

**PENERAPAN ALGORITMA A*(A-Star) UNTUK PENENTUAN
RUTE TERPENDEK GAME PRAMUKA
BERBASIS ANDROID**

SKRIPSI

Oleh:

NAUFAL WAFIQUURRAHMAN

NIM. 10650110



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

**PENERAPAN ALGORITMA A*(A-Star) UNTUK PENENTUAN
RUTE TERPENDEK GAME PRAMUKA
BERBASIS ANDROID**

SKRIPSI

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

**Oleh:
NAUFAL WAFIQUURRAHMAN
NIM. 10650110**

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

HALAMAN PERSETUJUAN

**PENERAPAN ALGORITMA A*(A-Star) UNTUK PENENTUAN
RUTE TERPENDEK GAME PRAMUKA
BERBASIS ANDROID**

SKRIPSI

Oleh :

Nama : Naufal Wafiqurrahman
NIM : 10650110
Jurusan : Teknik Informatika
Fakultas : Sains Dan Teknologi

Telah Disetujui, 12 Januari 2015

Dosen Pembimbing I

Dosen Pembimbing II

Hani Nurhayati, M.T

Fachrul Kurniawan, M.MT

NIP. 19780625 200801 2 006

NIP. 19771020 200912 1 001

Mengetahui,

Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdian

NIP. 19740424 200901 1 008

HALAMAN PENGESAHAN

**PENERAPAN ALGORITMA A*(A-Star) UNTUK PENENTUAN
RUTE TERPENDEK GAME PRAMUKA
BERBASIS ANDROID**

SKRIPSI

Oleh :

Naufal Wafiqurrahman
NIM. 10650110

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

Tanggal 08 Januari 2015

Susunan Dewan Penguji:

Tanda Tangan

- | | | | |
|--------------------|---|---|---|
| 1. Penguji Utama | : <u>Yunifa Miftachul Arif, M.T</u>
NIP. 19830616 201101 1 004 | (|) |
| 2. Ketua Penguji | : <u>Fressy Nugroho, M.T</u>
NIP. 19710722 201101 1 001 | (|) |
| 3. Sekretaris | : <u>Hani Nurhayati, M.T</u>
NIP. 19780625 200801 2 006 | (|) |
| 4. Anggota Penguji | : <u>Fachrul Kurniawