tempat lain. Dari responden yang ada, media belajar ilmu tajwid yang biasa digunakan, sebanyak 90 % belajar melalui buku pedoman ilmu tajwid, 3 % melalui software komputer dan 7 % tidak memiliki. Peneliti juga melakukan survei kepada responden yang menggunakan ponsel android ke dalam beberapa kategori, sebanyak 33 % menggunakan ponsel android untuk *e-learning*, sebanyak 21 % untuk sarana hiburan, 4 % untuk multiguna dan sebanyak 42 % untuk jejaring sosial

Berdasarkan survei tersebut, media belajar Al-Quran masih banyak yang menggunakan buku ilmu tajwid, sedangkan ponsel android banyak digunakan untuk jejaring sosial. Pengguna android sendiri sudah cukup banyak, tetapi pemanfaatan ponsel pintar tersebut sebagai media pembelajaran

Al-Quran belum maksimal. Oleh karena itu, peneliti membuat alternatif media belajar lain berbentuk *game* dengan mengambil judul **”Game Pengenalan Tajwid Berbasis Mobile Dengan Metode Iterative Deepening A* untuk Penentuan Arah Gerak Musuh”**. Penelitian ini akan membuat *game side scrolling* untuk mengenalkan ilmu tajwid tepatnya hukum nun sukun dan tanwin. Algoritma Iterative Deepening A* akan diterapkan pada karakter musuh untuk menentukan jalur terpendek dalam mengejar *player*. *Game* ini akan diimplementasikan pada *mobile device* dengan sistem operasi android.

1.2 Rumusan Masalah

Berdasarkan penjelasan pada latar belakang di atas, maka rumusan masalah dalam penelitian ini yaitu :

1. Bagaimana membuat *game* pengenalan tajwid menggunakan algoritma Iterative Deepening A* sebagai penentuan jalur terpendek musuh untuk mengejar *player*?

1.3 Batasan Masalah

Adapun batasan masalah dalam penelitian ini yaitu :

1. Ilmu tajwid yang digunakan sebagai materi *game* adalah hukum nun sukun dan tanwin menggunakan metode Qira’ati.
2. Contoh-contoh bacaan diambil dari penggalan ayat-ayat al-quran
3. *Game* bersifat *single player*
4. *Game* bertipe 2 dimensi dan berbasis mobile dengan OS Android
5. *Game* ditujukan untuk usia minimal 10 tahun

6. Penentuan arah gerak musuh adalah menentukan jalur terpendek musuh dalam mengejar *player*.

1.4 Tujuan Penelitian

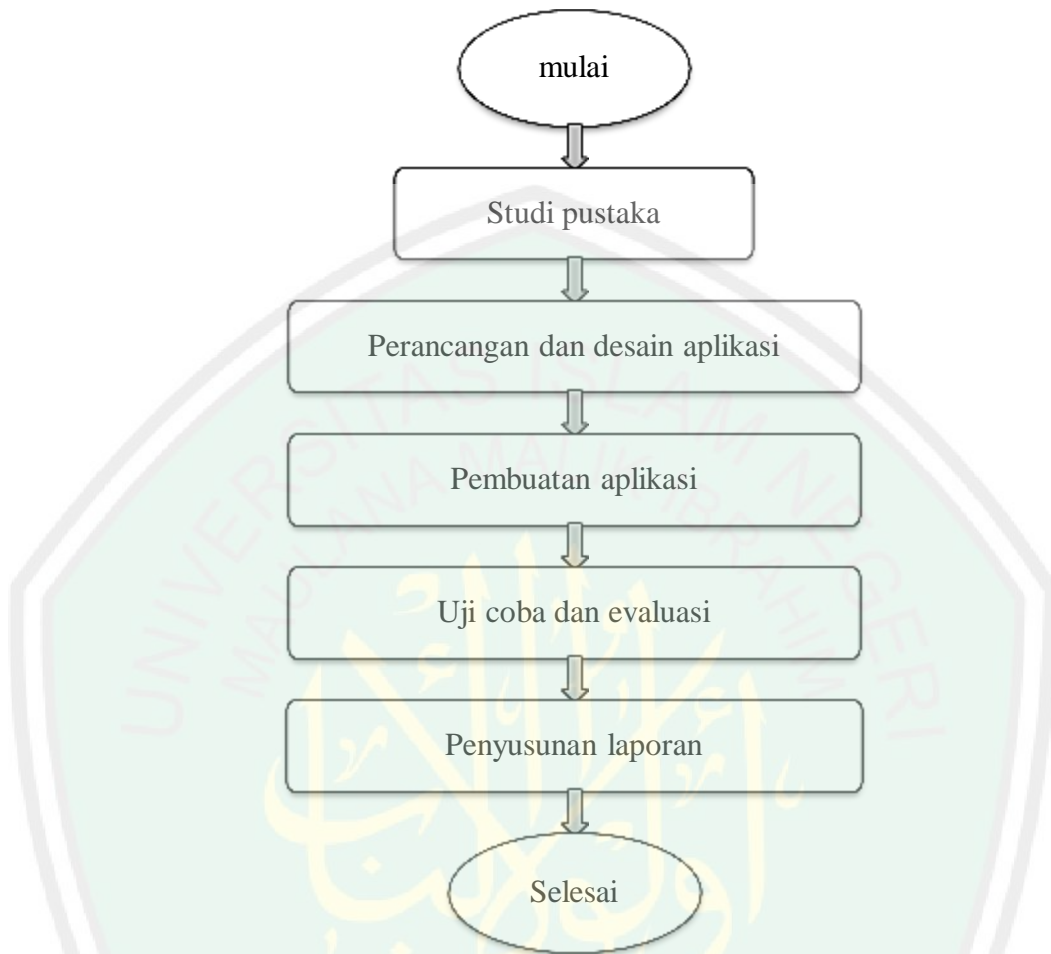
Penelitian ini bertujuan untuk membuat aplikasi *game* pengenalan tajwid yang diterapkan algoritma Iterative Deepening A* untuk menentukan arah gerak musuh (jalur terpendek musuh dalam mengejar *player*).

1.5 Manfaat Penelitian

Manfaat pembuatan aplikasi permainan ini adalah mengenalkan huruf-huruf dan contoh bacaan yang terdapat pada ilmu tajwid melalui permainan sebagai media belajar alternatif.

1.6 Metode Penelitian

Berdasarkan jenis dan analisis data yang dilakukan, metode penulisan skripsi ini menggunakan jenis penelitian studi dokumen dan bentuk analisis berupa analisis isi (*content analysis*). Pengumpulan data juga dilakukan dengan cara melakukan uji coba aplikasi pada pengguna yang disertai dengan kuisioner. Adapun tahap penelitian adalah sebagai berikut :



1.7 Sistematika Penulisan

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

BAB I Pendahuluan

Pendahuluan, membahas tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian, dan sistematika penulisan tugas akhir.

BAB II Kajian Teori

Kajian teori berisikan beberapa teori yang mendasari dalam penyusunan tugas akhir ini. Adapun yang dibahas dalam bab ini adalah penelitian

terkait dari *game* yang akan dibangun, dasar teori pendukung tentang *game* dan teori-teori keislaman sebagai materi dalam *game*.

BAB III Analisa dan Perancangan

Menganalisa kebutuhan sistem untuk membuat *game* meliputi spesifikasi kebutuhan *hardware*, *software* dan langkah-langkah pembuatan *game*.

BAB IV Hasil dan Pembahasan

Menjelaskan tentang hasil pembuatan dan pengujian *game* yang telah diterapkan algoritma.

BAB V Penutup

Berisi kesimpulan dan saran.



BAB II

KAJIAN TEORI

2.1 Ilmu Tajwid

Tajwid secara bahasa berasal dari kata “*jawwada-yujawwidu-tajwidan*” yang artinya membaguskan atau membuat jadi bagus. Imam Jalaluddin memberikan pengertian tentang tajwid yakni dengan mengembalikan huruf-huruf hijaiyah kepada *makhraj* dan sifatnya, menghaluskan pelafalan dengan sempurna tanpa berlebih-lebihan, tergesa-gesa maupun dipaksakan.

Tujuan dalam mempelajari ilmu tajwid yakni agar mampu membaca Al-Quran secara fasih sesuai dengan apa yang diajarkan oleh Rasulullah SAW, dapat memelihara lisan dari kesalahan-kesalahan ketika membaca kitab Allah SWT (Wahyudi, 2007). Hukum mempelajari ilmu tajwid adalah *farḍlu kifayah* yang artinya tidak diharuskan bagi setiap orang, cukup diwakili oleh beberapa orang saja. Namun, hukum membaca Al-Quran dengan menggunakan aturan tajwid adalah *farḍlu ‘ain* atau merupakan kewajiban pribadi, karenanya apabila seseorang membaca Al-Quran dengan tidak menggunakan ilmu tajwid, hukumnya berdosa.

Dalam ilmu tajwid, terdapat beberapa hukum bacaan, yaitu :

1. Hukum nun sukun dan tanwin
2. Hukum mim mati
3. Hukum *idgham*
4. Hukum mim dan nun bertasydid

5. Hukum lam *ta'rif*
6. Hukum lam *fi'il*, lam isim, dan lam huruf
7. Hukum *ra'*
8. Hukum Lam *Jalalah*
9. *Qolqolah*
10. Hukum *Madd*

Dalam *game* ini, materi tajwid yang digunakan adalah hukum nun sukun dan tanwin. Pembagian hukum nun sukun dan tanwin adalah sebagai berikut (Wahyudi,2007:91) :

a. *Idgham bighunnah*

Secara bahasa *idgham* artinya “memasukkan”, *bighunnah* artinya “dengan dengung”. *Idgham bighunnah* memiliki 4 huruf ya' (ي) , nun (ن) , mim (م) , wawu (و) yang terkumpul dalam *lafadz* يَنْمُو

Cara membaca *idgham bighunnah* adalah dengan memasukkan suara nun mati atau tanwin kepada huruf *idgham bighunnah* yang ada dihadapannya sehingga menjadi satu ucapan, seakan-akan satu huruf. Pada waktu meng-*idgham*-kan, suara harus ditasydidkan kepada huruf *idgham bighunnah* yang ada didepan nun mati atau tanwin, kemudian ditahan kira-kira dua ketukan dengan memakai *ghunnah* (sengau) ketika membacanya. Berbeda dengan hukum lainnya, disini hanya terjadi dua kata dan tidak terjadi pada satu kata.

Contoh :

وَجُوهٌ يَوْمَئِذٍ	-	أَنْ يُكَبِّرُوا = ي
حِطَّةً نَعْفِرُكُمْ	-	لَنْ نَصْبِرَ = ن
قَوْلًا مَعْرُوفًا	-	مِنْ مَفْعَدِهِمْ = م
نَفْسٍ وَاحِدَةٍ	-	مِنْ وَرَاءِ يَهُمْ = و

b. *Idgham bilaghunnah*

Bilaghunnah artinya tidak memakai *ghunnah* (dengung/sengau). Hukum *idgham bilaghunnah* terjadi apabila nun mati atau tanwin bertemu dengan salah satu dari dua huruf, yaitu lam (ل) dan ra (ر).

Cara membaca *idgham bilaghunnah* adalah dengan memasukkan suara nun mati atau tanwin sepenuhnya kepada huruf lam atau ra tanpa memakai dengung. Pada waktu meng-*idgham*-kan, suara harus ditasydidkan kepada huruf lam dan ra seraya menahan sejenak.

Contoh :

خَيْرٌ لَكَ	-	مِنْ لَدُنْكَ = ل
غَفُورٌ رَحِيمٌ	-	مِنْ رَبِّكُمْ = ر

c. *Iqlab*

Iqlab menurut bahasa ialah memindahkan sesuatu dari bentuk asalnya (kepada bentuk lain). Hukum *iqlab* terjadi apabila nun mati atau tanwin bertemu dengan huruf *ba'* (ب).

Cara membaca *iqlab* adalah dengan mengubah suara nun mati atau tanwin menjadi mim. Kedua bibir dirapatkan untuk mengeluarkan bunyi

dengan dibarengi dengung yang keluar dari pangkal hidung. Kemudian ditahan sejenak kira-kira dua ketukan.

Contoh :

سَيِّئَةٌ بِمَا - مِنْ بَعْدِ = ب

d. *Idzhar*

Idzhar menurut bahasa berarti penjelasan, penerangan, dan pengungkapan. Hukum *idzhar* terjadi apabila menghadapi salah satu dari huruf *halq*. Huruf *idzhar* yaitu hamzah (أ), ha' (ح), kho' (خ), 'ain (ع), ghain (غ), ha (ه).

Cara membacanya adalah dengan mengeluarkan setiap huruf dari *makhroj*-nya tanpa memakai dengung pada huruf yang dibaca *idzhar*.

فَوَاحِشَةً أَوْ مَمْلُوكَةً - وَإِنْ أَرَدْتُمْ

عَلَيْهَا حَكِيمًا - يَنْحِتُونَ

نَارًا خَالِدًا - وَإِنْ خِفْتُمْ

مَيْلًا عَظِيمًا - مِنْ عِنْدِكَ

مَيْتَافًا غَلِيظًا - مِنْ غَلِيظٍ

جُرْفٍ هَارٍ - مِنْهَا

e. *Ikhfa'*

Ikhfa' menurut bahasa adalah samar atau tertutup. Hukum *Ikhfa'* terjadi apabila nun mati atau tanwin menghadapi salah satu huruf *ikhfa'* yang berjumlah lima belas, yakni : ta (ت), tsa (ث), jim (ج), dal (د),

dzal (ذ), za (ز), sin (س), syin (ش), shad (ص), dhat (ض), tha (ط), zha (ظ), fa (ف), qaf (ق), kaf (ك)

Cara membaca huruf *ikhfa'* adalah dengan memadukan antara suara nun mati atau tanwin dengan suara huruf *ikhfa'* yang ada dihadapannya. Saat proses *ikhfa'* berlangsung, suara ditahan sejenak kira-kira dua ketukan, baru disambung dengan pengucapan huruf *ikhfa'*.

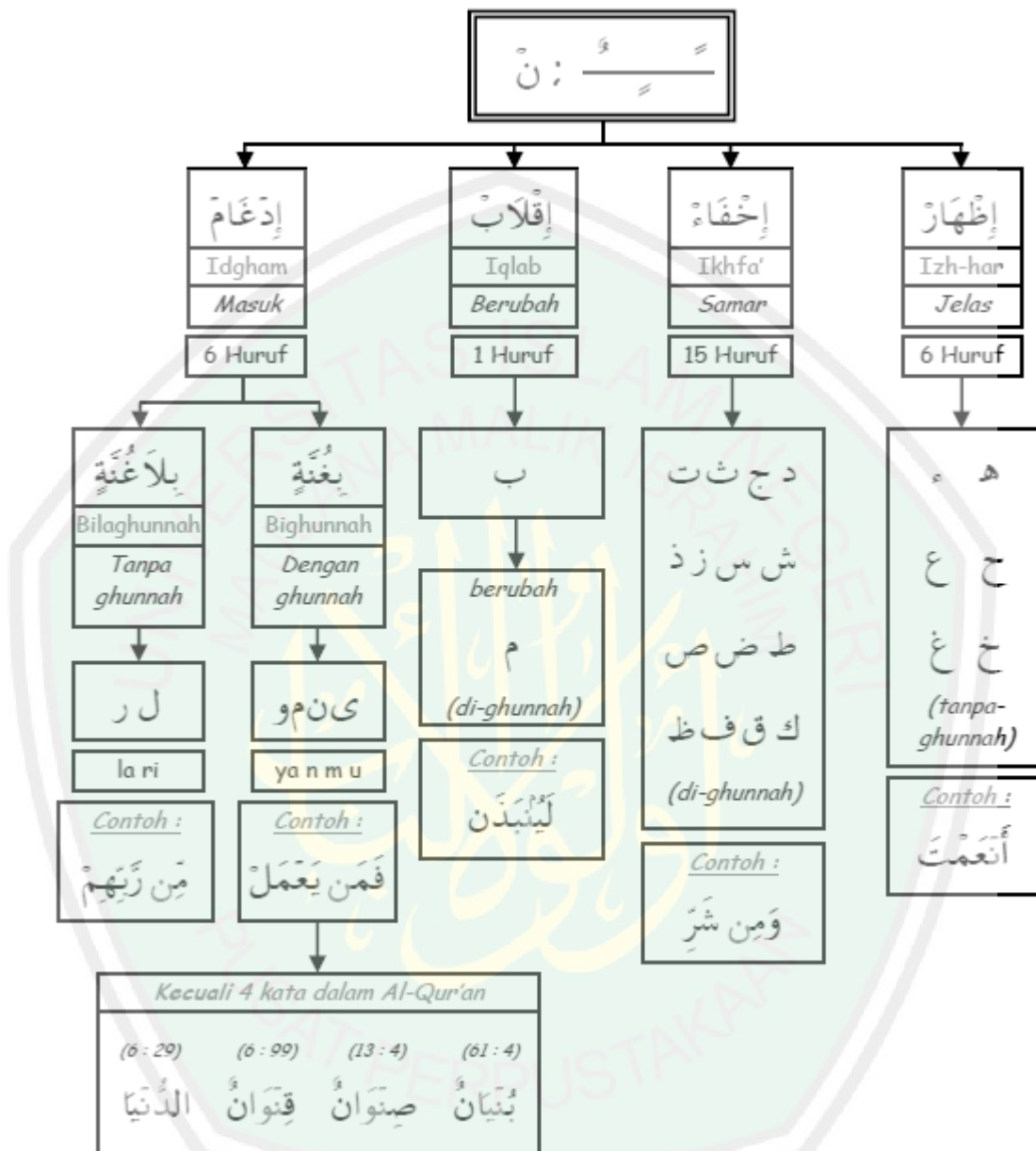
بَيِّنَاتٍ تَعْرِفُ - وَلَا أَنْتُمْ

أَيَّامٍ نُّمَّ - مَنْثُورًا

كُنُسَادِهَاقًا - أَنْدَادًا

أَنْقَضَ - مِنْ قَبْلِكَ

أَجْرًا كَبِيرًا - مَنْ كَانَ



Gambar 2.1 Hukum Nun Sukun dan Tanwin (Zaky, 2008)

2.2 Game

Game adalah permainan yang menarik dan menyenangkan. *Game* merupakan permainan elektronik yang menjadi fenomena global. *Game* menggunakan media komputer, telepon seluler maupun konsol seperti *playstation* atau *x-box* (Aeni, 2009).

- Jenis *Game*

Jenis-jenis *game* lebih dikenal dengan istilah *genre*. Samuel Henry (dalam Aeni,2009) mendefinisikan *genre*

“sebagai format atau gaya dari sebuah *game*. Format sebuah *game* bisa murni sebuah *genre* atau campuran (*hybrid*) dari beberapa *genre* lain dengan maksud membuat unsur permainan lebih bervariasi dan menantang”.

Berbagai jenis *game* yang beredar (Aeni,2009), beberapa diantaranya adalah:

- *Maze Game*

Game jenis ini berupa mengelilingi maze (lorong-lorong yang saling terhubung) dan memakan item-item untuk mendapatkan skor, tenaga, nyawa, kekebalan dan lain-lain. Ketika pemain mendapatkan item kekebalan, pemain dapat berbalik mengejar musuh. Contoh : *pacman*.

- *Board Game*

Jenis *game* berbentuk papan permainan, seperti *game* monopoli. *Game* ini mengandalkan kemampuan komputer untuk menjadi lawan pemain. Oleh karena itu, kecerdasan buatan komputer menjadi kunci dalam *game*. Semakin cerdas komputer, akan sulit untuk dikalahkan. Namun, menambah *game* ini menjadi lebih menantang untuk dimainkan.

- *Puzzle Game*

Genre ini banyak sekali macamnya, ada yang menyusun gambar, mengisi kotak-kotak kosong, menyusun kotak dengan menjatuhkan sesuatu dari sisi sebelah atas ke bawah untuk membentuk garis lurus. *Game* ini menghandalkan ketekunan, kecepatan, dan daya ingat yang tinggi. Contoh dari jenis *game puzzle* adalah *game* teka nusantara, tetris.

- *Fighting Game*

Game yang berisi pertarungan antara pemain dengan pemain lainnya atau pemain lawan komputer. *Game* ini memiliki kombinasi berbagai gerakan dalam pertarungan, seperti : memukul, menendang, menembak, dan sebagainya. *Game* ini dibuat dengan mengadopsi seni bela diri, atau ada juga gerakan yang liar. Kebanyakan musuh dalam *game* berupa karakter yang tidak masuk akal sama sekali. Contoh : *Street Fighter*.

- *Racing Game*

Game berbentuk balapan yang mengadopsi dari balapan sungguhan. Tempat lintasannya di desain mirip dengan aslinya, di dalam arena balap atau diluar arena balap. Pada *game* ini, pemain berlomba kecepatan, ketangkasan dalam mengemudikan kendaraan. Contoh *Racing game* adalah *Need for Speed*.

- *Turn Based Strategy Game*

Game ini memerlukan berbagai strategi dari pemain untuk memenangkan permainan. *Game* ini memiliki versi *offline* maupun *online*. Mekanisme permainan dengan cara menunggu pemain melakukan gerakan terlebih dahulu. Hampir serupa dengan catur tetapi dengan variasi gerakan dan efek yang jauh lebih banyak. Contoh *Game* ini adalah *Empire*.

- *Real Time Strategy Game*

Jenis *game* yang bertipe strategi yang kompleks, dimana pemain diajak untuk bergerak pintar agar misi yang ada dalam *game* tercapai. Sedikit berbeda dengan *Turn Based Strategy* yang harus menunggu pemain lain, maka pada *genre* ini tidak perlu menunggu, pemain yang memiliki strategi yang baiklah kemungkinan terbesar untuk menang. Pada permainan ini pemain harus melakukan berbagai gerakan sesuai dengan strategi. Contoh : *Warcraft. Clash of Clans*.

- *Role Playing Game*

Game yang mengutamakan skenario cerita dan biasanya pemain diajak masuk ke dalam cerita tersebut untuk menyelesaikan misi. Pemain akan berperan menjadi seorang karakter yang memiliki berbagai atribut, seperti kesehatan, intelegensi, kekuatan dan keahlian. Contoh : *Harvest Moon*.

- *Simulations Game*

Game berbentuk simulasi sesuatu yang dibuat seperti keadaan sebenarnya. Beberapa jenis *game* ini mengajak pemain untuk menciptakan lingkungan yang diinginkan, seperti membangun simulasi sebuah kota, negara atau koloni. Pemain mengatur berbagai sumber daya, menentukan berbagai keputusan yang diinginkan dalam proses pembangunan yang sedang terjadi. Contoh : *Sim City*.

- *Adventure Game*

Game dengan *genre* petualangan. Pemain diajak untuk berjalan ke suatu tempat. Disepanjang perjalanan pemain menemukan berbagai hal dan peralatan yang akan disimpan. Peralatan ini akan digunakan untuk menjadi petunjuk atau senjata untuk menghadapi rintangan. Porsi pertarungan tidak menonjol dalam *game*. Pada umumnya, *game* lebih kepada pemecahan misteri. Contoh : *Beyond Good and Evil*.

- *Educational Game*

Genre permainan yang ditujukan untuk media pembelajaran. *Game* ini lebih mengacu kepada isi dan tujuan *game* bukan *genre* yang sesungguhnya, seperti *game arabic* sudoku, *game* pembelajaran huruf hijaiyah. Tujuan *genre* ini untuk memancing minat belajar sambil bermain.

- *Side Scrolling Game*

Penekanan permainan pada *genre* ini adalah pemain bergerak ke samping mengikuti rute dan menyelesaikan tugasnya. Ada yang melompat, berlari, mengendap dan menghindari halangan seperti jurang. Contoh *game* yang terkenal adalah Mario Bross.

- Elemen Dasar *Game*

Menurut Teresa Dillon (dalam Aeny,2010) elemen-elemen dasar sebuah *game* adalah:

- *Game Rule*

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

- *Plot*

Plot berisi informasi-informasi tentang hal-hal yang akan dilakukan oleh pemain dalam memainkan *game* secara detail, perintah tentang misi yang harus dicapai dalam *game*.

- *Thema*

Pesan moral yang disampaikan dalam *game*.

- *Character*

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

- *Object*

Sebuah benda yang sangat penting dan biasanya digunakan pemain sebagai petunjuk untuk memecahkan masalah, adakalanya pemain harus punya keahlian dan pengetahuan untuk bisa memecahkannya.

- *Text, grafik dan sound*

Dalam *game*, harus memiliki kombinasi dari media *text*, grafik, maupun *sound* agar terkesan menarik untuk dimainkan.

- *Animasi*

Merupakan hal pokok yang melekat pada dunia *game*, khususnya untuk gerakan karakter-karakter yang ada dalam *game*, properti dari objek.

- *User Interface*

Fitur-fitur antarmuka yang mengkomunikasikan user dengan *game*.

2.3 NPC (*Non-Player Character*)

NPC (*Non-Player Character*) dapat disebut dengan *Autonomous character*. *Autonomous character* merupakan jenis *otonomous agent* yang digunakan komputasi komputer dan media interaktif seperti *game* (Arif, 2010). *Non-player character* merupakan karakter dalam suatu *game* yang dijalankan oleh sistem atau *user* lain dalam *game online*. Pada *game* tajwid ini, musuh yang ada disetiap permainan menjadi karakter *NPC* yang digerakkan sistem. Musuh ini diberi kecerdasan buatan menggunakan

algoritma Iterative Deepening A* untuk menemukan jalur terpendek ke *player*.

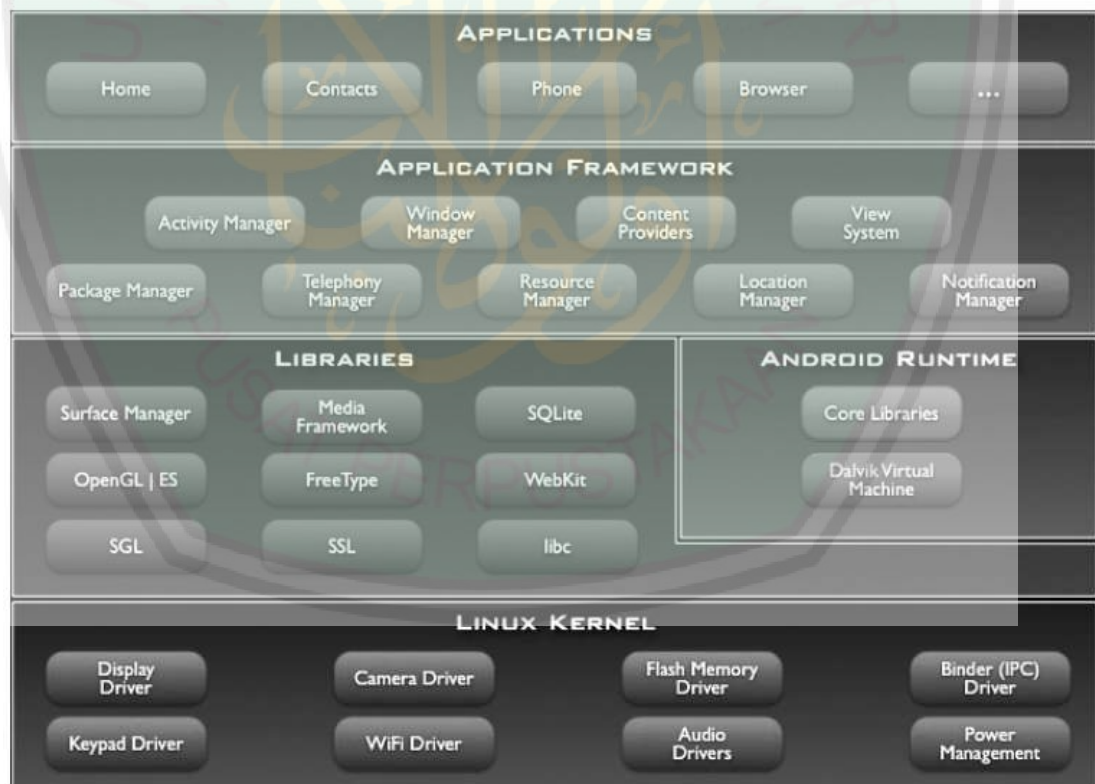
2.4 Android

Android merupakan sistem operasi berbasis *linux* yang diperuntukkan untuk perangkat mobile dimana mencakup *middleware* dan aplikasi (Safaat,2001). Sistem operasi android diambil alih oleh *Google* pada bulan Juli 2005, dan baru dirilis pada 5 November 2007. Pihak ketiga diperbolehkan untuk mengembangkan aplikasi android dengan menyertakan term yang sama dikarenakan berlisensi di bawah *GNU, General Public Lisensi Versi 2 (GPLv2)*. Pendistribusiannya di bawah Lisensi *Apache Software (ASL/Apache2)*, yang memungkinkan untuk distribusi kedua dan seterusnya.

Adapun arsitektur android adalah sebagai berikut (Safaat, 2001) :

1. *Application* dan *Widgets*, layer yang berhubungan dengan aplikasi untuk browsing dan fungsi-fungsi seperti telepon dan sms.
2. *Application Frameworks*, layer dimana para pembuat aplikasi melakukan pengembangan dan pembuatan aplikasi yang akan dijalankan di sistem operasi android. Beberapa komponen yang terdapat pada layer ini adalah *Views, Content Provider, Resource Manager, Notification Manager* dan *Activity Manager*.
3. *Libraries*, layer yang dapat digunakan untuk menjalankan aplikasi. *Library* yang disertakan seperti *library* untuk pemutaran audio dan video, tampilan, grafik, *SQLite, SSL* dan *Webkit*, dan 3D.

4. *Android Run Time*, layer yang memiliki *Core Libraries* dan *Dalvik Virtual Machine* (DVK). *Core libraries* memiliki fungsi untuk menerjemahkan bahasa Java/C. Sedangkan DVK adalah virtual mesin berbasis *register* yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien.
5. *Linux Kernel*, layer yang merupakan inti dari sistem operasi android yang berisi *file-file* sistem untuk mengatur sistem proses, *memory*, *resource*, dan *driver*. *Linux kernel* yang digunakan Android adalah *linux kernel release 2.6*.



Gambar 2.2 Arsitektur android (Zechner, 2011)

Android telah merilis beberapa versi yang memiliki perbedaan disetiap versi-versi tersebut. Berikut ini merupakan versi android yang telah rilis:

1. Android versi 1.0 dan 1.1, versi rintisan awal yang terdapat fitur jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan *gmail*, dan pemberitahuan email.
2. Android versi 1.5 Cupcake, terdapat beberapa pembaruan agar lebih stabil dan *powerful* untuk *smartphone* dan tablet. Penambahan beberapa fitur yakni merekam dan menonton video dengan modus kamera, fitur *bluetooth* A2DP, animasi pada layar, dan *keyboard touchscreen* yang dapat disesuaikan dengan sistem.
3. Android versi 1.6 Donut, dirilis September 2009. Versi ini memperbaiki *bugs* dari versi sebelumnya. Selain itu, terdapat beberapa penambahan fitur yakni kamera dan video yang lebih baik, proses pencarian internet yang lebih baik dari sebelumnya, fitur navigasi, penggunaan baterai indikator dan kontrol *applet* VPN.
4. Android versi 2.0/2.1 Éclair, versi 2.0 dirilis pada bulan Oktober 2009 dan 2.1 dirilis pada Januari 2010. Éclair ini dilengkapi dengan beberapa fitur tambahan, seperti : *flash*, *digital zoom*, *bluetooth* 2.1, fitur *multitouch*, *live wallpaper*, daftar kontak, peningkatan *Google Maps*, perubahan *UI* pada browser dan dukungan HTML5.
5. Android versi 2.2 Froyo: Frozen Yoghurt, dirilis Mei 2010. Android versi ini hadir dengan perubahan antara lain dukungan *Flash* 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi *javascript*, pemasangan aplikasi pada *SD Card*, *WiFi Hotspot* portabel, Gif animasi.

6. Android Versi 2.3 Gingerbread, dirilis Desember 2010. Penambahan fitur untuk *copy paste*, desain ulang UI, SIP *internet calling*, *wireless NFC*, *dual camera*, penambahan fitur sensor seperti *barometer*, *gravimeter*, *download manager*, mendukung format video VP8 dan WebM.
7. Android versi 3.0 / 3.1 Honeycomb, dibuat khusus untuk tablet. Android versi ini mendukung ukuran layar yang lebih besar. Versi ini mendukung *multiprocessing*, akselerasi *hardware* untuk grafis.
8. Android versi 4.0 ICS : Ice Cream Sandwich, dirilis pada Oktober 2011. Terdapat penambahan fitur baru yakni *facelock*, pemantauan penggunaan dan kontrol, berbagi informasi menggunakan NFC, mencari email secara *offline*.
9. Android versi 4.1/4.2/4.3 Jelly Bean, versi ini dilengkapi dengan penambahan beberapa fitur yakni: perbaikan desain UI, *multi restricted profiles*, *bluetooth smart technology*, peningkatan performa dari versi sebelumnya.
10. Android versi 4.4 Kitkat, merupakan versi terbaru dari Android yang didukung dengan peningkatan kecepatan, dan responsif, perbaikan desain UI, penggunaan memory yang sedikit, peningkatan kerja NFC, *printing framework*, sensor dengan daya sedikit.

Beberapa keunggulan *Platform* android adalah sebagai berikut :

1. Lengkap (*Complete Platform*). Para desainer dapat melakukan pendekatan yang komprehensif ketika sedang mengembangkan

platform android. Android menyediakan banyak *tools* dalam membangun *software* dan merupakan sistem operasi yang aman.

2. Terbuka (*Open Source Platform*). Platform Android disediakan melalui lisensi *open source*.
3. Bebas (*Free Platform*). Android merupakan *platform* atau aplikasi yang bebas untuk dikembangkan. Tidak ada lisensi atau biaya royalti untuk dikembangkan pada *platform Android*.

2.5 Algoritma A*

2.5.1 Istilah – Istilah dalam A*

Dalam algoritma A*, terdapat istilah-istilah yang umum digunakan (Matthews, 2004) :

- *map* atau *graph*, merupakan ruang dalam algoritma A* yang digunakan untuk menemukan jalur antar dua tempat. *Map* bukan berarti peta dalam arti harfiahnya. Namun, *map* berbentuk kotak segi enam, segi tiga, segi empat, atau bahkan ruang dalam *game trees*. *Map* merupakan sebuah area dimana algoritma A* bekerja.
- *Nodes*, struktur-struktur yang merepresentasikan tempat-tempat dalam *map*. Tetapi, *map* merupakan struktur data sendiri dari *node*. *Node* menyimpan informasi untuk algoritma A* sebagaimana tempatnya. *Node* digunakan sebagai alat pencatat untuk menyimpan proses pencarian penentuan jalur (*pathfinding*).
- *Distances (heuristic)*, digunakan untuk menentukan baik tidaknya node yang akan diperiksa.

- *Cost* (biaya), biaya yang terdapat dalam jalur node awal sampai node n yang menjadi komponen algoritma A* untuk mendapatkan biaya minimum.

2.5.2 Algoritma A*

Menurut Suyanto (2011), Algoritma A* merupakan algoritma *Best First Search* yang menggabungkan *Uniform Cost Search* dan *Greedy Best-First Search*. Biaya yang diperhitungkan didapat dari biaya sebenarnya ditambah dengan biaya perkiraan. Dalam notasi matematika dituliskan sebagai : $f(n) = g(n) + h(n)$. Dengan perhitungan biaya seperti ini, algoritma A* adalah *complete* dan optimal.

Algoritma ini memiliki dua senarai yaitu OPEN dan CLOSED dimana terdapat tiga kondisi bagi setiap suksesor yang dibangkitkan (Suyanto, 2011:31), yaitu :

1. Sudah berada di *OPEN*

Apabila suksesor sudah pernah berada di node *OPEN*, dilakukan pengecekan apakah perlu pengubahan *parent* atau tidak tergantung pada nilai g -nya melalui *parent* lama atau *parent* baru. Jika melalui *parent* baru nilai g lebih kecil, maka dilakukan pengubahan *parent*. Jika pengubahan *parent* dilakukan, maka dilakukan pula perbaruan nilai g dan f pada suksesor tersebut. Dengan perubahan ini, suksesor tersebut memiliki kesempatan yang lebih besar untuk terpilih sebagai simpul terbaik (*best node*).

2. Sudah berada di *CLOSED*

Apabila suksesor sudah pernah berada di node *CLOSED*, dilakukan pengecekan apakah perlu perubahan *parent* atau tidak. Jika ya, maka dilakukan perbaruan nilai g dan f pada suksesor tersebut serta pada semua “anak cucunya” yang sudah pernah berada di *OPEN*. Dengan perbaruan ini, maka semua anak cucunya tersebut memiliki kesempatan lebih besar untuk terpilih sebagai simpul terbaik (*best node*).

3. Tidak berada di *OPEN* maupun *CLOSED*

Apabila suksesor tidak pernah berada di *OPEN* maupun *CLOSED*, maka suksesor tersebut dimasukkan ke dalam *OPEN*. Tambahkan suksesor tersebut sebagai suksesornya *best node*. Hitung biaya suksesor tersebut dengan rumus $f = g + h$.

Adapun maksud dari rumus tersebut adalah sebagai berikut

(Matthews, 2004):

g = biaya yang diperoleh dari node awal ke node n

h = biaya perkiraan yang diperoleh dari node n ke *goal* (tujuan).

f = penambahan dari nilai g dan h yang mempresentasikan total biaya dalam penentuan jalur. Nilai f terkecil merupakan nilai terbaik dalam pencapaian hasil akhir (*goal*).

Algoritma A^* termasuk *heuristic search*, fungsi *heuristic* memainkan peranan yang sangat menentukan. Suatu fungsi dapat diterima sebagai fungsi heuristik jika biaya perkiraan yang dihasilkan

tidak melebihi dari biaya sebenarnya. Ketika fungsi heuristik memberikan biaya perkiraan yang melebihi biaya sebenarnya, maka proses pencarian tersesat dan membuat *heuristic* menjadi tidak optimal. Suatu fungsi heuristik dikatakan baik jika bisa memberikan biaya perkiraan yang mendekati biaya sebenarnya.

Dalam menentukan nilai heuristik untuk persoalan *map* atau *grid*, dapat menggunakan fungsi heuristik, diantaranya (Amit, 2012) :

- Pada *grid* kotak persegi yang memungkinkan 4 arah gerakan, dapat menggunakan fungsi heuristik *Manhattan Distance*
- Pada *grid* kotak persegi yang memungkinkan 8 arah gerakan, dapat menggunakan fungsi heuristik *diagonal distance*
- Pada *grid* kotak persegi yang memungkinkan ke setiap arah gerakan, dapat menggunakan fungsi heuristik *Euclidean Distance*
- Pada *grid* segi enam yang memungkinkan 6 arah gerakan, dapat menggunakan fungsi heuristik *manhattan distance* yang sudah disesuaikan dengan *grid* segi enam

2.5.3 Iterative Deepening A* (IDA*)

Iterative Deepening A* merupakan pendekatan yang banyak digunakan dalam algoritma kecerdasan buatan yang dimulai dengan jawaban perkiraan, kemudian membuatnya menjadi akurat. Kata “*Depth*” merupakan *cutoff* dari nilai *f*. ketika nilai *f* terlalu besar, node tidak akan diambil (misalnya, tidak akan ditambahkan dalam OPEN).

Algoritma IDA* merupakan salah satu variasi dari algoritma A*. Menurut Suyanto (2011), Algoritma IDA* memiliki batasan yang berupa *f-limit* (nilai gabungan antara biaya sebenarnya dan biaya perkiraan). Pada algoritma ini, setiap iterasi akan mengembalikan nilai *f-limit* baru yang akan digunakan sebagai batasan pencarian untuk iterasi berikutnya.

Algoritma IDA* juga *complete* dan optimal. Tetapi karena dilakukan secara iteratif, maka IDA* mungkin membangkitkan simpul-simpul yang sama berulang-ulang. Hal ini membutuhkan waktu yang lama. Adapun keuntungan dari algoritma ini adalah jumlah memori yang dibutuhkan menjadi jauh lebih sedikit. Jika menghadapi masalah keterbatasan memori, maka algoritma ini bisa menjadi pilihan yang tepat (Suyanto, 2011). Hal ini dikarenakan karena algoritma IDA* melakukan serangkaian pencarian independen *depth-first* yang dibatasi oleh nilai *f* (Kurkovsky, 2009).

Penerapan algoritma IDA* dapat menggunakan fungsi heuristik (*h*)

Diagonal Distance : (Yofid I, 2011)

$$\text{num_diagonal_steps} = -\text{abs}(nX - \text{goalX}) + (nX - \text{goalY})$$

$$\text{num_straight_steps} = \text{abs}(nX - \text{goalX}) + (nY - \text{goalY})$$

$$- 2 * \text{num_diagonal_steps}$$

$$h(n) = \text{num_straight_steps} + \sqrt{2} * \text{num_diagonal_steps}$$

sedangkan untuk menghitung nilai *g* menggunakan rumus :

$$g(n) = \text{abs}(\text{startX} - nX) + \text{abs}(\text{startY} - nY)$$

sehingga,

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

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Game Sejenis

3.1.1 Game Super Mario Bros

A. Pengenalan

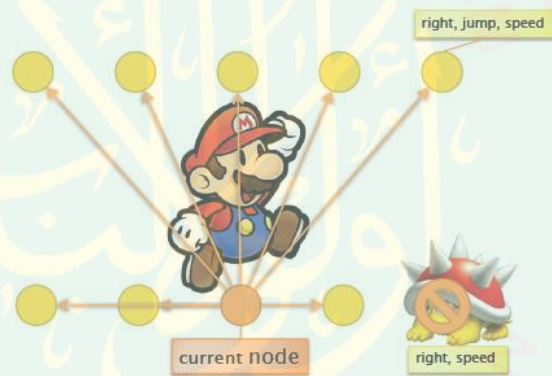
Game super Mario Bros merupakan game bergenre petualangan 2 dimensi. Tujuan permainan adalah menjelajahi kerajaan jamur, menghancurkan dan melawan prajurit *Bowser*. Musuh-musuh dalam permainan ini dihancurkan dengan menginjak sambil melompat. Mario akan mendapat koin dan bonus lain dengan balok bertanda tanya. Untuk menambah tenaga mario, pemain harus mendapatkan jamur merah. Kekuatan Mario akan bertambah dan mampu melontarkan proyektil bila memperoleh bunga berapi.

B. Game Modifikasi Super Mario Bros

Game ini dikembangkan oleh Robin Baumgarten dari Imperial Collage, London. Hal yang dikembangkan adalah dengan ditambahkan algoritma A* untuk menemukan jalur lompatan yang optimal agar terhindar dari hadangan musuh. Penerapan algoritma A* diimplementasikan pada karakter mario sebagai *agent* (Julian Togelius, 2009). Karakter utama pada game modifikasi ini berjalan sendiri tanpa adanya campur tangan *user* (pengguna).



Gambar 3.1 *Interface game super mario bros* Robin Baumgarten



Gambar 3.2 A* dalam *game super mario bros*

3.1.2 *Game Santri Story*

Game Santri Story dibuat untuk mengenalkan huruf hijaiyah kepada anak-anak. *Game* ini menceritakan perjalanan santri menuju masjid. Santri diharuskan mengumpulkan huruf hijaiyah yang tersebar di area permainan. Terdapat musuh berbentuk tuyul yang akan menghadang Santri. Kecerdasan buatan diterapkan dalam proses pengacakan letak posisi huruf dan musuh. Karakter musuh belum memiliki kecerdasan buatan sehingga untuk mencapai misi mudah dicapai.



Gambar 3.3 Antarmuka depan *game* santri story



Gambar 3.4 Antarmuka *game* santry story

3.1.3 *Game Side Scrolling 2d* Petualangan Kabayan

Game ini bercerita tentang petualangan seorang pemuda bernama Kabayan sebagai karakter utama untuk merebut kembali makanan tradisional dipulau Jawa yang telah dicuri oleh perampok bernama Rewog (Hidayat, 2013). *Game* ini berbasis *desktop* dengan menggunakan *game engine* *Greenfoot*. *Game* ini menggunakan algoritma A* yang diterapkan pada karakter musuh. Algoritma yang diterapkan dapat digunakan, namun membutuhkan memori yang besar. Hal ini dapat diatasi karena berbasis *desktop* yang sudah memiliki kebutuhan *hardware* yang cukup tinggi.

Komponen karakter dan item-item pada *game* masih sedikit dan pengenalan makanan khas masih terbatas di pulau Jawa saja.



Gambar 3.5 Antarmuka depan *game* petualangan kabayan



Gambar 3.6 Antarmuka *game* petualangan Kabayan ketika bermain

3.2 Analisis Game Pengenalan Tajwid

3.2.1 Keterangan Umum

Game yang dibangun adalah *game* edukasi ber-genre *side scrolling* yang mengaplikasikan perkembangan teknologi *mobile* sebagai alat untuk mengenalkan bagian ilmu tajwid melalui media *visual* dan *audio* yang

ditanamkan dalam *game*. *Game* ditujukan untuk pengguna minimal berusia 10 tahun. *Game* ini menerapkan sistem *single player* dan dibangun dengan grafis dua dimensi. Adapun kelebihan yang dimiliki :

1. Memberikan pengenalan sekaligus edukasi tentang bagian ilmu tajwid tepatnya hukum nun sukun dan tanwin
2. *Game* yang dibangun lebih bersifat bermain sambil belajar

Pada permulaan permainan, pemain dapat melakukan pengaturan terlebih dahulu di menu *setting* atau langsung bermain. Menu *setting* terdapat pengaturan tingkat kesulitan dalam bermain, yakni : mudah, sedang, dan sulit. Semakin sulit level, maka akan misi untuk berhasil juga akan sulit. Secara *default*, pengaturan tingkat kesulitan bermain adalah mudah. Selanjutnya, pemain memulai permainan dengan pemilihan level. Terdapat lima level dalam permainan yaitu level pertama dengan materi tajwid idgham bighunnah, level kedua dengan materi tajwid *idgham bilaghunnah*, level ketiga dengan materi tajwid *iqlab*, level keempat dengan materi tajwid *izh-har*, dan level lima dengan materi tajwid *ikhfa'*.

Permainan dilanjutkan dengan petunjuk misi yang harus dilakukan, Pemain berpetualang untuk mengumpulkan item huruf-huruf bacaan tajwid dan item pertanyaan sesuai dengan materi tajwid yang tersebar di area permainan.

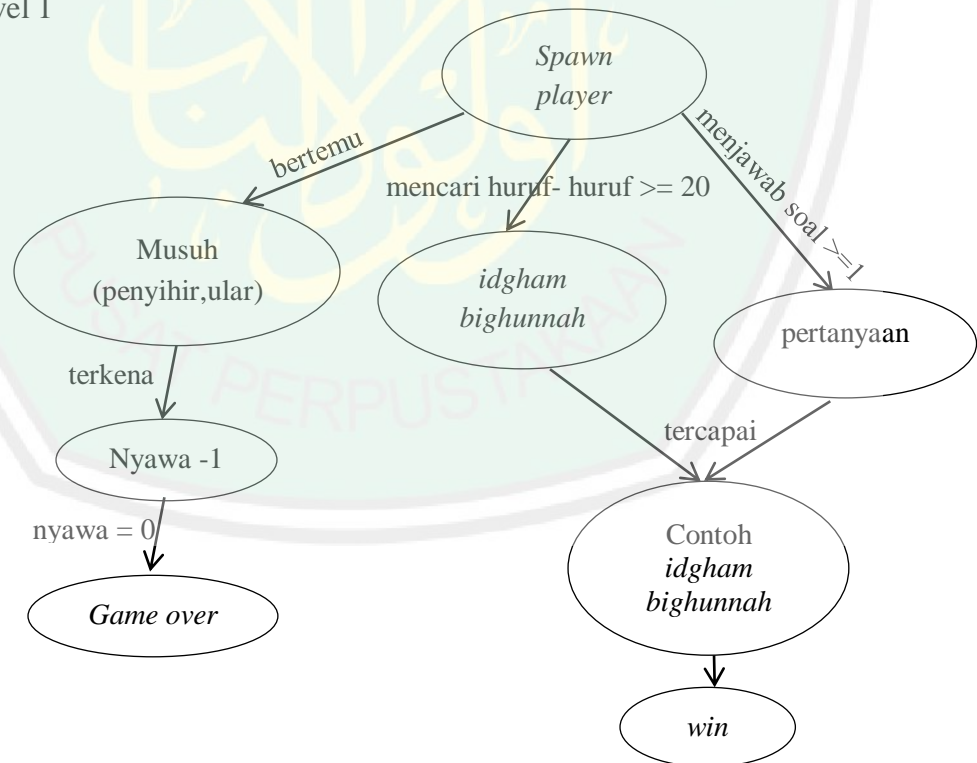
Objek dalam penelitian ini, yaitu algoritma *iterative deepening A** (IDA*) akan diterapkan dalam pergerakan musuh. Musuh yang tersebar

di area permainan diterapkan algoritma tersebut untuk menentukan jalur terpendek musuh dalam mengejar pemain. Selain itu, setiap permainan posisi huruf-huruf tersebut dilakukan secara acak agar pemain tidak merasa bosan dan tidak mudah diketahui.

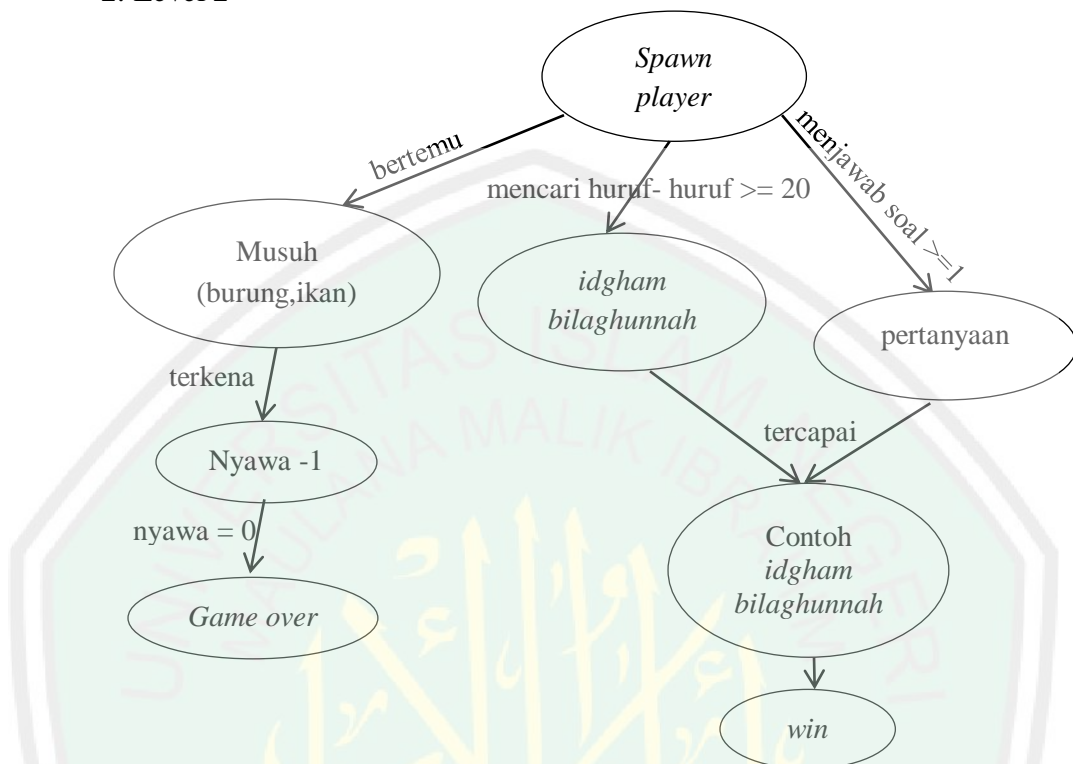
Pemain akan mendapatkan contoh-contoh bacaan tajwid jika misi berhasil dan dapat melihat contoh-contoh bacaan yang didapatkan di menu koleksi. Pada menu tersebut, contoh bacaan akan ditemukan penandaan warna yang berbeda sebagai tanda adanya bacaan tajwid yang akan mengeluarkan suara jika disentuh.

3.2.2 FSM (*Finite State Machine*)

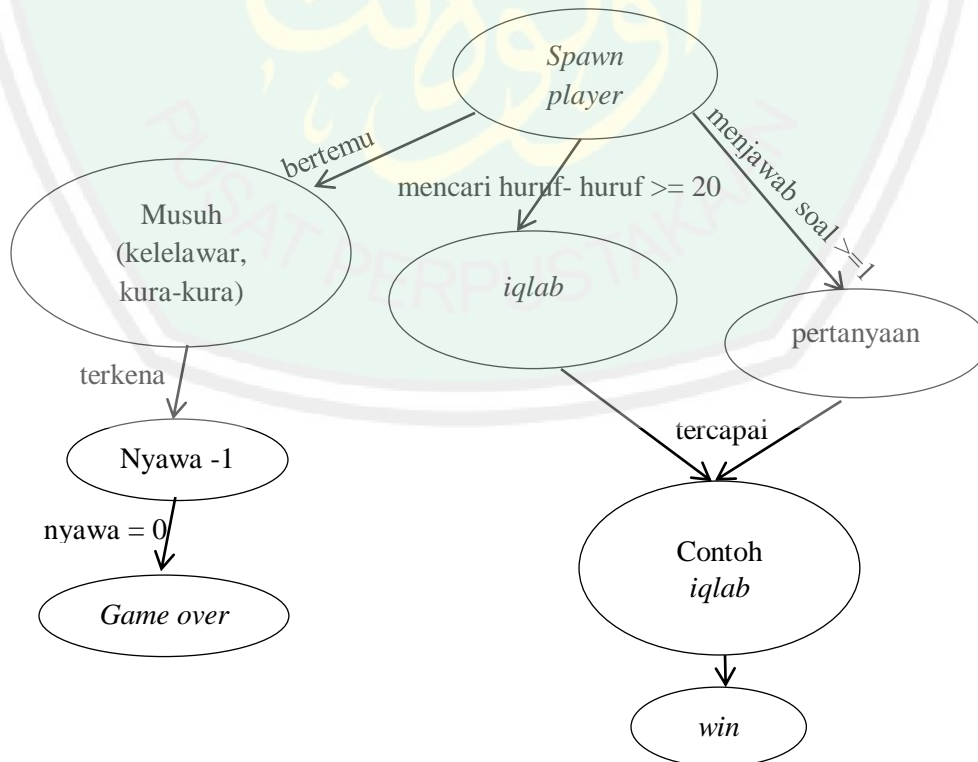
1. Level 1



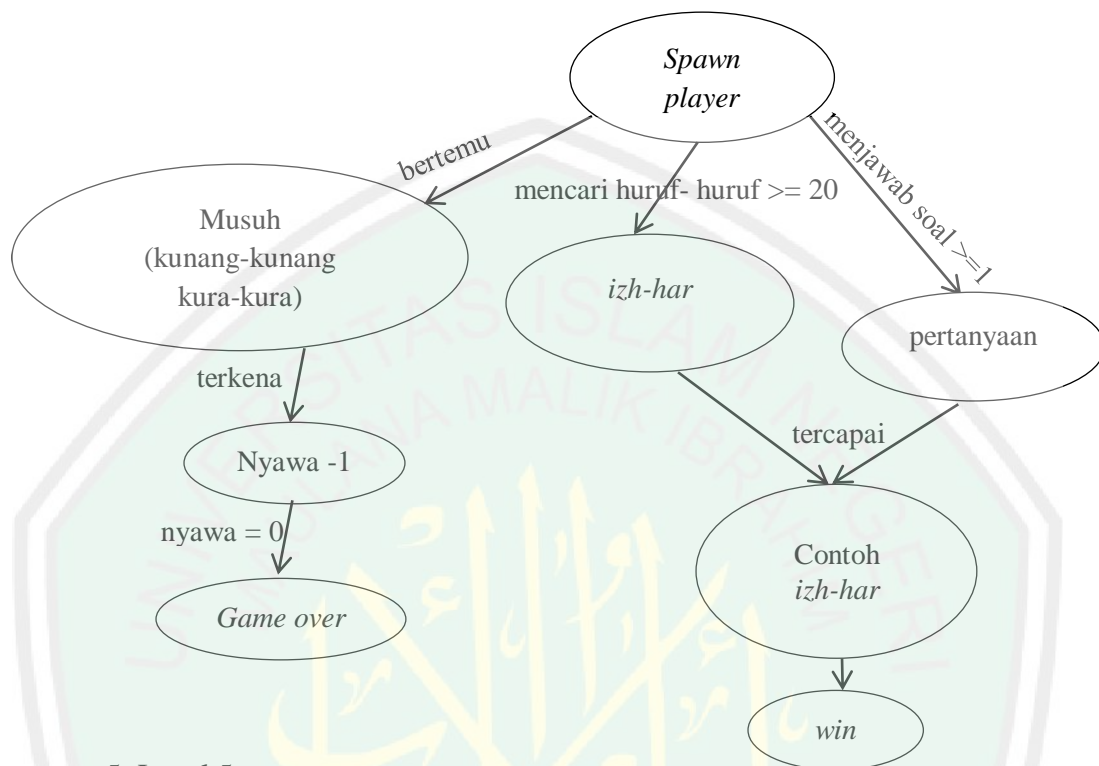
2. Level 2



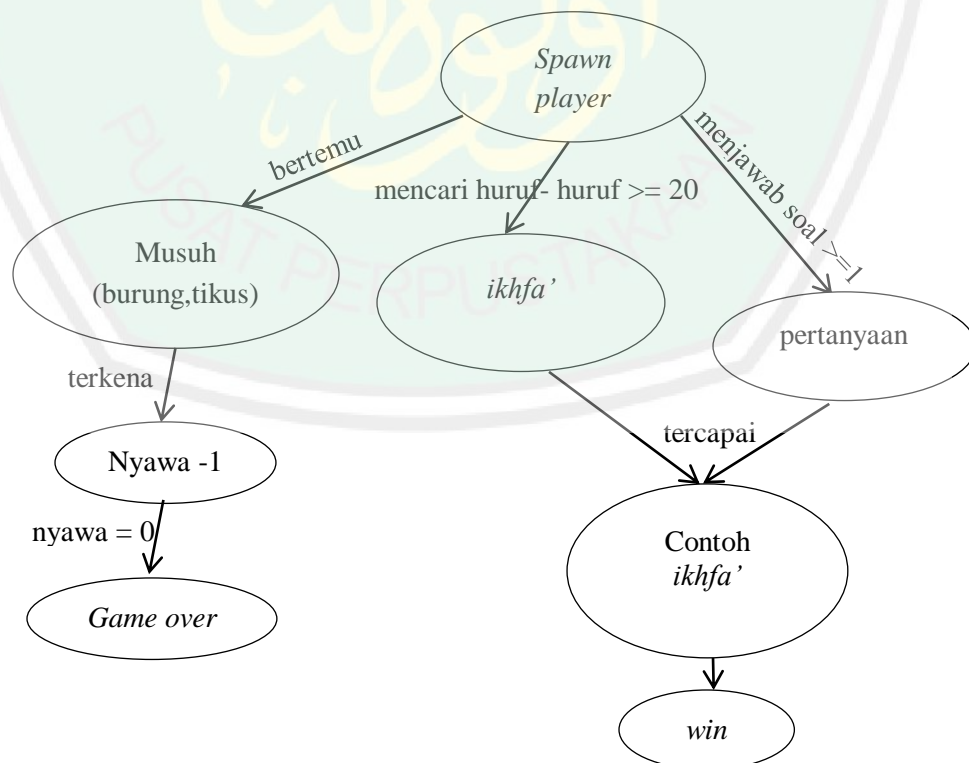
3. Level 3



4. Level 4



5. Level 5



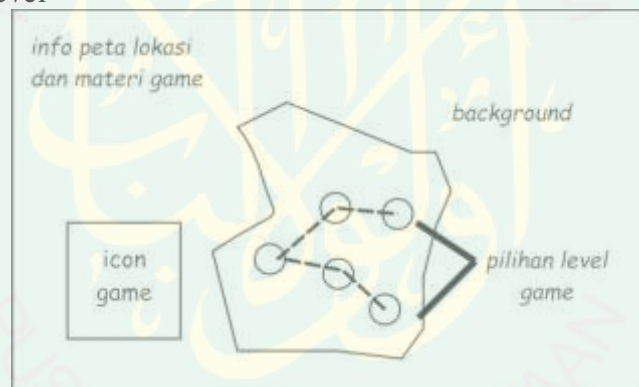
3.2.3 Storyboard Game

1. Menu Utama



Gambar 3.7 Desain menu utama *game* tajwid

2. Level



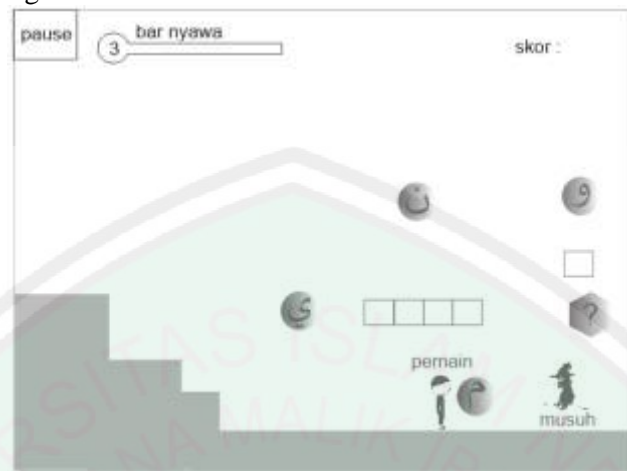
Gambar 3.8 Desain untuk memilih level permainan

4. Misi



Gambar 3.9 Desain misi pada setiap level

5. Mengambil item huruf



Gambar 3.10 Desain pemain ketika mengambil item huruf

6. Mengambil item pertanyaan



Gambar 3.11 Desain pemain ketika mengambil item pertanyaan

PERTANYAAN

Pilihan Jawaban A

Pilihan Jawaban B

Pilihan Jawaban C

7. Misi sukses



Gambar 3.1 Desain misi sukses

8. Misi gagal



Gambar 3.2 Desain misi gagal

3.2.4 Storyline

Game ini menceritakan seorang anak bernama Zaid yang berpetualang menelusuri tempat-tempat makam para sunan yang ada di Jawa Timur, yakni : makam Syekh Maulana Malik Ibrahim, Sunan Giri, Sunan Ampel, Sunan Bonang, dan Sunan Drajad. *Game* ini memiliki jumlah level sebanyak lima. Setiap level memiliki misi yang harus

diselesaikan agar dapat mengunjungi makam-makam tersebut. Untuk mencapai misi tersebut, Zaid akan menghadapi musuh yang telah ditanamkan kecerdasan buatan sehingga menambah sulitnya *game* ini. *Game* ini memiliki tiga tingkat kesulitan yakni mudah, sedang, dan sulit.

3.2.5 Gameplay

Pemain berusaha mencari item yang didalamnya terdapat huruf-huruf bacaan tajwid dan item permata bersimbol “?” sesuai level yang tersebar di area permainan. Letak posisi huruf dan item permata dilakukan secara acak. Setiap level berisi huruf-huruf yang berbeda sesuai dengan hukum bacaan sebagai berikut :

1. Level pertama : materi tajwid idgham bighunnah
2. Level kedua : materi tajwid idgham bilaghunnah
3. Level ketiga : materi tajwid iqlab
4. Level keempat : materi tajwid idzhar
5. Level kelima : materi tajwid ikhfa’

Dalam mencari item-item tersebut, pemain harus menghadapi banyak musuh sesuai dengan tingkat kesulitannya. Pemain dapat berjalan ke kiri, ke kanan, dan melompat. Dalam hal kemampuan, pemain dapat menendang dan menembak. Sedangkan musuh dapat bergerak ke kiri dan ke kanan. Terdapat musuh tertentu yang dapat bergerak secara diagonal dimana diterapkan algoritma IDA*. Adapun variasi kemampuan musuh dapat memukul, dan menggigit. Pemain dibekali dengan *health*

(kesehatan) sebanyak 200, *health* ini akan berkurang jika terkena musuh. Contoh-contoh bacaan akan didapatkan jika jumlah huruf mencapai batas minimal yang dapat dilihat di petunjuk sebelum bermain. Contoh bacaan tersebut akan disimpan dalam menu koleksi.

3.2.6 Scoring

Dalam hal *scoring*, pemain mendapat skor bila mengumpulkan item huruf-huruf bacaan tajwid dengan skor satu dan permata yang berisi pertanyaan dengan skor satu jika benar.

3.2.7 Tingkat Kesulitan

Dalam *game* ini dibagi menjadi tiga tingkatan yaitu mudah, sedang, dan sulit. Semakin tinggi tingkatan, maka semakin sulit pula untuk mencapai misi. Tingkat kesulitan didasarkan pada kecepatan gerak musuh, *health* musuh dan pengumpulan huruf tajwid. Berikut deskripsi secara rinci mengenai tingkatan kesulitan dalam *game*.

A. Tingkat mudah

- Kecepatan gerak musuh normal
- *Health* (kesehatan) musuh sebanyak 1
- Huruf tajwid sesuai level

B. Tingkat sedang

- Kecepatan gerak musuh ditambah 0.25 dari kecepatan normal
- *Health* (kesehatan) musuh sebanyak 2
- Huruf tajwid setiap level, diacak dari sebagian ilmu tajwid

C. Tingkat sulit

- Kecepatan gerak musuh ditambah 0.5 dari kecepatan normal
- *Health* (kesehatan) musuh sebanyak 3
- Huruf tajwid setiap level, diacak dari seluruh materi ilmu tajwid

3.2.8 Konten-konten Game

Konten-konten aplikasi yang ada dalam *game* adalah :

1. Materi Tajwid

Adapun materi tajwid yang digunakan adalah hukum nun sukun dan tanwin yang ada di tiap-tiap level.

2. Latar

Pada setiap level memiliki latar yang berbeda-beda.

Level pertama : latar tempat di Kota Gresik

Level kedua : latar tempat di Kota Gresik

Level ketiga : latar tempat di Kota Surabaya

Level keempat : latar tempat di Kota Lamongan

Level kelima : latar tempat di Kota Tuban

3. Tokoh utama (pemain)

Pemain berpetualang menelusuri tempat-tempat makam para sunan yang ada di Jawa Timur dengan tantangan mengumpulkan item-item yang didalamnya terdapat huruf-huruf bacaan tajwid dan permata untuk menjawab pertanyaan dengan benar sebagai syarat untuk lanjut ke level selanjutnya.

4. Musuh

Setiap level terdapat beberapa musuh yang harus dikalahkan.

Adapun musuh adalah sebagai berikut :

Tabel 3.1 Deskripsi musuh dalam *game*

No.	Nama	Keahlian	Level
1.	Penyihir	terbang, memukul	level 1
2.	Ikan	Menggigit	level 2
3.	Burung	Mematuk	level 2 dan 5
4.	kura-kura	Menggigit	level 3
5.	Tikus	Menggigit	level 5
6.	Ular	Menjulurkan lidahnya	level 1, dan 4
7.	Kunang-kunang	Menggigit	level 4
8.	Kelelawar	terbang, menggigit	Level 3

3.2.9 Skenario Game

Skenario *game* dalam aplikasi ini sebagai berikut :

Skenario <i>Game</i>
1. Pemain dapat bergerak ke kanan, ke kiri, melompat di area permainan
2. Pemain berusaha mengumpulkan item yang didalamnya terdapat huruf-huruf bacaan tajwid dan item permata yang didalamnya terdapat pertanyaan
3. Pemain berusaha melawan atau menghindari dari musuh di area permainan
4. Pemain berusaha mendapatkan contoh-contoh bacaan tajwid dengan mengumpulkan huruf dan menjawab item pertanyaan sesuai level

3.3 Perancangan algoritma Iterative Deepening A* pada musuh

Penerapan algoritma terdapat pada musuh. Musuh dapat bergerak ke delapan arah (kiri,kanan,atas,bawah, dan empat arah diagonal).

A. Implementasi Algoritma Iterative Deepening A*

Algoritma Iterative Deepening A* diimplementasikan dengan alur sebagai berikut :

1. Melakukan permainan
2. Inisialisasi konten-konten yang ada dalam *game*
3. Menempatkan karakter musuh untuk jadi simpul awal
4. Menghitung simpul terbaik musuh ke simpul player
5. Menentukan jalur terpendek menggunakan algoritma IDA*
6. Jika ditemukan solusi, maka musuh bergerak menuju player
7. Jika sudah mencapai player, musuh memukul

Simpul awal berisi nilai koordinat posisi x dan y musuh. Simpul yang dihitung adalah nilai g (biaya yang diperoleh dari node awal ke node n) dan h (biaya perkiraan yang diperoleh dari node n ke tujuan) yang dijumlahkan. Hasil penjumlahan tersebut disebut nilai f . Adapun rumus untuk mencari nilai h adalah :

$$\begin{aligned} \text{num_diagonal_steps} &= -\text{abs}(nX - \text{goalX}) + (nY - \text{goalY}) \\ \text{num_straight_steps} &= \text{abs}(nX - \text{goalX}) + (nY - \text{goalY}) \\ &- 2 * \text{num_diagonal_steps} \\ h(n) &= \text{num_straight_steps} + \sqrt{2} * \text{num_diagonal_steps} \end{aligned}$$

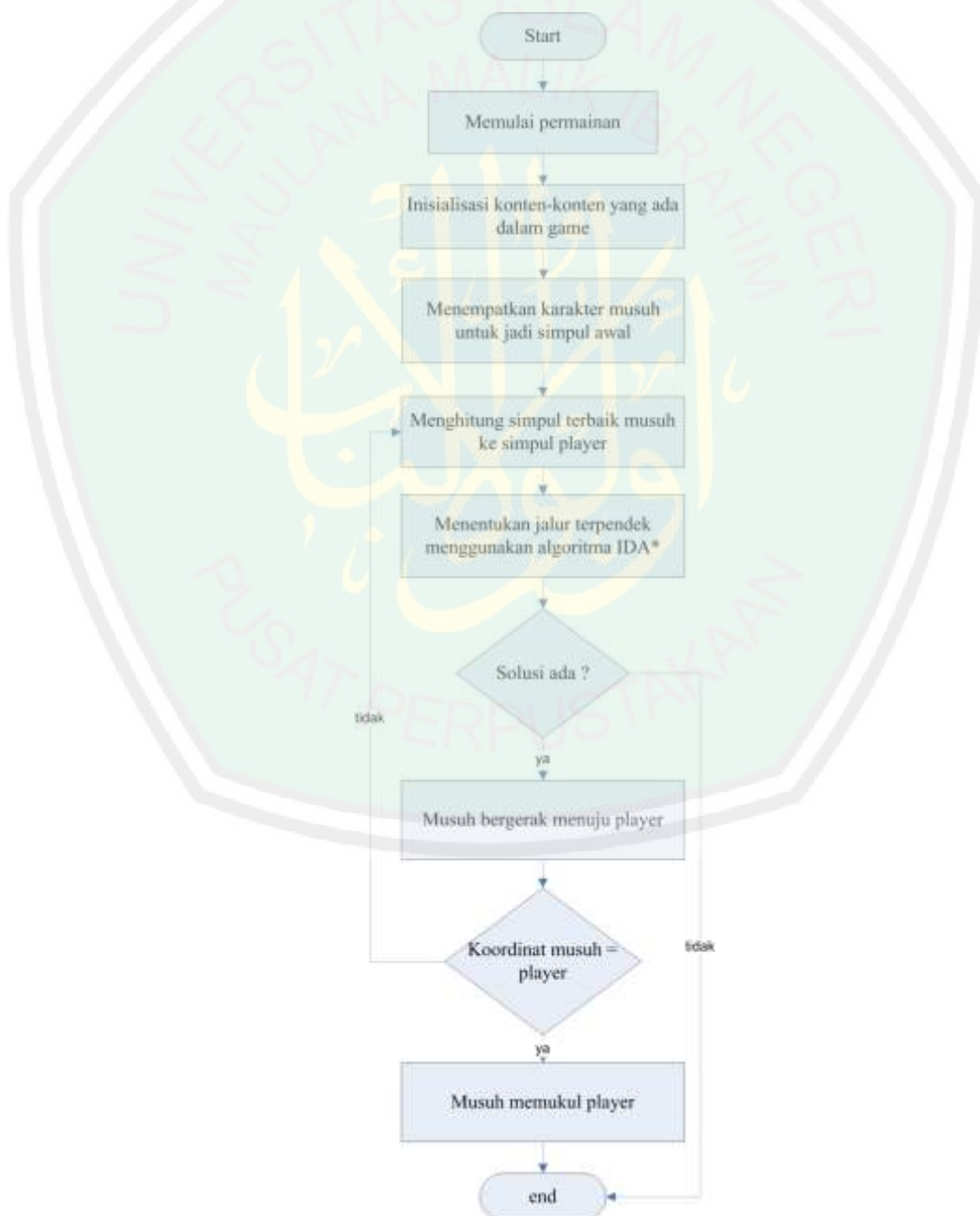
sedangkan untuk menghitung nilai g menggunakan rumus :

$$g(n) = \text{abs}(\text{startX} - nX) + \text{abs}(\text{startY} - nY)$$

sehingga,

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

Hasil perhitungan tersebut akan diambil nilai terendah dari f untuk disimpan menjadi simpul terbaik (*best node*) disetiap iterasi.



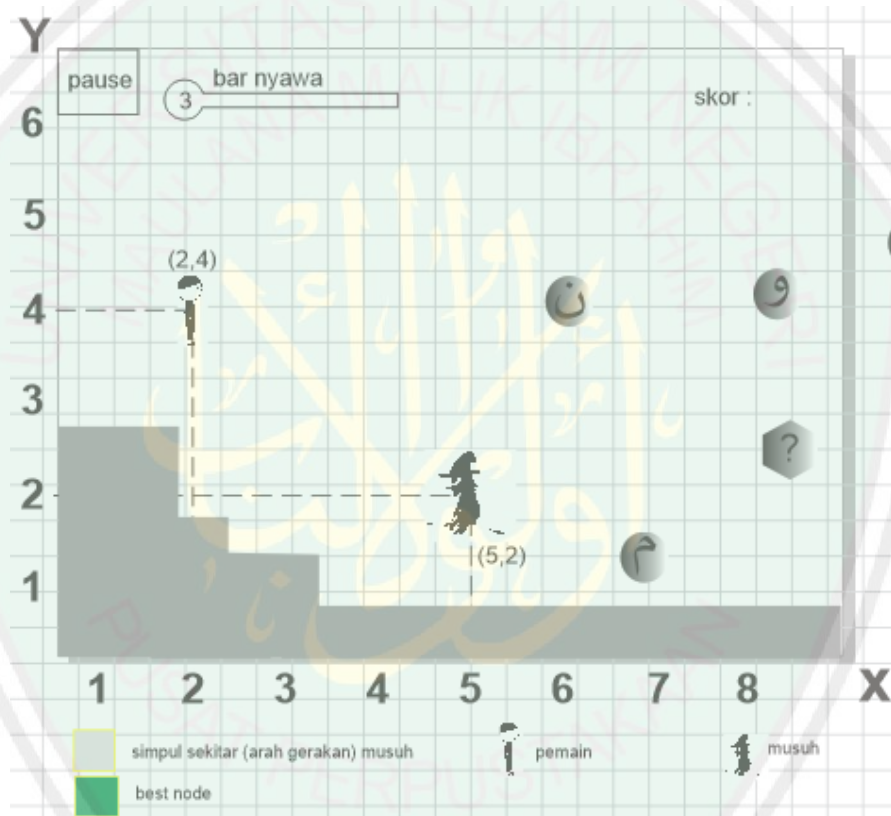
Gambar 3.3 Blok diagram implementasi IDA*

B. Simulasi manual perhitungan algoritma Iterative Deepening A*

(1) Kondisi awal : pemain (2,4), musuh (5,2)

Masukkan koordinat musuh ke OPEN

Inisialisasi $f\text{-limit} = \text{MAX_VALUE}$, $f\text{-Cost} = \text{MAX_VALUE}$, $\text{next-f} = f\text{-cost}$



Gambar 3.4 Tampilan awal koordinat pemain dan musuh

(2) Hitung semua *node* terdekat yang mungkin dilewati, abaikan penghalang

Langkah 1 :

Gerak kiri (4,2)

$$\text{diagonal} = -(\text{abs}(4 - 2) + \text{abs}(2 - 4)) = -(2+2) = -4$$

$$\text{lurus} = (\text{abs}(4 - 2) + \text{abs}(2 - 4)) - (2 \times -4)$$

$$= 4 - (-8) = 12$$

$$\begin{aligned} h(n) &= \text{lurus} + \sqrt{2} * \text{diagonal} \\ &= 12 + 1.4(-4) = 6.4 \end{aligned}$$

$$\begin{aligned} g(1) &= (\text{abs}(5+(-1))) + \text{abs}(2-0) \\ &= 4 + 2 = 6 \end{aligned}$$

$$\begin{aligned} f(1) &= h(n) + g(n) \\ &= 6.4 + 6 = 12.4 \end{aligned}$$

$$\begin{aligned} \text{next-}f &= \min(\text{next-}f, f(1)) \\ &= \min(\text{MAX_VALUE}, 12.4) \\ &= 12.4 \end{aligned}$$

Masukkan nilai $f(1)$ gerak kiri(4,2) ke dalam OPEN. Lakukan pengecekan, apakah gerak kiri(4,2) ada dalam CLOSE? Jika tidak, masukkan ke CLOSE. Jika iya, lakukan pengecekan apakah nilai g saat ini lebih kecil dari g sebelumnya? Jika iya, lakukan penggantian node dalam CLOSE. Jika tidak, abaikan. Hapus gerak kiri(4,2) dalam OPEN. Hapus nilai $f(1)$.

Gerak kanan (6,2)

$$\text{diagonal} = -(\text{abs}(6 - 2) + \text{abs}(2 - 4)) = -(4+2) = -6$$

$$\begin{aligned} \text{lurus} &= (\text{abs}(6 - 2) + \text{abs}(2 - 4)) - (2*-6) \\ &= 6 - (-12) = 18 \end{aligned}$$

$$\begin{aligned} h(n) &= \text{lurus} + \sqrt{2} * \text{diagonal} \\ &= 18 + 1.4(-6) = 9.6 \end{aligned}$$

$$\begin{aligned} g(1) &= (\text{abs}(5+(1))) + \text{abs}(2+0) \\ &= 6 + 2 = 8 \end{aligned}$$

$$\begin{aligned} f(1) &= h(n) + g(n) \\ &= 9.6 + 8 = 17.6 \end{aligned}$$

$$\begin{aligned} \text{next-}f &= \min(\text{next-}f, f(1)) \\ &= \min(12.4, 17.6) \\ &= 12.4 \end{aligned}$$

Masukkan nilai $f(1)$ gerak kanan(6,2) ke dalam OPEN. Lakukan pengecekan, apakah gerak kanan(6,2) ada dalam CLOSE? Jika tidak, masukkan ke CLOSE. Jika iya, lakukan pengecekan apakah nilai g saat ini lebih kecil dari g sebelumnya? Jika iya, lakukan penggantian node dalam CLOSE. Jika tidak, abaikan. Hapus gerak kanan(6,2) dalam OPEN. Hapus nilai $f(1)$.

Gerak atas (5,3)

$$\text{diagonal} = -(\text{abs}(5 - 2) + \text{abs}(3 - 4)) = -(3+1) = -4$$

$$\begin{aligned} \text{lurus} &= (\text{abs}(5 - 2) + \text{abs}(3 - 4)) - (2 * -4) \\ &= 4 - (-8) = 12 \end{aligned}$$

$$\begin{aligned} h(n) &= \text{lurus} + \sqrt{2} * \text{diagonal} \\ &= 12 + 1.4(-4) = 6.4 \end{aligned}$$

$$\begin{aligned} g(1) &= (\text{abs}(5+(0)) + \text{abs}(2+1)) \\ &= 5 + 3 = 8 \end{aligned}$$

$$\begin{aligned} f(1) &= h(n) + g(n) \\ &= 6.4 + 8 = 14.4 \end{aligned}$$

$$\begin{aligned} \text{next-}f &= \min(\text{next-}f, f(1)) \\ &= \min(12.4, 14.4) \\ &= 12.4 \end{aligned}$$

Masukkan nilai $f(1)$ gerak atas(5,3) ke dalam OPEN. Lakukan pengecekan, apakah gerak atas(5,3) ada dalam CLOSE? Jika tidak, masukkan ke CLOSE. Jika iya, lakukan pengecekan apakah nilai g saat ini lebih kecil dari g sebelumnya? Jika iya, lakukan penggantian node dalam CLOSE. Jika tidak, abaikan. Hapus gerak atas(5,3) dalam OPEN. Hapus nilai $f(1)$.

Gerak diagonal kiri atas (4,3)

$$\text{diagonal} = -(\text{abs}(4 - 2) + \text{abs}(3 - 4)) = -(2+1) = -3$$

$$\begin{aligned} \text{lurus} &= (\text{abs}(4 - 2) + \text{abs}(3 - 4)) - (2 * -3) \\ &= 3 - (-6) = 9 \end{aligned}$$

$$\begin{aligned} h(n) &= \text{lurus} + \sqrt{2} * \text{diagonal} \\ &= 9 + 1.4(-3) = 4.8 \end{aligned}$$

$$\begin{aligned} g(1) &= (\text{abs}(5+(-1)) + \text{abs}(2+1)) \\ &= 4 + 3 = 7 \end{aligned}$$

$$\begin{aligned} f(1) &= h(n) + g(n) \\ &= 4.8 + 7 = 11.8 \end{aligned}$$

$$\begin{aligned} \text{next-}f &= \min(\text{next-}f, f(1)) \\ &= \min(12.4, 11.8) \\ &= 11.8 \end{aligned}$$

Masukkan nilai $f(1)$ gerak diagonal kiri atas (4,3) ke dalam OPEN. Lakukan pengecekan, apakah gerak diagonal kiri atas (4,3) ada dalam CLOSE? Jika tidak, masukkan ke CLOSE. Jika iya, lakukan pengecekan apakah nilai g saat ini lebih kecil dari g sebelumnya? Jika iya, lakukan penggantian node dalam CLOSE. Jika tidak, abaikan. Hapus gerak diagonal kiri atas (4,3) dalam OPEN. Hapus nilai $f(1)$.

Gerak diagonal kanan atas (6,3)

$$\text{diagonal} = -(\text{abs}(6 - 2) + \text{abs}(3 - 4)) = -(4+1) = -5$$

$$\begin{aligned} \text{lurus} &= (\text{abs}(6 - 2) + \text{abs}(3 - 4)) - (2 * -5) \\ &= 5 - (-10) = 15 \end{aligned}$$

$$\begin{aligned} h(n) &= \text{lurus} + \sqrt{2} * \text{diagonal} \\ &= 15 + 1.4(-5) = 8 \end{aligned}$$

$$\begin{aligned} g(1) &= (\text{abs}(5+(1)) + \text{abs}(2+1)) \\ &= 6 + 3 = 9 \end{aligned}$$

$$\begin{aligned} f(1) &= h(n) + g(n) \\ &= 8 + 9 = 17 \end{aligned}$$

$$\text{next-}f = \min(\text{next-}f, f(1))$$

$$\begin{aligned}
 &= \min(11.8, 17) \\
 &= 11.8
 \end{aligned}$$

Masukkan nilai $f(1)$ gerak diagonal kanan atas (6,3) ke dalam OPEN. Lakukan pengecekan, apakah gerak kiri(4,2) ada dalam CLOSE? Jika tidak, masukkan ke CLOSE. Jika iya, lakukan pengecekan apakah nilai g saat ini lebih kecil dari g sebelumnya? Jika iya, lakukan penggantian node dalam CLOSE. Jika tidak, abaikan. Hapus gerak kiri(4,2) dalam OPEN. Hapus nilai $f(1)$.

Setelah semua node terdekat yang mungkin dibangkitkan, simpan nilai $next-f$ ke dalam $bestnode$. Berdasarkan perhitungan di atas, $next-f$ bernilai 11.8 yang berada pada koordinat (4,3) dengan arah diagonal kiri atas.

Langkah 2 : musuh di posisi(4,3), pemain (2,4)

$$f\text{-limit} = next\text{-}f$$

Gerak kiri (3,3)

$$diagonal = -(\text{abs}(3 - 2) + \text{abs}(3 - 4)) = -(1+1) = -2$$

$$\begin{aligned}
 lurus &= (\text{abs}(3 - 2) + \text{abs}(3 - 4)) - (2 * -2) \\
 &= 2 - (-4) = 6
 \end{aligned}$$

$$\begin{aligned}
 h(n) &= lurus + \sqrt{2} * diagonal \\
 &= 6 + 1.4(-2) = 3.2
 \end{aligned}$$

$$\begin{aligned}
 g(1) &= (\text{abs}(4+(-1)) + \text{abs}(3+0)) \\
 &= 3 + 3 = 6
 \end{aligned}$$

$$\begin{aligned}
 f(1) &= h(n) + g(n) \\
 &= 3.2 + 6 = 9.2
 \end{aligned}$$

$$\begin{aligned}
 next\text{-}f &= \min(next\text{-}f, f(1)) \\
 &= \min(11.8, 9.2) \\
 &= 9.2
 \end{aligned}$$

Masukkan nilai $f(2)$ gerak kiri (4,3) ke dalam OPEN. Lakukan pengecekan, apakah gerak kiri(4,3) ada dalam CLOSE? Jika tidak, masukkan ke CLOSE. Jika iya, lakukan pengecekan apakah nilai g saat ini lebih kecil dari g sebelumnya? Jika iya, lakukan penggantian node dalam CLOSE. Jika tidak, abaikan. Hapus gerak kiri(4,3) dalam OPEN. Hapus nilai $f(2)$.

Gerak diagonal kiri atas (3,4)

$$\text{diagonal} = -(\text{abs}(3 - 2) + \text{abs}(4 - 4)) = -(1+0) = -1$$

$$\begin{aligned} \text{lurus} &= (\text{abs}(3 - 2) + \text{abs}(4 - 4)) - (2 * -1) \\ &= 1 - (-2) = 3 \end{aligned}$$

$$\begin{aligned} h(n) &= \text{lurus} + \sqrt{2} * \text{diagonal} \\ &= 3 + 1.4(-1) = 1.6 \end{aligned}$$

$$\begin{aligned} g(1) &= (\text{abs}(4+(-1)) + \text{abs}(3+1)) \\ &= 3 + 4 = 7 \end{aligned}$$

$$\begin{aligned} f(1) &= h(n) + g(n) \\ &= 1.6 + 7 = 8.6 \end{aligned}$$

$$\begin{aligned} \text{next-}f &= \min(\text{next-}f, f(1)) \\ &= \min(9.2, 8.6) \\ &= 8.6 \end{aligned}$$

Masukkan nilai $f(2)$ gerak diagonal kiri atas (3,4) ke dalam OPEN. Lakukan pengecekan, apakah gerak diagonal kiri atas(3,4) ada dalam CLOSE? Jika tidak, masukkan ke CLOSE. Jika iya, lakukan pengecekan apakah nilai g saat ini lebih kecil dari g sebelumnya? Jika iya, lakukan penggantian node dalam CLOSE. Jika tidak, abaikan. Hapus gerak diagonal kiri atas (3,4) dalam OPEN. Hapus nilai $f(2)$.

Gerak diagonal kiri bawah (3,2)

$$\text{diagonal} = -(\text{abs}(3 - 2) + \text{abs}(2 - 4)) = -(1+2) = -3$$

$$\begin{aligned} \text{lurus} &= (\text{abs}(3 - 2) + \text{abs}(2 - 4)) - (2 * -3) \\ &= 3 - (-6) = 9 \end{aligned}$$

$$\begin{aligned} h(n) &= \text{lurus} + \sqrt{2} * \text{diagonal} \\ &= 9 + 1.4(-3) = 4.8 \end{aligned}$$

$$\begin{aligned} g(1) &= (\text{abs}(4+(-1)) + \text{abs}(3+(-1))) \\ &= 3 + 2 = 5 \end{aligned}$$

$$\begin{aligned} f(1) &= h(n) + g(n) \\ &= 4.8 + 5 = 9.8 \end{aligned}$$

$$\begin{aligned} \text{next-}f &= \min(\text{next-}f, f(1)) \\ &= \min(9.2, 8.6) \\ &= 8.6 \end{aligned}$$

Masukkan nilai $f(2)$ gerak diagonal kiri bawah (3,2) ke dalam OPEN. Lakukan pengecekan, apakah gerak diagonal kiri bawah(3,2) ada dalam CLOSE? Jika tidak, masukkan ke CLOSE. Jika iya, lakukan pengecekan apakah nilai g saat ini lebih kecil dari g sebelumnya? Jika iya, lakukan penggantian node dalam CLOSE. Jika tidak, abaikan. Hapus gerak diagonal kiri bawah (3,2) dalam OPEN. Hapus nilai $f(2)$.

Gerak atas (4,4)

$$\text{diagonal} = -(\text{abs}(4 - 2) + \text{abs}(4 - 4)) = -(2+0) = -2$$

$$\begin{aligned} \text{lurus} &= (\text{abs}(4 - 2) + \text{abs}(4 - 4)) - (2 * -2) \\ &= 2 - (-4) = 6 \end{aligned}$$

$$\begin{aligned} h(n) &= \text{lurus} + \sqrt{2} * \text{diagonal} \\ &= 6 + 1.4(-2) = 3.2 \end{aligned}$$

$$\begin{aligned} g(1) &= (\text{abs}(4+(0)) + \text{abs}(3+1)) \\ &= 4 + 4 = 8 \end{aligned}$$

$$\begin{aligned} f(1) &= h(n) + g(n) \\ &= 3.2 + 8 = 11.2 \end{aligned}$$

$$\text{next-}f = \min(\text{next-}f, f(1))$$

$$\begin{aligned}
 &= \min(8.6, 11.2) \\
 &= 8.6
 \end{aligned}$$

Masukkan nilai $f(2)$ gerak atas (4,4) ke dalam OPEN. Lakukan pengecekan, apakah gerak atas (4,4) ada dalam CLOSE? Jika tidak, masukkan ke CLOSE. Jika iya, lakukan pengecekan apakah nilai g saat ini lebih kecil dari g sebelumnya? Jika iya, lakukan penggantian node dalam CLOSE. Jika tidak, abaikan. Hapus gerak atas(4,4) dalam OPEN. Hapus nilai $f(2)$.

Gerak diagonal kanan atas (5,4)

$$\begin{aligned}
 \text{diagonal} &= -(\text{abs}(5 - 2) + \text{abs}(4 - 4)) = -(3+0) = -3 \\
 \text{lurus} &= (\text{abs}(5 - 2) + \text{abs}(4 - 4)) - (2 * -3) \\
 &= 3 - (-6) = 9 \\
 h(n) &= \text{lurus} + \sqrt{2} * \text{diagonal} \\
 &= 9 + 1.4(-3) = 4.8 \\
 g(1) &= (\text{abs}(4+(1)) + \text{abs}(3+1)) \\
 &= 5 + 4 = 9 \\
 f(1) &= h(n) + g(n) \\
 &= 4.8 + 9 = 13.8 \\
 \text{next-}f &= \min(\text{next-}f, f(1)) \\
 &= \min(8.6, 13.8) \\
 &= 8.6
 \end{aligned}$$

Masukkan nilai $f(2)$ gerak kanan atas (5,4) ke dalam OPEN. Lakukan pengecekan, apakah gerak kanan atas(5,4) ada dalam CLOSE? Jika tidak, masukkan ke CLOSE. Jika iya, lakukan pengecekan apakah nilai g saat ini lebih kecil dari g sebelumnya? Jika iya, lakukan penggantian node dalam CLOSE. Jika tidak, abaikan. Hapus gerak

kanan atas (5,4) dalam OPEN. Hapus nilai $f(2)$.

Gerak kanan (5,3)

$$\text{diagonal} = -(\text{abs}(5 - 2) + \text{abs}(3 - 4)) = -(3+1) = -4$$

$$\begin{aligned} \text{lurus} &= (\text{abs}(5 - 2) + \text{abs}(3 - 4)) - (2 * -4) \\ &= 4 - (-8) = 12 \end{aligned}$$

$$\begin{aligned} h(n) &= \text{lurus} + \sqrt{2} * \text{diagonal} \\ &= 12 + 1.4(-4) = 6.4 \end{aligned}$$

$$\begin{aligned} g(1) &= (\text{abs}(4+(1)) + \text{abs}(3+0)) \\ &= 5 + 3 = 8 \end{aligned}$$

$$\begin{aligned} f(1) &= h(n) + g(n) \\ &= 6.4 + 8 = 14.4 \end{aligned}$$

$$\begin{aligned} \text{next-}f &= \min(\text{next-}f, f(1)) \\ &= \min(8.6, 14.4) \\ &= 8.6 \end{aligned}$$

Masukkan nilai $f(2)$ gerak kanan (5,3) ke dalam OPEN. Lakukan pengecekan, apakah gerak kanan(5,3) ada dalam CLOSE? Jika tidak, masukkan ke CLOSE. Jika iya, lakukan pengecekan apakah nilai g saat ini lebih kecil dari g sebelumnya? Jika iya, lakukan penggantian node dalam CLOSE. Jika tidak, abaikan. Hapus gerak kanan(5,3) dalam OPEN. Hapus nilai $f(2)$.

Gerak diagonal kanan bawah (5,2)

$$\text{diagonal} = -(\text{abs}(5 - 2) + \text{abs}(2 - 4)) = -(3+2) = -5$$

$$\begin{aligned} \text{lurus} &= (\text{abs}(5 - 2) + \text{abs}(2 - 4)) - (2 * -5) \\ &= 5 - (-10) = 15 \end{aligned}$$

$$\begin{aligned} h(n) &= \text{lurus} + \sqrt{2} * \text{diagonal} \\ &= 15 + 1.4(-5) = 8 \end{aligned}$$

$$g(1) = (\text{abs}(4+(1)) + \text{abs}(3+(-1)))$$

$$\begin{aligned}
 &= 5 + 2 = 7 \\
 f(1) &= h(n) + g(n) \\
 &= 8 + 7 = 15 \\
 \text{next-}f &= \min(\text{next-}f, f(1)) \\
 &= \min(8.6, 15) \\
 &= 8.6
 \end{aligned}$$

Masukkan nilai $f(2)$ gerak diagonal kanan bawah (5,2) ke dalam OPEN. Lakukan pengecekan, apakah gerak diagonal kanan bawah(5,2) ada dalam CLOSE? Jika tidak, masukkan ke CLOSE. Jika iya, lakukan pengecekan apakah nilai g saat ini lebih kecil dari g sebelumnya? Jika iya, lakukan penggantian node dalam CLOSE. Jika tidak, abaikan. Hapus gerak diagonal kanan bawah(5,2) dalam OPEN. Hapus nilai $f(2)$.

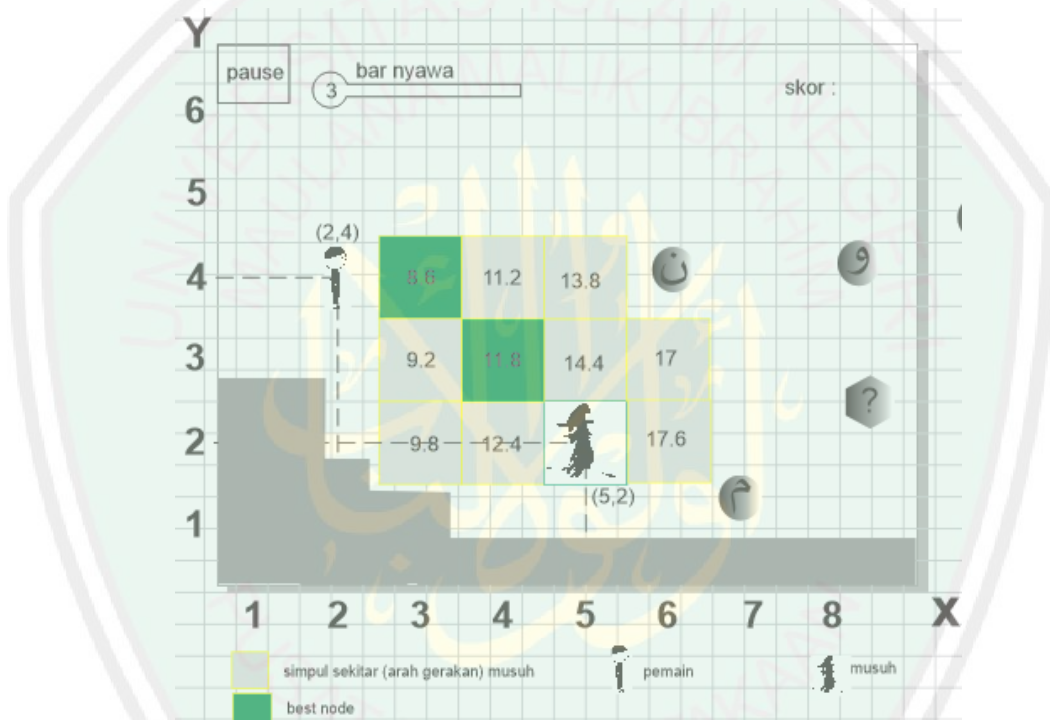
Gerak bawah (4,2)

$$\begin{aligned}
 \text{diagonal} &= -(\text{abs}(4 - 2) + \text{abs}(2 - 4)) = -(2+2) = -4 \\
 \text{lurus} &= (\text{abs}(4 - 2) + \text{abs}(2 - 4)) - (2*-4) \\
 &= 4 - (-8) = 12 \\
 h(n) &= \text{lurus} + \sqrt{2} * \text{diagonal} \\
 &= 12 + 1.4(-4) = 6.4 \\
 g(1) &= (\text{abs}(4+(0)) + \text{abs}(3+(-1))) \\
 &= 4 + 2 = 6 \\
 f(1) &= h(n) + g(n) \\
 &= 6.4 + 6 = 12.6 \\
 \text{next-}f &= \min(\text{next-}f, f(1)) \\
 &= \min(8.6, 12.6) \\
 &= 8.6
 \end{aligned}$$

Masukkan nilai $f(2)$ gerak bawah (4,2) ke dalam OPEN. Lakukan pengecekan, apakah gerak bawah(4,2) ada dalam CLOSE? Jika tidak, masukkan ke CLOSE. Jika iya, lakukan pengecekan apakah nilai g saat ini lebih kecil dari g sebelumnya? Jika iya, lakukan penggantian node

dalam CLOSE. Jika tidak, abaikan. Hapus gerak bawah(4,2) dalam OPEN. Hapus nilai $f(2)$.

Setelah semua node terdekat yang mungkin dibangkitkan, simpan nilai $next-f$ ke dalam $bestnode$. Berdasarkan perhitungan di atas, $next-f$ bernilai 8.6 dengan koordinat (3,4) dengan arah diagonal kiri atas. Jadi, dalam $bestnode$ tersimpan $\{next-f(2)$ koordinat (3,4), $next-f(1)$ koordinat (4,3) $\}$.



Gambar 3.5 Jalur terbaik untuk pergerakan Musuh

(3) Musuh bergerak menuju player sesuai dengan nilai dalam

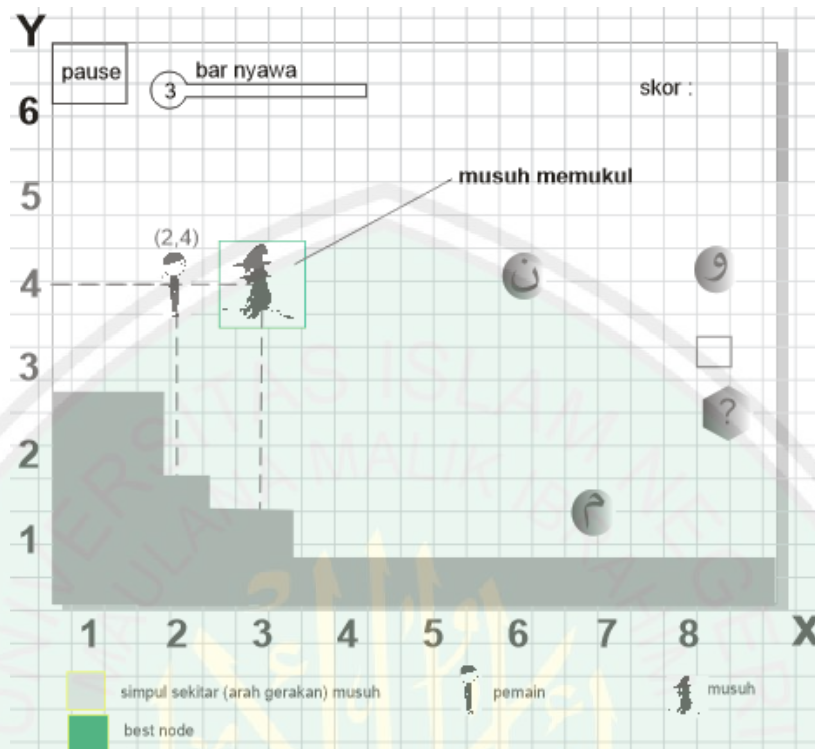
bestnode

Jika koordinat musuh – 1 dengan pemain, maka jalankan

backtracking dari *bestnode* $\{f(2)$ koordinat (3,4), $f(1)$ koordinat

(4,3) $\}$ secara terbalik

(4) Musuh memukul player



Gambar 3.6 Musuh memukul *player*

3.4 Perancangan Aplikasi Game

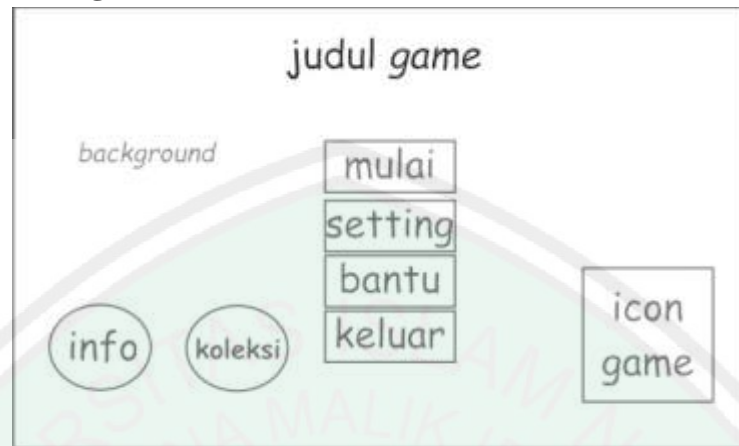
3.4.1 Perancangan Antarmuka Intro

Awal game dimainkan, akan muncul *splashscreen* tentang judul aplikasi. Selanjutnya, akan muncul tampilan menu *game*. Adapun desain gambar adalah sebagai berikut.



Gambar 3.7 Desain *splashscreen game*

3.4.2 Perancangan Antarmuka Menu *Game*



Gambar 3.8 Desain menu *game*

Penjelasan dari setiap tombol menu adalah sebagai berikut :

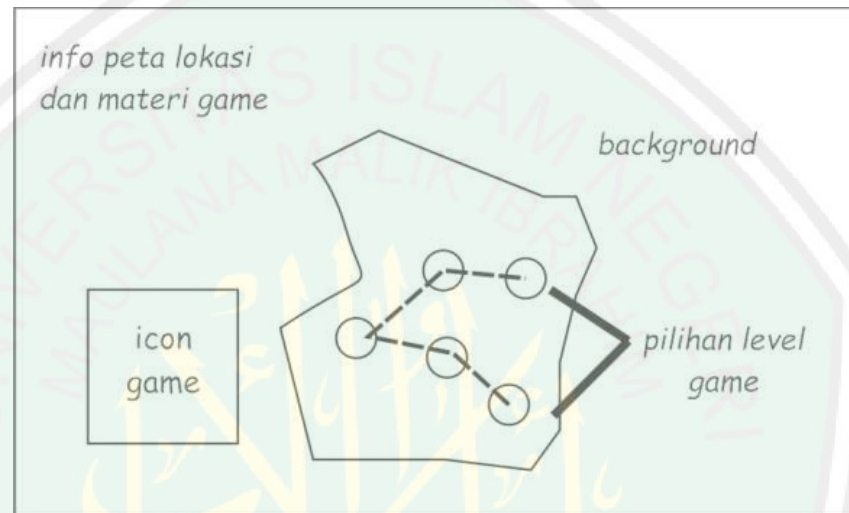
1. Mulai : tombol untuk memulai *game*
2. Setting : tombol untuk mengatur *game*
3. Bantu : tombol untuk membantu pemain untuk mengetahui fungsi tiap-tiap tombol ketika bermain
4. Koleksi : tombol untuk melihat koleksi bacaan tajwid yang telah berhasil didapatkan
5. Info : tombol berisi informasi pembuat *game*
6. Keluar : tombol untuk keluar *game*

3.4.3 Perancangan Antarmuka *Game*

Awal permainan, pemain akan dibekali kesempatan bermain sebanyak tiga kali dengan kesehatan sebanyak 100 %.

- Perancangan level permainan

Perancangan ini menggambarkan kondisi layar aplikasi game ketika pemain pada posisi untuk memulai permainan. Terdapat pilihan sebanyak lima level. Berikut adalah desain pilihan level.



Gambar 3.9 Desain menu pilihan level

1. Level pertama

Pada level pertama, Materi tajwid dalam *game* adalah tentang bacaan *idgham bighunnah*. Misi yang harus dilakukan adalah mengunjungi makam Syeh Maulana Malik Ibrahim. Untuk sampai ke tujuan, Zaid harus mengumpulkan huruf-huruf *idgham bighunnah* minimal sebanyak 20 dan menjawab pertanyaan dengan benar bacaan *idgham bighunnah* minimal 1.

2. Level kedua

Pada level kedua, materi tajwid dalam *game* adalah tentang bacaan *idgham bilaghunnah*. Misi yang harus dilakukan adalah mengunjungi makam sunan Giri yang ada di kota Gresik. Untuk sampai ke tujuan,

Zaid harus mengumpulkan huruf-huruf idgham bilaghunnah minimal sebanyak 20 dan menjawab pertanyaan dengan benar bacaan *idgham bilaghunnah* minimal satu.

Pemain akan dihadang oleh musuh berbentuk penyihir dan kura-kura. Kesehatan pemain akan berkurang bila terkena serangan dari musuh.

3. Level ketiga

Pada level ketiga, materi tajwid dalam *game* adalah tentang bacaan *iqlab*. Misi yang harus dilakukan adalah mengunjungi makam sunan Ampel yang ada di kota Surabaya. Untuk sampai ke tujuan, Zaid harus mengumpulkan huruf-huruf *iqlab* minimal sebanyak 20 dan menjawab pertanyaan dengan benar bacaan *iqlab* minimal satu.

Pemain akan dihadang oleh musuh berbentuk penyihir dan kura-kura. Kesehatan pemain akan berkurang bila terkena serangan dari musuh.

4. Level keempat

Materi tajwid dalam level ini adalah tentang bacaan *izh-har*. Misi yang harus dilakukan adalah mengunjungi makam sunan Drajad yang ada di kota Lamongan. Untuk sampai ke tujuan, Zaid harus mengumpulkan huruf-huruf *izh-har* minimal sebanyak 20 dan menjawab pertanyaan dengan benar bacaan *izh-har* minimal satu.

Pemain akan dihadang oleh musuh berbentuk penyihir dan kura-kura. Kesehatan pemain akan berkurang bila terkena serangan dari musuh.

5. Level kelima

Pada level terakhir, materi tajwid dalam *game* adalah tentang bacaan *ikhfa'*. Misi yang harus dilakukan adalah mengunjungi makam sunan Bonang yang ada di kota Tuban. Untuk sampai ke tujuan, Zaid harus mengumpulkan huruf-huruf *ikhfa'* minimal sebanyak 20 dan menjawab pertanyaan dengan benar bacaan *ikhfa'* minimal satu.

Pemain akan dihadang oleh musuh berbentuk penyihir dan kura-kura. Kesehatan pemain akan berkurang bila terkena serangan dari musuh.

3.4.4 Kebutuhan sistem

Bagian spesifikasi kebutuhan sistem ini, akan dipaparkan tentang kebutuhan perangkat keras (hardware) maupun perangkat lunak (software) yang mendukung dalam pembuatan maupun pada saat pengoperasian program aplikasi.

1. Kebutuhan Perangkat Keras

Kebutuhan perangkat keras yang digunakan untuk mendukung proses pembuatan aplikasi game ini yaitu :

- a. Processor intel pentium B950 2.10 GHz
- b. RAM 2 GB
- c. VGA Nvidia GeForce GT 520M 1 GB
- d. Hardisk 320 GB

- e. Keyboard
- f. Mouse
- g. LCD/Monitor
- h. Speaker

2. Kebutuhan Perangkat Lunak

a. IDE Eclipse

IDE Eclipse merupakan *software open source* yang digunakan untuk melakukan *coding* pembuatan aplikasi bahasa pemrograman Java. Pembuatan aplikasi ini menggunakan *eclipse* versi juno.

b. Inkscape

Software yang digunakan untuk desain grafis berbasis vektor. Pada pembuatan aplikasi *game* ini, *corel draw* digunakan untuk mendesain *background game*, desain *text*, dan desain NPC (*Non Playable Character*).

c. Gimp

Software yang digunakan untuk membantu dalam proses pembuatan desain grafis dengan memberikan efek-efek dalam *game*.

d. Audacity

Software yang digunakan untuk melakukan *editing sound* dan *music* pada aplikasi.

3.4.5 Cara Memainkan *Game*

Cara memainkan game ini cukup mudah. Pemain mengoperasikan game dengan touch (sentuhan). Terdapat berbagai tombol yang digunakan dalam game ini. Tombol tersebut yakni tombol arah panah yang digunakan untuk bergerak ke kiri dan ke kanan serta dua tombol yang digunakan untuk melompat dan menembak.

3.4.6 Kebutuhan PC Pemain

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

Tabel 3.2 Daftar spesifikasi komputer

Kebutuhan	Spesifikasi Minimum	Spesifikasi Rekomendasi
<i>Smartphone</i>	Android versi 2.2 (Froyo)	Android Versi ICS (Ice Cream Sandwich)
RAM	256 MB	512 MB

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Aplikasi

Berikut ini merupakan hasil implementasi aplikasi game pengenalan ilmu tajwid :



Gambar 4.1 Tampilan *splashscreen* game tajwid



Gambar 4.2 Tampilan menu utama

Tampilan menu utama terdapat beberapa tombol yang memiliki fungsi masing-masing, antara lain : mulai, *setting*, bantu, koleksi, info, dan keluar. Apabila pemain menekan tombol *setting*, maka akan diarahkan ke menu

pengaturan *game* yang berisi pengaturan tingkat kesulitan, *audio*, dan kecepatan pemain.



Gambar 4.3 Tampilan menu *setting*

Pada tombol bantu, akan menampilkan informasi tentang petunjuk penggunaan fungsi tiap-tiap tombol ketika bermain.



Gambar 4.4 Tampilan petunjuk permainan

Jika pemain menekan tombol info yang bersimbol “!”, akan menuju halaman informasi pembuat *game*.

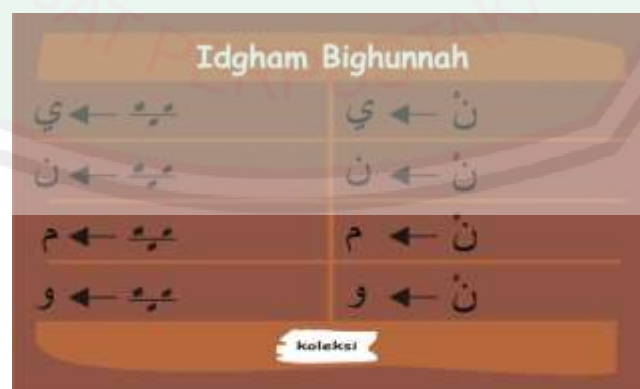


Gambar 4.5 Tampilan info pembuat *game*

Tombol koleksi digunakan untuk melihat koleksi bacaan tajwid yang telah berhasil didapatkan.



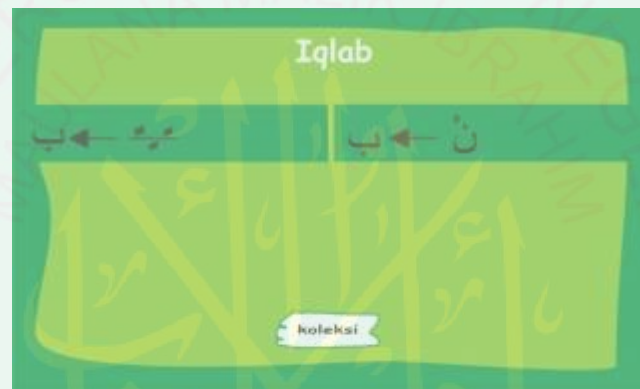
Gambar 4.6 Tampilan koleksi bacaan



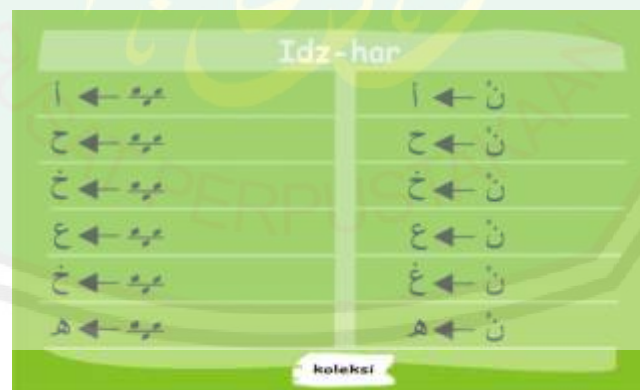
Gambar 4.7 Tampilan koleksi bacaan *idgham bighunnah*



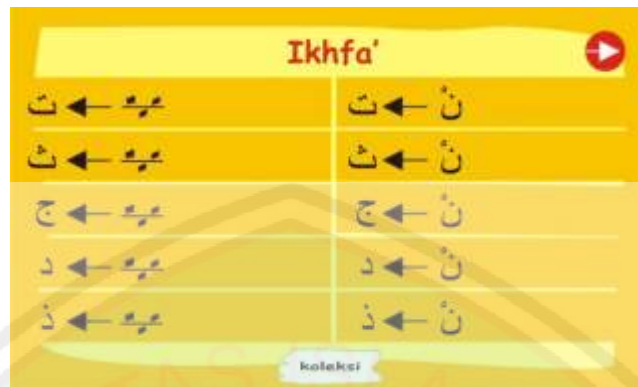
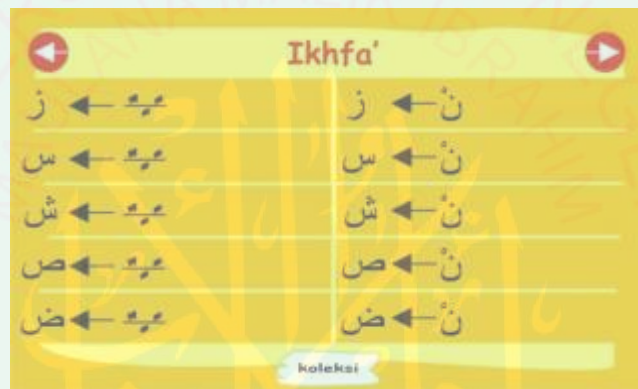
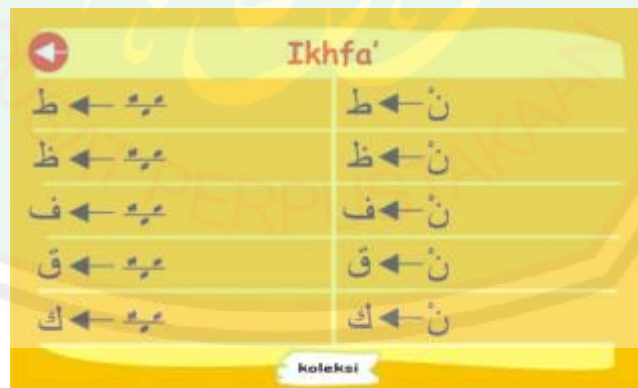
Gambar 4.8 Tampilan koleksi bacaan *idgham bilaghunnah*



Gambar 4.9 Tampilan koleksi bacaan *iqlab*



Gambar 4.10 Tampilan koleksi bacaan *izh-har*

Gambar 4.11 Tampilan koleksi bacaan *ikhfa' 1*Gambar 4.12 Tampilan koleksi bacaan *ikhfa' 2*Gambar 4.13 Tampilan koleksi bacaan *ikhfa' 3*

Untuk memainkan game tajwid ini, pengguna harus menekan tombol mulai. Menu berikutnya akan ditampilkan pilih level. Pemain memilih salah satu dari kelima level tersebut. Level satu tentang *idgham bighunnah*, level dua tentang

idgham *bilaghunnah*, level tiga tentang *iqlab*, level 4 tentang *izh-har*, dan level lima tentang *ikhfa*'.



Gambar 4.14 Tampilan memilih level

Tampilan selanjutnya adalah petunjuk tentang misi yang harus dilakukan oleh pemain.



Gambar 4.15 Tampilan misi level satu

Tahapan selanjutnya adalah dengan menekan disekitar area misi tersebut. Selanjutnya akan diarahkan menuju permainan untuk mengumpulkan item-item huruf bacaan tajwid dan item permata untuk memunculkan pertanyaan sesuai level materi tajwid.



Gambar 4.16 Tampilan *game* level pertama (*idgham bighunnah*)



Gambar 4.17 Tampilan *game* level kedua (*idgham bilaghunnah*)



Gambar 4.18 Tampilan *game* level tiga (*iqlab*)



Gambar 4.19 Tampilan *game* level empat (*izh-har*)



Gambar 4.20 Tampilan *game* level lima (*ikhfa'*)

Pemain akan mendapatkan contoh bacaan sesuai dengan levelnya jika misi sukses dilakukan.



Gambar 4.21 Tampilan misi sukses

Namun, jika misi yang dilakukan gagal, akan tampil pesan misi gagal. Pemain dapat bermain lagi dengan menekan di area gambar tersebut.



Gambar 4.22 Tampilan misi gagal

4.2 Implementasi Algoritma Iterative Deepening A*

Algoritma *Iterative Deepening A** tergantung pada penentuan *f-limit* dan *next-f* yang digunakan untuk proses pencarian *player*. Inisialisasi awal nilai-nilai tersebut terdapat dalam *class F.java*. Berikut kode program yang dimaksud:

```
public static int
fLimit=Integer.MAX_VALUE,next f=Integer.MAX_VALUE;
```

f-limit digunakan untuk proses pembatasan nilai *f* dalam pencarian. Inisialisasi awal, diberi nilai tak hingga. *next-f* digunakan untuk membandingkan nilai *f* terkecil yang nantinya akan digunakan untuk penentuan *bestnode* dan *f-limit* selanjutnya.

Kode program tersebut akan dipanggil dalam sebuah *method* yang diberi nama "IDAstar". Method ini akan dipanggil di *class* yang berisikan musuh.

Berikut ini kode program yang dimaksud:

```
int h, g, fCost, ortogonal, diagonal,xy;
open = new ArrayList<Node>();
close = new ArrayList<Node>();
bestnode = new ArrayList<Node>();

Node awal = new
Node(0,0,0,0,Integer.MAX_VALUE,(int)centerX,(int)centerY,""
```

```

);
    open.add(awal);
    bestnode.add(open.get(0));
    close.add(open.get(0));

    for (int i = 0; i < moveX.length; i++) {
        Node op = new
Node(Integer.parseInt(moveX[i].toString()),
        Integer.parseInt(moveY[i].toString()),
String.valueOf(gerak[i]));
        arah.add(op);
    }
    if(hihi.tujuan==false){
        for (int i = 0; i < arah.size(); i++) {
            Node p = (Node) arah.get(i);
            xy=Math.abs(((int) centerX + p.getX())
                - tokoh.getCenterX())
                +Math.abs(((int) centerY + p.getY())
                - tokoh.getCenterY());
            diagonal = -(xy);
            ortogonal = xy - (2 * diagonal);
            h= (int)Math.abs((ortogonal+1.4*diagonal));
            g=Math.abs((int)centerX+p.getX()) +
Math.abs((int)centerY+p.getY());
            fCost = h + g;

            p.posisix = (int) (centerX + p.getX());
            p.posisiy = (int) (centerY + p.getY());
            F.fCost=fCost;

            if(F.fCost<=F.flimit){
                Node ops = new Node((int) p.getX(), (int)
p.getY(), g, h, fCost, p.posisix, p.posisiy, p.getGerak());
                for (int j = 0; j < close.size(); j++) {
                    Node v = (Node) close.get(j);
                    if (v.getPosisix() == p.posisix &&
v.getPosisiy() == p.posisiy) {
                        if (g > v.g) {
                            } else {
                                close.set(j, v);
                            }
                        sama = sama - 1;
                    } else {
                        }
                    jumlah = sama;
                }
                sama = 8;

                if (jumlah == 8) {

```

```

        open.add(ops);
        open.remove(ops);
        close.add(ops);
    }
}

bestnode.add(Collections.min(close));

kecepatanX = Collections.min(close).getX();
kecepatanY = Collections.min(close).getY();
for (int i = bestnode.size() - 1; i <
bestnode.size(); i++) {
    Node s = (Node) bestnode.get(i);
    centerX = centerX + s.getX();
    centerY = centerY + s.getY();

    F.flimit=s.getFCost();
}

if(Math.abs(centerX-tokoh.getCenterX())<20 &&
Math.abs(centerY-tokoh.getCenterY())<50){
    hihi.tujuan=true;
    open.removeAll(getOpen());
}
}

```

Method tersebut digunakan untuk memberikan proses penentuan jalur terpendek bagi musuh untuk menemukan pemain utama. Pada awalnya, membuat *arraylist* *open*, *close*, dan *bestnode* untuk menampung hasil proses perhitungan algoritma. Setelah itu, menentukan posisi musuh untuk dijadikan sebagai simpul awal. Bila musuh belum sampai tujuan ke *player*(pemain), algoritma akan tetap berjalan. Algoritma menghitung nilai *h* dan *g* untuk menentukan nilai *f* terkecil yang akan dijadikan sebagai *bestnode* dan penentuan pembatasan pencarian berikutnya.

Method IDAstar tersebut membutuhkan kelas bantuan yang digunakan untuk menyimpan nilai *arraylist* pada node OPEN dan untuk membandingkan nilai *f* terkecil. Berikut kode program yang dimaksud.

```

int x,y,g,h,fCost=Integer.MAX_VALUE,posisiX,posisiY;
    String gerak;

    public Node(int x, int y, int g,int h, int fCost,int
    posisiX,int posisiY,String gerak) {
        this.x = x;
        this.y = y;
        this.g = y;
        this.h = y;
        this.fCost = fCost;
        this.posisiX=posisiX;
        this.posisiY=posisiY;
        this.gerak=gerak;
    }

    public Node(int x, int y,String gerak) {
        this.x = x;
        this.y = y;
        this.gerak=gerak;
    }

    public int getFCost() {
        return fCost;
    }

    public void setFCost(int fCost) {
        this.fCost = fCost;
    }

    public int getX() {
        return x;
    }

    public void setX(int x) {
        this.x = x;
    }

    public int getY() {
        return y;
    }

    public void setY(int y) {
        this.y = y;
    }

```

```

public int getG() {
    return g;
}

public void setG(int g) {
    this.g = g;
}

public int getH() {
    return h;
}

public void setH(int h) {
    this.h = h;
}

public String getGerak() {
    return gerak;
}

public void setGerak(String gerak) {
    this.gerak = gerak;
}

public int getPosisix() {
    return posisix;
}

public void setPosisix(int posisix) {
    this.posisix = posisix;
}

public int getPosisiy() {
    return posisiy;
}

public void setPosisiy(int posisiy) {
    this.posisiy = posisiy;
}

@Override
public int compareTo(Node o) {
    F.fCost = fCost;
    F.next_f = o.getFCost();

    if (F.fCost < F.next_f) {
        return -1;
    } else if (F.fCost > F.next_f) {
        return 1;
    } else {

```

```

        return 0;
    }
}

```

Constructor Node memiliki beberapa parameter yang digunakan untuk menampung berbagai informasi tentang posisi koordinat musuh dan nilai-nilai variabel hasil eksekusi algoritma. Nilai-nilai tersebut akan dipanggil untuk menjalankan proses *backtracking* jalur-jalur terpilih ke posisi *player*.

4.3 Uji Coba

4.3.1 Uji Coba Algoritma Iterative Deepening A* (IDA*)

Pengujian penerapan algoritma Iterative Deepening A* pada game tajwid ini menggunakan perangkat keras *Cross Mobile A88* dengan spesifikasi sebagai berikut :

Prosesor : Quad-Core ARMv7 Processor (VFPv4,NEON)

Memory Internal : 1.74 GB

RAM : 512 MB

Resolusi :480 x 854

OS : Android 4.2.1 (jelly bean)

Pengujian dilakukan terhadap pergerakan musuh menemukan pemain sebanyak tiga kali dengan posisi koordinat musuh yang berbeda-beda. Perhitungan waktu menggunakan kode program bawaan dari bahasa pemrograman yang dpanggil ketika algoritma mulai dijalankan dan dilakukan

perhitungan eksekusi waktu ketika algoritma selesai dijalankan. Adapun kode program yang dimaksud adalah sebagai berikut :

```
double waktumulai=System.currentTimeMillis();
waktu=(System.currentTimeMillis()-waktumulai)/1000;
```

a. Pengujian pertama

Pengujian pertama pemain berada dikoordinat(91,336) musuh dikoordinat (690,180). Berikut ini merupakan tampilan game yang akan diuji.



Gambar 4.23 Tampilan aplikasi yang akan diuji

Pada awalnya musuh akan *standby* (diam) diposisi yang sudah ditentukan. Musuh akan menjalankan algoritma bila jarak koordinat x antara pemain dengan musuh lebih kecil dari 600. Jika posisi pemain memenuhi persyaratan algoritma, maka proses algoritma akan dijalankan.



Gambar 4.24 Posisi pemain yang memenuhi persyaratan algoritma

Berikut ini adalah proses algoritma pencarian jalur dengan *iterative deepening A**:





Gambar 4.25 Hasil uji coba IDA* 1



Gambar 4.26 Musuh sampai ke node goal (tujuan)

Tabel 4.1 Percobaan pertama pada game tajwid

Aplikasi	Game Tajwid	Keterangan
Algoritma	Iterative Deepening A*	
Parameter	Koordinat pemain : X = 91 Y = 336 Koordinat musuh : X = 690 Y = 180	
Waktu Eksekusi	0.087 detik	
Jumlah node terpilih	194 Node	
Path (jalur) terpilih	(687),(183)##(684),(186)##(681),(189)## (678),(192)##(675),(195)##(672),(198)## (669),(201)##(666),(204)##(663),(207)## (660),(210)##(657),(213)##(654),(216)## (651),(219)##(648),(222)##(645),(225)## (642),(228)##(639),(231)##(636),(234)## (633),(237)##(630),(240)##(627),(243)## (624),(246)##(621),(249)##(618),(252)## (615),(255)##(612),(258)##(609),(261)## (606),(264)##(603),(267)##(600),(270)## (597),(273)##(594),(276)##(591),(279)## (588),(282)##(585),(285)##(582),(288)##	Solusi ditemukan

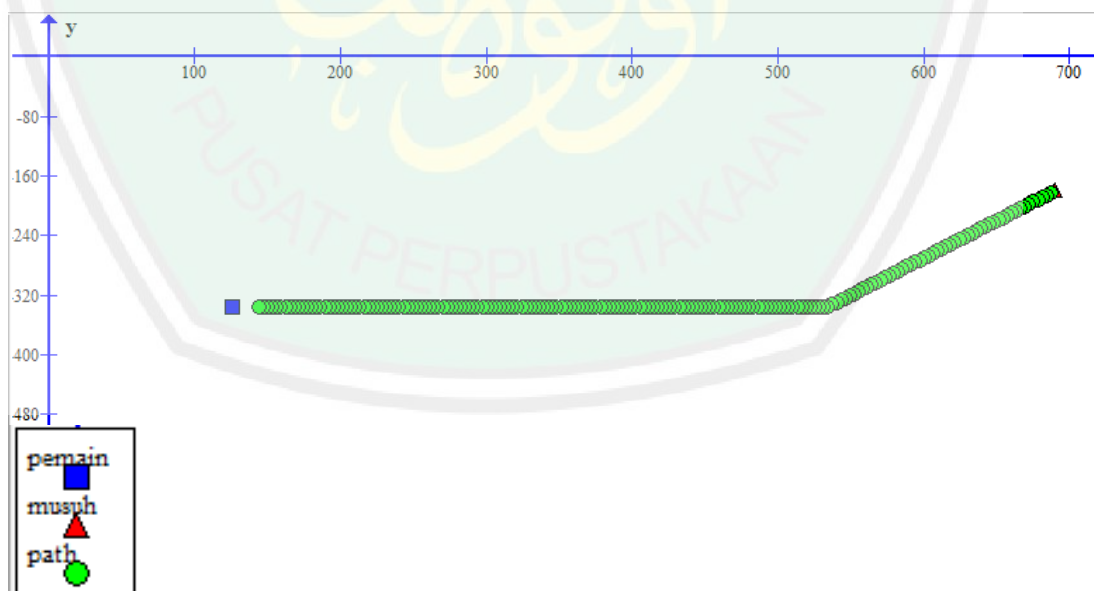
Algoritma	Game Tajwid	Keterangan
Path (jalur) terpilih	(579),(291)##(576),(294)##(573),(297)## (570),(300)##(567),(303)##(564),(306)## (561),(309)##(558),(312)##(555),(315)## (552),(318)##(549),(321)##(546),(324)## (543),(327)##(540),(330)##(537),(333)## (534),(336)##(531),(336)##(528),(336)## (525),(336)##(522),(336)##(519),(336)## (516),(336)##(513),(336)##(510),(336)## (507),(336)##(504),(336)##(501),(336)## (498),(336)##(495),(336)##(492),(336)## (489),(336)##(486),(336)##(483),(336)## (480),(336)##(477),(336)##(474),(336)## (471),(336)##(468),(336)##(465),(336)## (462),(336)##(459),(336)##(456),(336)## (453),(336)##(450),(336)##(447),(336)## (444),(336)##(441),(336)##(438),(336)## (435),(336)##(432),(336)##(429),(336)## (426),(336)##(423),(336)##(420),(336)## (417),(336)##(414),(336)##(411),(336)## (408),(336)##(405),(336)##(402),(336)##	

Algoritma	Game Tajwid	Keterangan
Path (jalur) terpilih	(399),(336)##(396),(336)##(393),(336)## (390),(336)##(387),(336)##(384),(336)## (381),(336)##(378),(336)##(375),(336)## (372),(336)##(369),(336)##(366),(336)## (363),(336)##(360),(336)##(357),(336)## (354),(336)##(351),(336)##(348),(336)## (345),(336)##(342),(336)##(339),(336)## (336),(336)##(333),(336)##(330),(336)## (327),(336)##(324),(336)##(321),(336)## (318),(336)##(315),(336)##(312),(336)## (309),(336)##(306),(336)##(303),(336)## (300),(336)##(297),(336)##(294),(336)## (291),(336)##(288),(336)##(285),(336)## (282),(336)##(279),(336)##(276),(336)## (273),(336)##(270),(336)##(267),(336)## (264),(336)##(261),(336)##(258),(336)## (255),(336)##(252),(336)##(249),(336)## (246),(336)##(243),(336)##(240),(336)## (237),(336)##(234),(336)##(231),(336)## (228),(336)##(225),(336)##(222),(336)##	

Algoritma	Game Tajwid	Keterangan
Path (jalur) terpilih	(219),(336)##(216),(336)##(213),(336)## (327),(336)##(324),(336)##(321),(336)## (318),(336)##(315),(336)##(312),(336)## (309),(336)##(306),(336)##(303),(336)## (300),(336)##(297),(336)##(294),(336)## (291),(336)##(288),(336)##(285),(336)## (282),(336)##(279),(336)##(276),(336)## (273),(336)##(270),(336)##(267),(336)## (264),(336)##(261),(336)##(258),(336)## (255),(336)##(252),(336)##(249),(336)## (246),(336)##(243),(336)##(240),(336)## (237),(336)##(234),(336)##(231),(336)## (228),(336)##(225),(336)##(222),(336)## (219),(336)##(216),(336)##(213),(336)## (210),(336)##(207),(336)##(204),(336)## (201),(336)##(198),(336)##(195),(336)## (192),(336)##(189),(336)##(186),(336)## (183),(336)##(180),(336)##(177),(336)## (174),(336)##(171),(336)##(168),(336)## (165),(336)##(162),(336)##(159),(336)##	

Algoritma	Game Tajwid	Keterangan
Path (jalur) terpilih	(156),(336)##(153),(336)##(150),(336)## (147),(336)##(144),(336)##(141),(336)## (138),(336)##(135),(336)##(132),(336)## (129),(336)##(126),(336)##(123),(336)## (120),(336)##(117),(336)##(114),(336)## (111),(336)##(108),(336)	

Berdasarkan tabel 4.1 di atas, pencarian posisi pemain dengan koordinat XY (91,336) dan musuh(690,180) menggunakan algoritma *iterative deepening A** ditemukan adanya solusi dengan jumlah node terpilih sebanyak 194 node. Waktu yang dilakukan untuk menemukan solusi adalah 0.087 detik.



Gambar 4.27 Grafik *path* (jalur) terpilih yang dilalui musuh

b. Pengujian kedua

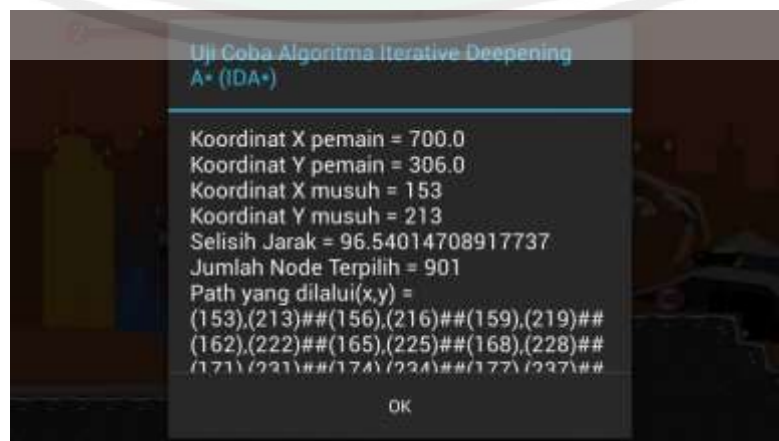
Pengujian kedua pemain berada dikoordinat(700,306) musuh dikoordinat (153,213). Berikut ini merupakan tampilan game yang akan diuji.



Gambar 4.28 Tampilan aplikasi yang akan diuji

Pada awalnya musuh akan *standby* (diam) diposisi yang sudah ditentukan. Musuh akan menjalankan algoritma bila jarak koordinat x antara pemain dengan musuh lebih kecil dari 600. Jika posisi pemain memenuhi persyaratan algoritma, maka proses algoritma akan dijalankan.

Berikut ini adalah proses algoritma pencarian jalur dengan *iterative deepening A**:



Uji Coba Algoritma Iterative Deepening
A* (IDA*)

(171),(231)##(174),(234)##(177),(237)##
(180),(240)##(183),(243)##(186),(246)##
(189),(249)##(192),(249)##(195),(249)##
(198),(249)##(201),(249)##(204),(249)##
(207),(249)##(210),(249)##(213),(249)##
(216),(249)##(219),(249)##(222),(249)##
(225),(249)##(228),(249)##(231),(249)##
(234),(249)##(237),(249)##(240),(249)##
(243),(249)##(246),(249)##(249),(249)##
(252),(249)##(255),(249)##(258),(249)##

OK

Uji Coba Algoritma Iterative Deepening
A* (IDA*)

(261),(249)##(264),(249)##(267),(249)##
(270),(249)##(273),(249)##(276),(249)##
(279),(249)##(282),(249)##(285),(249)##
(288),(249)##(291),(249)##(294),(249)##
(297),(249)##(300),(249)##(303),(249)##
(306),(249)##(309),(249)##(312),(249)##
(315),(249)##(318),(249)##(321),(249)##
(324),(249)##(327),(249)##(330),(249)##
(333),(249)##(336),(249)##(339),(249)##
(342),(249)##(345),(249)##(348),(249)##

OK

Uji Coba Algoritma Iterative Deepening
A* (IDA*)

(351),(249)##(354),(249)##(357),(249)##
(360),(249)##(363),(249)##(366),(249)##
(369),(249)##(372),(249)##(375),(249)##
(378),(249)##(381),(249)##(384),(249)##
(387),(249)##(390),(249)##(393),(249)##
(396),(249)##(399),(249)##(402),(249)##
(405),(249)##(408),(249)##(411),(249)##
(414),(249)##(417),(249)##(420),(249)##
(423),(249)##(426),(249)##(429),(249)##
(432),(249)##(435),(249)##(438),(249)##

OK

Uji Coba Algoritma Iterative Deepening
A* (IDA*)

(441),(249)##(444),(249)##(447),(249)##
(450),(249)##(453),(249)##(456),(249)##
(459),(249)##(462),(249)##(465),(249)##
(468),(249)##(471),(249)##(474),(249)##
(477),(249)##(480),(249)##(483),(249)##
(486),(249)##(489),(249)##(492),(249)##
(495),(249)##(498),(249)##(501),(249)##
(504),(249)##(507),(249)##(510),(249)##
(513),(249)##(516),(249)##(519),(249)##
(522),(249)##(525),(249)##(528),(249)##

OK



Gambar 4.29 Hasil uji coba IDA* 2



Gambar 4.30 Musuh sampai ke node goal (tujuan)

Tabel 4.2 Percobaan kedua pada game tajwid

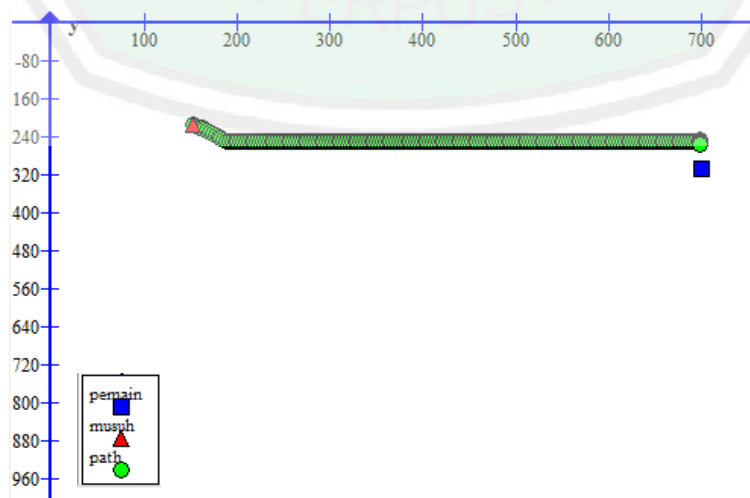
Aplikasi	Game Tajwid	Keterangan
Algoritma	Iterative Deepening A*	
Parameter	Koordinat pemain : X = 700 Y = 306 Koordinat musuh : X = 153 Y = 213	
Waktu Eksekusi	0.122 detik	
Jumlah node terpilih	194 Node	
Path (jalur) terpilih	(153),(213)##(156),(216)##(159),(219)## (162),(222)##(165),(225)##(168),(228)## (171),(231)##(174),(234)##(177),(237)## (180),(240)##(183),(243)##(186),(246)## (189),(249)##(192),(249)##(195),(249)## (198),(249)##(201),(249)##(204),(249)## (207),(249)##(210),(249)##(213),(249)## (216),(249)##(219),(249)##(222),(249)## (225),(249)##(228),(249)##(231),(249)## (234),(249)##(237),(249)##(240),(249)## (243),(249)##(246),(249)##(249),(249)## (252),(249)##(255),(249)##(258),(249)## (261),(249)##(264),(249)##(267),(249)##	Solusi ditemukan

Algoritma	Game Tajwid	Keterangan
Path (jalur) terpilih	(270),(249)##(273),(249)##(276),(249)## (279),(249)##(282),(249)##(285),(249)## (288),(249)##(291),(249)##(294),(249)## (297),(249)##(300),(249)##(303),(249)## (306),(249)##(309),(249)##(312),(249)## (315),(249)##(318),(249)##(321),(249)## (324),(249)##(327),(249)##(330),(249)## (333),(249)##(336),(249)##(339),(249)## (342),(249)##(345),(249)##(348),(249)## (351),(249)##(354),(249)##(357),(249)## (360),(249)##(363),(249)##(366),(249)## (369),(249)##(372),(249)##(375),(249)## (378),(249)##(381),(249)##(384),(249)## (387),(249)##(390),(249)##(393),(249)## (396),(249)##(399),(249)##(402),(249)## (405),(249)##(408),(249)##(411),(249)## (414),(249)##(417),(249)##(420),(249)## (423),(249)##(426),(249)##(429),(249)## (432),(249)##(435),(249)##(438),(249)## (441),(249)##(444),(249)##(447),(249)## (450),(249)##(453),(249)##(456),(249)## (459),(249)##(462),(249)##(465),(249)##	

Algoritma	Game Tajwid	Keterangan
Path (jalur) terpilih	(468),(249)##(471),(249)##(474),(249)## (477),(249)##(480),(249)##(483),(249)## (486),(249)##(489),(249)##(492),(249)## (495),(249)##(498),(249)##(501),(249)## (504),(249)##(507),(249)##(510),(249)## (513),(249)##(516),(249)##(519),(249)## (522),(249)##(525),(249)##(528),(249)## (531),(249)##(534),(249)##(537),(249)## (540),(249)##(543),(249)##(546),(249)## (549),(249)##(552),(249)##(555),(249)## (558),(249)##(561),(249)##(564),(249)## (567),(249)##(570),(249)##(573),(249)## (576),(249)##(579),(249)##(582),(249)## (585),(249)##(588),(249)##(591),(249)## (594),(249)##(597),(249)##(600),(249)## (603),(249)##(606),(249)##(609),(249)## (612),(249)##(615),(249)##(618),(249)## (621),(249)##(624),(249)##(627),(249)## (630),(249)##(633),(249)##(636),(249)## (639),(249)##(642),(249)##(645),(249)## (648),(249)##(651),(249)##(654),(249)##	

Algoritma	Game Tajwid	Keterangan
Path (jalur) terpilih	(657),(249)##(660),(249)##(663),(249)## (666),(249)##(669),(249)##(672),(249)## (675),(249)##(678),(249)##(681),(249)## (678),(212)##(681),(215)##(684),(218)## (687),(221)##(690),(224)##(693),(227)## (696),(230)##(699),(233)##(699),(236)## (699),(239)##(699),(242)##(699),(242)## (699),(245)##(699),(248)##(699),(251)## (699),(254)##(699),(257)##	

Berdasarkan tabel 4.1 di atas, pencarian posisi pemain dengan koordinat XY (700,306) dan musuh(153,213) menggunakan algoritma *iterative deepening A** ditemukan adanya solusi dengan jumlah node terpilih sebanyak 194 node. Waktu yang dilakukan untuk menemukan solusi adalah 0.122 detik.



Gambar 4.31 Grafik *path* (jalur) terpilih yang dilalui musuh

c. Pengujian ketiga

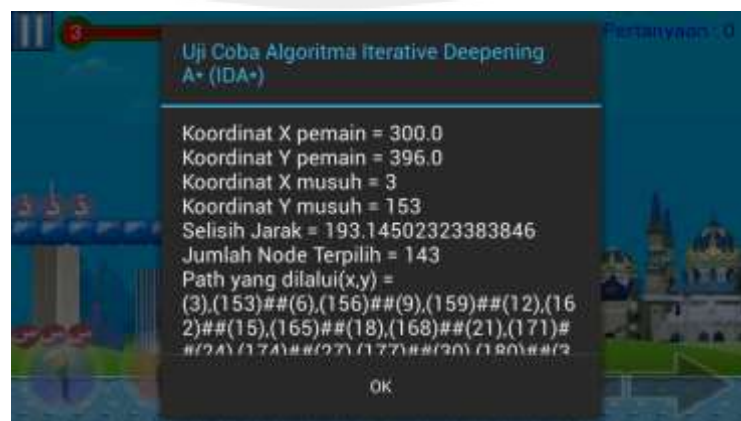
Pengujian ketiga pemain berada dikoordinat(300,396) musuh dikoordinat (3,153). Berikut ini merupakan tampilan game yang akan diuji.

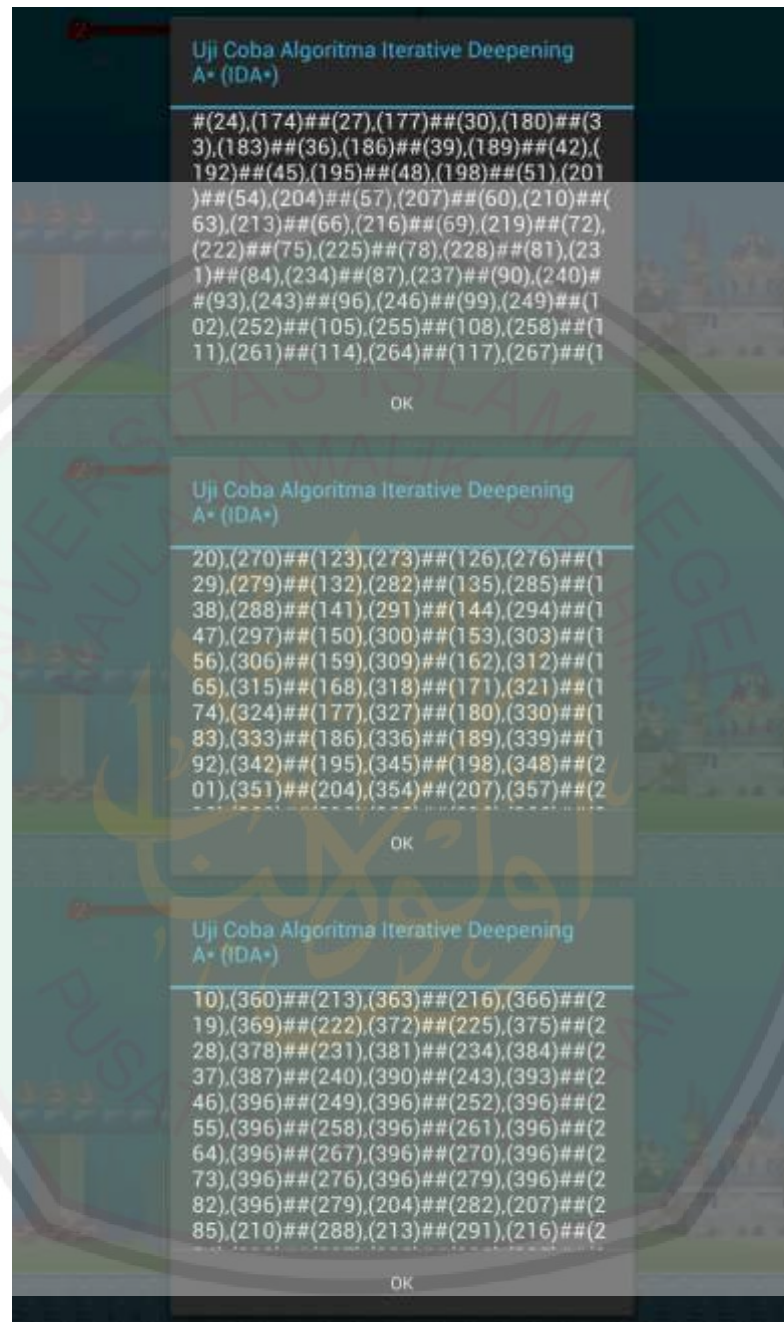


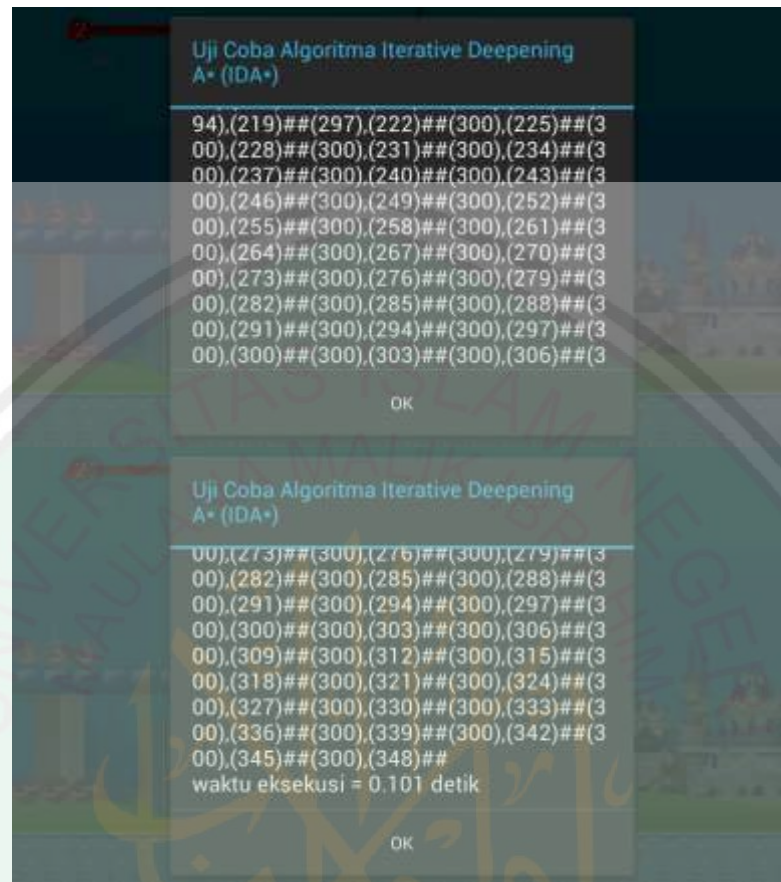
Gambar 4.32 Tampilan aplikasi yang akan diuji

Pada awalnya musuh akan *standby* (diam) diposisi yang sudah ditentukan. Musuh akan menjalankan algoritma bila jarak koordinat x antara pemain dengan musuh lebih kecil dari 600. Jika posisi pemain memenuhi persyaratan algoritma, maka proses algoritma akan dijalankan.

Berikut ini adalah proses algoritma pencarian jalur dengan *iterative deepening A**:







Gambar 4.33 Hasil uji coba IDA* 3



Gambar 4.34 Musuh sampai ke node goal (tujuan)

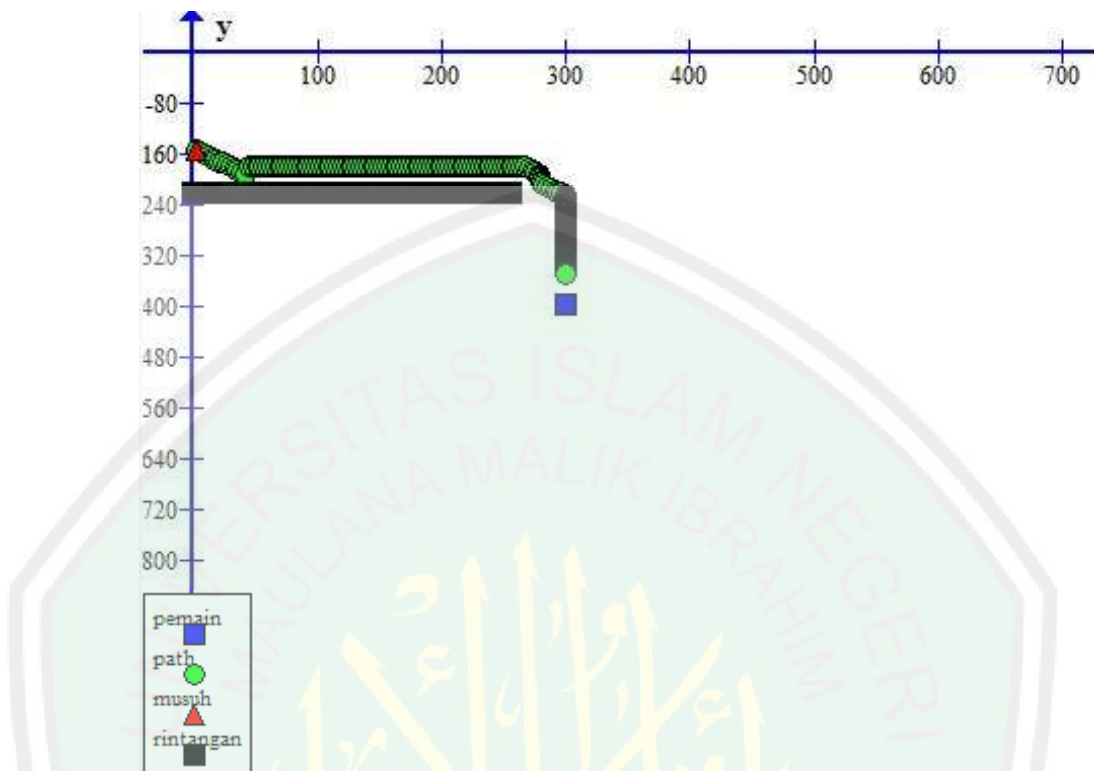
Tabel 4.3 Percobaan ketiga pada game tajwid

Aplikasi	Game Tajwid	Keterangan
Algoritma	Iterative Deepening A*	
Parameter	Koordinat pemain : X = 300 Y = 396 Koordinat musuh : X = 3 Y = 153	
Waktu Eksekusi	0.101 detik	
Jumlah node terpilih	143 Node	
Path (jalur) terpilih	(3),(153)##(6),(156)##(9),(159)##(12),(162)##(15),(165)##(18),(168)##(21),(171)##(24),(174)##(27),(177)##(30),(180)##(33),(183)##(36),(186)##(39),(189)##(42),(192)##(45),(195)##(48),(198)##(51),(201)##(54),(204)##(57),(207)##(60),(210)##(63),(213)##(66),(216)##(69),(219)##(72),(222)##(75),(225)##(78),(228)##(81),(231)##(84),(234)##(87),(237)##(90),(240)##(93),(243)##(96),(246)##(99),(249)##(102),(252)##(105),(255)##(108),(258)##(111),(261)##(114),(264)##(117),(267)##(120),(270)##(123),(273)##(126),(276)##	Solusi ditemukan

Algoritma	Game Tajwid	Keterangan
Path (jalur) terpilih	(129),(279)##(132),(282)##(135),(285)## (138),(288)##(141),(291)##(144),(294)## (147),(297)##(150),(300)##(153),(303)## (156),(306)##(159),(309)##(162),(312)## (165),(315)##(168),(318)##(171),(321)## (174),(324)##(177),(327)##(180),(330)## (183),(333)##(186),(336)##(189),(339)## (192),(342)##(195),(345)##(198),(348)## (201),(351)##(204),(354)##(207),(357)## (210),(360)##(213),(363)##(216),(366)## (219),(369)##(222),(372)##(225),(375)## (228),(378)##(231),(381)##(234),(384)## (237),(387)##(240),(390)##(243),(393)## (246),(396)##(249),(396)##(252),(396)## (255),(396)##(258),(396)##(261),(396)## (264),(396)##(267),(396)##(270),(396)## (273),(396)##(276),(396)##(279),(396)## (282),(396)##(279),(204)##(282),(207)## (285),(210)##(288),(213)##(291),(216)## (294),(219)##(297),(222)##(300),(225)## (300),(228)##(300),(231)##(300),(234)## (300),(237)##(300),(240)##(300),(243)##	

Algoritma	Game Tajwid	Keterangan
Path (jalur) terpilih	(300),(246)##(300),(249)##(300),(252)## (300),(255)##(300),(258)##(300),(261)## (300),(264)##(300),(267)##(300),(270)## (300),(273)##(300),(276)##(300),(279)## (300),(282)##(300),(285)##(300),(288)## (300),(291)##(300),(294)##(300),(297)## (300),(300)##(300),(303)##(300),(306)## (300),(309)##(300),(312)##(300),(315)## (300),(318)##(300),(321)##(300),(324)## (300),(327)##(300),(330)##(300),(333)## (300),(336)##(300),(339)##(300),(342)## (300),(345)##(300),(348)	

Berdasarkan tabel 4.1 di atas, pencarian posisi pemain dengan koordinat XY (300,396) dan musuh(3,153) menggunakan algoritma *iterative deepening A** ditemukan adanya solusi dengan jumlah node terpilih sebanyak 143 node. Waktu yang dilakukan untuk menemukan solusi adalah 0.101 detik.



Gambar 4.35 Grafik *path* (jalur) terpilih yang dilalui musuh

4.3.2 Uji Coba Pengguna

Pengujian kepada pengguna dilakukan dengan pengambilan sampel secara acak terhadap responden dengan rentang usia 10 – 17 tahun. Jumlah responden yang diambil sebanyak 30. Pengujian dilakukan pada tanggal 28 Januari dan 11 Februari 2014. Pengambilan data menggunakan kuisisioner yang berisi 8 pertanyaan dengan skor penilaian 1 sampai 9. Semakin tinggi skor penilaian menandakan bahwa predikat penilaian tinggi. Proses pengukuran hasil menggunakan rumus statistika kategorisasi data. Berikut ini merupakan rumus untuk mengukur hasil uji coba kuisisioner :

Jumlah Item = 8

Skor Maksimal	= 9
Skor Minimal	= 1
Nilai tertinggi	= <i>Skor maksimal X jumlah item</i> = $9 \times 8 = 72$
Nilai Terendah	= <i>Skor minimal X jumlah item</i> = $1 \times 8 = 8$
Mean	= $\frac{1}{2} \times (\text{skor maksimal} + \text{skor minimal}) \times \text{jumlah item}$ = $0.5 \times (9 + 1) \times 8 = 40$
Standar Deviasi	= $\frac{1}{6} \times (\text{nilai tertinggi} - \text{nilai terendah})$ = $\frac{1}{6} \times (72 - 8)$ = 11
Predikat :	
Sedang	= <i>Mean - (1 X Standar Deviasi)</i> = $40 - (1 \times 11) = 29$
Sedang	= <i>Mean + (1 X Standar Deviasi)</i> = $40 + (1 \times 11) = 51$
Sedang	= 29 s.d. 51
Tinggi	= ≥ 52
Rendah	= ≤ 28

Berikut ini merupakan rekapitulasi hasil uji coba responden dengan menerapkan rumus tersebut.

Tabel 4.4 Rekapitulasi hasil uji coba responden

No	Item								Jumlah	Predikat
	1	2	3	4	5	6	7	8		
1	6	9	7	8	7	6	6	9	58	Tinggi
2	7	9	8	7	8	7	7	9	62	Tinggi
3	9	9	7	7	7	7	7	9	62	Tinggi
4	7	9	7	7	8	7	7	9	61	Tinggi
5	8	9	9	8	7	9	8	9	67	Tinggi
6	8	9	8	8	8	7	7	9	64	Tinggi
7	8	9	8	8	9	9	9	9	69	Tinggi
8	9	9	9	8	8	9	8	9	69	Tinggi
9	8	9	9	9	8	9	8	9	69	Tinggi
10	9	9	8	8	8	9	8	9	68	Tinggi
11	9	9	8	8	9	9	9	9	70	Tinggi
12	8	9	8	8	8	8	8	9	66	Tinggi
13	9	9	9	8	7	8	9	9	68	Tinggi
14	9	9	8	7	8	8	8	9	66	Tinggi
15	9	9	9	8	9	9	9	9	71	Tinggi
16	9	9	9	9	9	9	9	9	72	Tinggi
17	9	9	9	9	8	9	9	9	71	Tinggi
18	9	9	8	8	9	8	8	9	68	Tinggi
19	9	9	9	9	9	9	9	9	72	Tinggi
20	9	9	9	8	9	9	9	9	71	Tinggi
21	9	9	9	9	8	9	9	9	71	Tinggi
22	9	9	9	9	8	9	9	9	71	Tinggi
23	9	9	9	9	8	9	9	9	71	Tinggi
24	9	9	9	9	9	9	9	9	72	Tinggi
25	9	9	9	9	8	8	8	9	69	Tinggi
26	9	9	9	9	7	8	8	9	68	Tinggi
27	9	9	9	9	7	9	9	9	70	Tinggi
28	9	9	9	9	7	9	9	9	70	Tinggi
29	9	9	9	9	8	8	8	9	69	Tinggi
30	9	9	8	8	8	8	8	9	67	Tinggi

Dari data tersebut, dapat disimpulkan bahwa aplikasi ini mendapat predikat tinggi sebanyak 100 % dari total 30 responden. Hasil ini menunjukkan bahwa secara keseluruhan dari item-item pertanyaan yang diajukan dapat memenuhi kebutuhan dan tujuan aplikasi yang dibuat.

Adapun analisis per item pertanyaan menunjukkan hasil sebagai berikut :

Tabel 4.5 Analisis pertanyaan kuisisioner

No	Item	Total (Prosentase %)								
		1	2	3	4	5	6	7	8	9
1	Saya tahu cara memainkan game tajwid ini	0(0)	0(0)	0(0)	0(0)	0(0)	1(3.3)	2(6.7)	5(16.7)	22(73.3)
2	Game tajwid ini menarik dan menyenangkan	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	30(100)
3	Game ini membantuku untuk mengenal ilmu tajwid	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	3(10)	9(30)	18(60)
4	Suara contoh bacaan memudahkan untuk belajar ilmu tajwid	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	4(13.3)	13(43.3)	13(43.3)
5	Tombol yang ada di game ini cukup nyaman	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	7(23.3)	15(50)	8(26.7)
6	Materi ilmu tajwid yang disampaikan bisa saya pahami	0(0)	0(0)	0(0)	0(0)	0(0)	1(3.3)	4(13.3)	8(26.7)	17(56.7)
7	Saya mengalami kesulitan dalam menjawab soal	0(0)	0(0)	0(0)	0(0)	0(0)	1(3.3)	4(13.3)	11(36.7)	14(46.7)
8	Tampilan gambar dalam game ini menarik	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	30(100)

Berdasarkan tabel tersebut, pemain mengetahui cara memainkan game tajwid ini sebanyak 73.3 persen dengan skor 9, 16.7 % dengan skor 8, 6.7 % dengan skor 7, dan 3.3 % dengan skor 6. Dari data tersebut, dapat disimpulkan bahwasannya, pemain(*player*) sangat mengetahui cara memainkan *game* ini. Selain itu, sebanyak 100 % dari total responden menilai bahwasannya *game* tajwid ini sangat menarik dan menyenangkan untuk dimainkan. Sebanyak 60 % memberi skor 9 bahwasannya memainkan game ini dapat membantu untuk mengenal ilmu tawid, lainnya 30 % dengan skor 8, 10 % dengan skor 7. Adapun dari suara contoh bacaan yang terdapat dalam game memudahkan untuk belajar ilmu tajwid mendapat 43.3 % dengan skor 9, 43.3 % dengan skor 8, dan 13.3 % dengan skor 7. Dari data tersebut, dapat disimpulkan bahwasannya suara contoh bacaan dalam game cukup membantu untuk mengenalkan ilmu tajwid.

Mengenai adanya penggunaan tombol dalam game, sebanyak 26.7 % menilai dengan skor 9, 50 % dengan skor 8, 23.3 % dengan skor 7. Hal ini dapat disimpulkan bahwasannya penggunaan tombol dalam game dirasa cukup nyaman. Selain itu, sebanyak 56.7 % memberi skor 9 terhadap materi ilmu tajwid yang disampaikan dalam game, sisanya sebanyak 26.7 % dengan skor 8, 13.3 % dengan skor 7, dan 3.3 % dengan skor 6. Dari data tersebut, dapat disimpulkan bahwa materi ilmu tajwid yang ada dalam game dapat dipahami. Adapun tingkat kesulitan dalam menjawab pertanyaan dalam game, responden memberikan nilai 9 sebanyak 46.7 %, 36.7 % dengan skor 8, 13,3 % dengan skor 7, dan 3.3 % dengan skor 6. Dari segi tampilan, sebanyak 100 % responden mengatakan bahwa tampilan gambar dalam game ini sangat menarik dengan skor 9.

4.4 Integrasi Game Tajwid dengan Islam

Al-Quran merupakan firman Allah yang harus dibaca, dipelajari, dan diamalkan. Sebagai seorang muslim, minimal yang harus dikerjakan adalah membaca Al-Quran dengan benar dan sesuai dengan kaidah-kaidahnya. Untuk mencapai hal tersebut, perlu mempelajari ilmu khusus untuk membaca Al-Quran yang dikenal dengan ilmu tajwid. Tujuan mempelajari tajwid adalah untuk menjaga lidah agar terhindar dari kesalahan-kesalahan dalam membaca Al-Quran. Dasar hukum wajibnya membaca Al-Quran dengan tajwid terdapat dalam surat Al-Muzzammil: 4

وَرَتِّلِ الْقُرْآنَ تَرْتِيلاً ﴿٤﴾

dan bacalah Al-Quran itu dengan perlahan-lahan (bertajwid).

Ayat ini memerintahkan kepada setiap muslim agar selalu membaca Al-Quran dengan tidak tergesa-gesa sehingga membantu dalam memahami dan merenungi kandungan isi Al-Quran.

Untuk mempelajari ilmu tajwid dapat menggunakan media yang bermacam-macam yakni : buku, buku digital, *software*, bahkan *game*. Dalam *game* pengenalan tajwid ini, dibuat untuk memudahkan dalam mempelajari ilmu tajwid. Melalui *game* ini, pemain akan mengenal huruf-huruf bacaan tajwid dengan cara mengambil item-item huruf tajwid yang tersebar di area permainan sesuai hukum bacaannya yakni *idgham bighunnah*, *idgham bilaghunnah*, *iqlab*, *izh-har*, *ikhfa'*. Tidak hanya itu, pemain juga akan belajar materi tajwid dengan menjawab pertanyaan yang tersebar di area permainan.

Apabila petunjuk misi berhasil dilakukan pemain maka akan mendapatkan contoh bacaan tajwid beserta suara membacanya. Berdasarkan hasil uji coba kepada 30 responden tentang fitur, desain, tujuan, dan manfaat *game*, aplikasi *game* tajwid ini mendapat predikat tinggi sebanyak 100 %. Hasil ini, dapat disimpulkan bahwa belajar ilmu tajwid melalui *game* dapat diterapkan, lebih mudah dan mengasyikkan.



BAB V

PENUTUP

5.1 Kesimpulan

Game pengenalan tajwid ini berbentuk *side scrolling* dengan *genre* petualangan yang didalamnya terdapat edukasi yaitu untuk mengenalkan bagian ilmu tajwid. Materi ilmu tajwid yang ada dalam *game* yakni hukum nun sukun dan tanwin meliputi : *idgham bighunnah, idgham bilaghunnah, iqlab, izh-har, dan ikhfa'*. Konten edukasi ilmu tajwid diterapkan dengan cara mengenalkan masing-masing huruf tiap bacaan dengan menyebarkan huruf-huruf tersebut di area permainan untuk ditemukan dan menjawab beberapa pertanyaan. Pemain memperoleh satu contoh bacaan bila mampu menjalankan misi yang ada di setiap permainan tersebut.

Algoritma *Iterative Deepening A** diterapkan pada arah gerak musuh. Dari hasil uji coba yang dilakukan, algoritma ini dapat digunakan untuk menentukan jalur terpendek musuh untuk mengejar *player* (pemain) dengan waktu yang sangat cepat. Dari rata-rata hasil uji, penyelesaian dengan menggunakan algoritma ini membutuhkan sekitar 0.31 detik. Berdasarkan hasil uji coba kepada 30 responden tentang fitur, desain, tujuan, dan manfaat *game*, aplikasi *game* tajwid ini mendapat predikat tinggi sebanyak 100 %.

5.2 Saran

Beberapa saran untuk pengembangan selanjutnya, antara lain :Adanya perbaikan dalam penggunaan kode program agar lancar bila dimainkan dengan spesifikasi RAM *mobile* yang rendah.

1. Adanya variasi misi disetiap level agar tidak monoton.
2. Penambahan materi tajwid yang tidak hanya hukum nun sukun dan tanwin
3. Pengembangan *multiplatform* agar dapat dimainkan diberbagai sistem operasi *mobile*.
4. Adanya penambahan metode untuk mengatur perilaku musuh atau *player*.

DAFTAR PUSTAKA

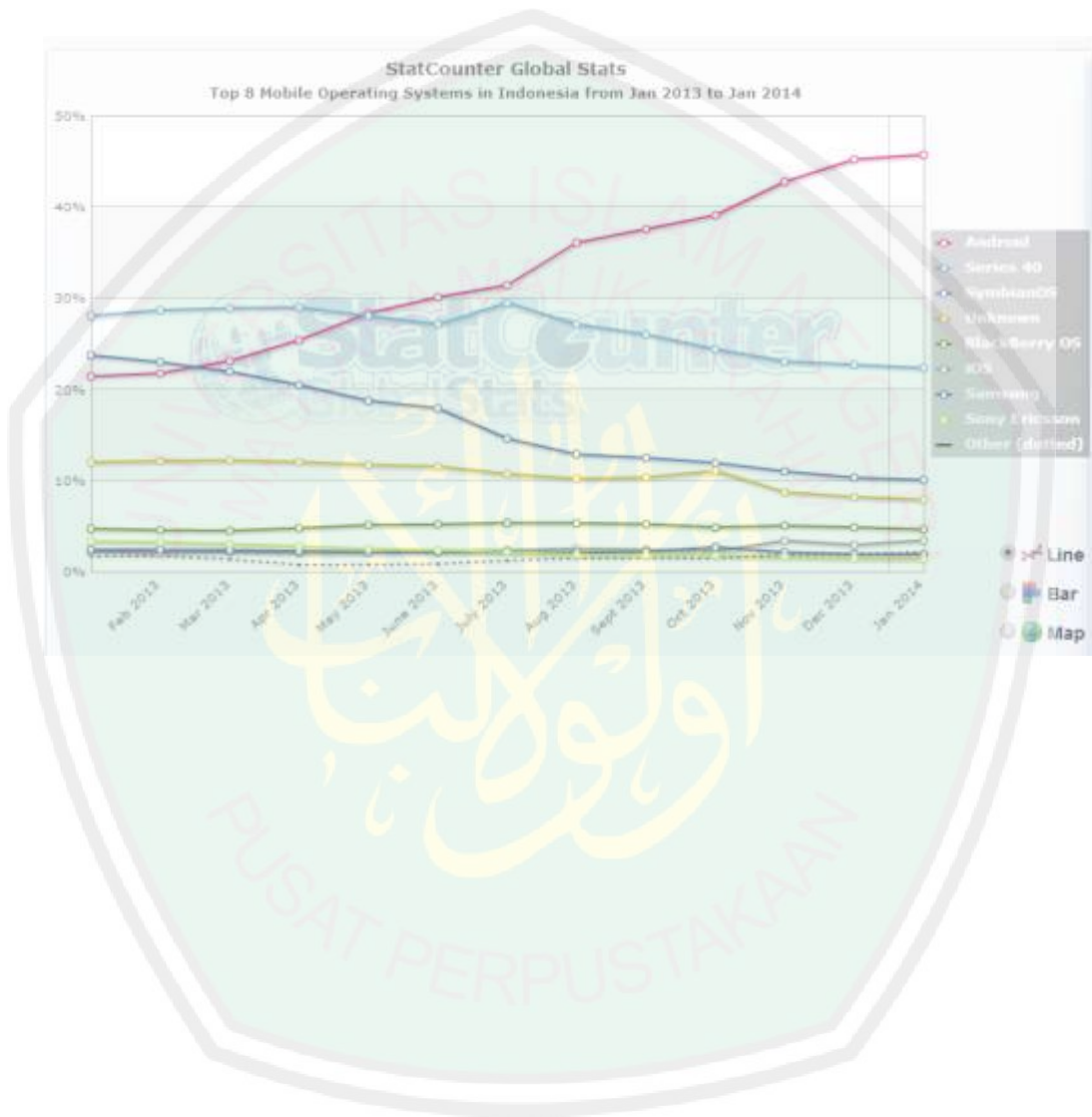
- Aeni, W. A. (2009, Agustus 20). *m-edukasi*. Retrieved September 20, 2013, from m-edukasi.kemdikbud.go.id: <http://m-edukasi.kemdikbud.go.id/artikel-mobile-learning-isi.php?kodenya=2009-ac>
- Amit. (2012, October 12). *Heuristics From Amit's Thoughts on Pathfinding*. Retrieved Oktober 23, 2013, from stanford University Web Site: <http://theory.stanford.edu/~amitp/GameProgramming/Heuristics.html>
- Arif, Y. M. (2010). *Strategi Menyerang Pada Game FPS Menggunakan Hierarchy Finite State Machine dan Logika Fuzzy*. Master Thesis ITS, Electrical Engineering.
- Counter, S. (2014, Januari 31). *StatCounter Global Stats Indonesia*. Retrieved Februari 01, 2014, from StatCounter: http://gs.statcounter.com/#mobile_os-ID-monthly-201301-201401
- Hidayat, R. R. (2013). *Membangun Game Side Scrolling 2D Petualangan Kabayan Menyelamatkan Makanan Tradisional*. Bandung: UNIKOM.
- Julian Togelius, S. K. (2009). The 2009 Mario AI Competition. *IEEE*.
- Kurkovsky, S. (2009). Experimenting with IDA* search algorithm. *Springer Science+Business Media B*, 277-286.
- Matthews, J. (2004). Basic A* Pathfinding. In S. Rabin, *AI Game Programming WISDOM* (pp. 105-113). USA: Charles River Media.
- Safaat, N. (2011). *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Informatika.
- Suyanto, S. (2011). *Artificial Intelligence*. Bandung: Informatika.
- Wahyudi, M. (2007). Ilmu Tajwid Plus. In M. Wahyudi, *Ilmu Tajwid Plus* (p. 4). Surabaya: Halim Jaya.
- Yofid I, H. M. (2011). Simulasi Multi-agent dengan Hexagonal Grid. *Seminar Nasional Pascasarjana XI*.
- Zaky, A. (2008). *Tuntunan Tahsin & Kaidah Tajwid*. Semarang: Pustaka Zaky & Wimba.
- Zechner, M. (2011). *Beginning Android Games*. New York: Apress.



LAMPIRAN-LAMPIRAN

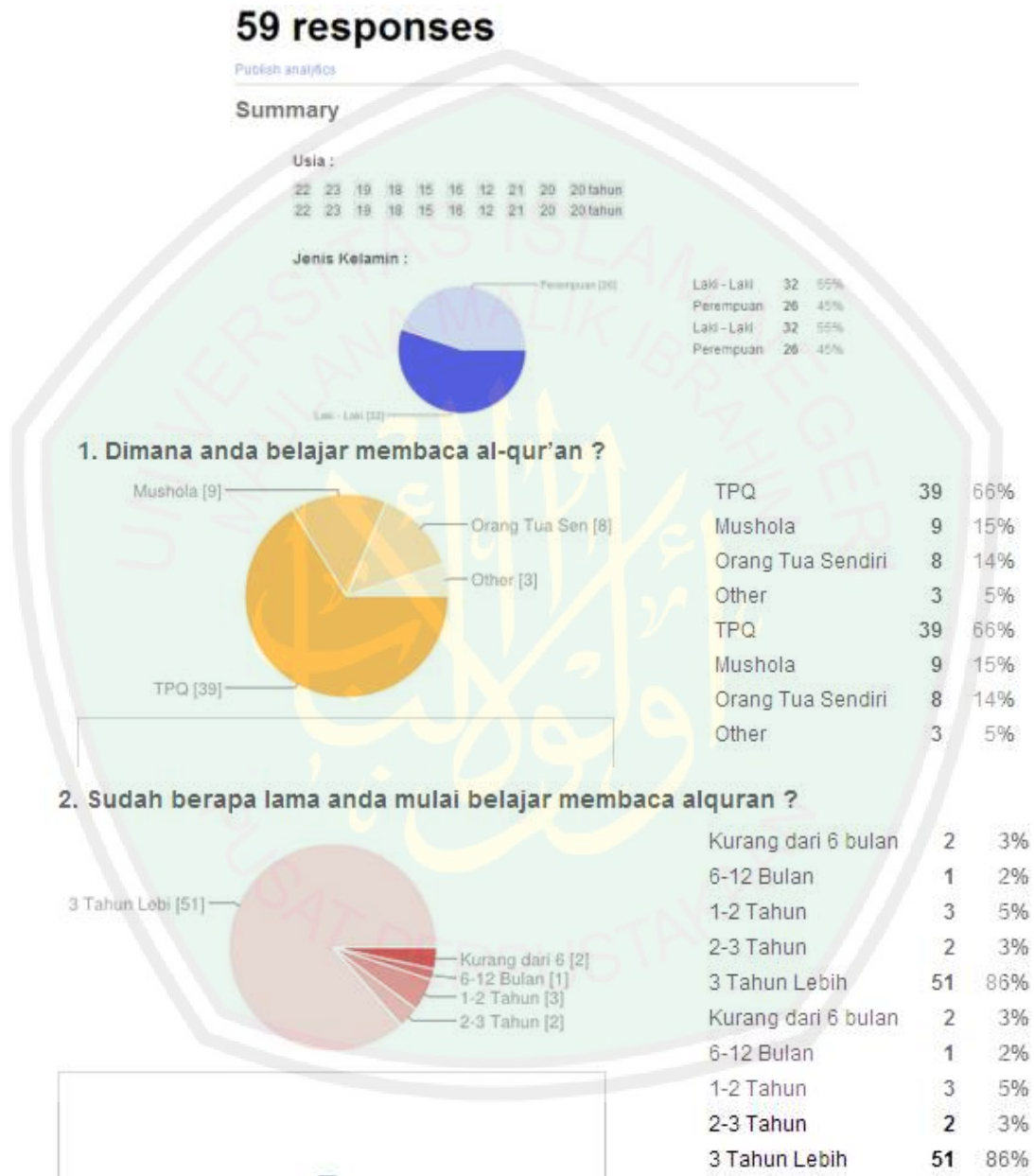
LAMPIRAN 1

Data statistik pengguna sistem operasi *mobile* di Indonesia bulan januari 2013 – januari 2014



LAMPIRAN 2

Hasil survei awal penelitian pembuatan game tajwid



LAMPIRAN 3

Kuisisioner untuk ujicoba aplikasi bagi pengguna

KUISISIONER

GAME PETUALANGAN UNTUK MENGENAL ILMU TAJWID

Petunjuk :

1. Isi nama dan usia anda
2. Baca dan pahami pernyataan-pernyataan di bawah dengan teliti
3. Berilah skor antara 1 – 9 pada kotak setelah pernyataan.

Nama :

Usia : Tahun

1. Saya tahu cara memainkan game tajwid ini
2. Game tajwid ini menarik dan menyenangkan
3. Game ini membantuku untuk mengenal ilmu tajwid
4. Suara contoh bacaan memudahkan untuk belajar ilmu tajwid
5. Tombol yang ada di game ini cukup nyaman
6. Materi ilmu tajwid yang disampaikan bisa saya pahami
7. Saya tidak mengalami kesulitan dalam menjawab soal
8. Tampilan gambar dalam game ini menarik

LAMPIRAN 4

Data responden yang melakukan ujicoba terhadap aplikasi

No	Nama	Usia (tahun)
1	Leli	11
2	Dinda	12
3	Rifqul	16
4	Firda	10
5	Levi	10
6	Sukma	12
7	Lail	12
8	badruz	10
9	amel	13
10	alif	11
11	alan	10
12	ovi	16
13	ayil	16
14	aron	15
15	said	14
16	syahril	15
17	azam	15
18	andy	13
19	rizal	14
20	syahid	13
21	ahmad	12
22	kamil	14
23	roni	15
24	lia	14
25	rudi	12
26	rega	11
27	toni	17
28	bayu	15
29	farah	16
30	atik	13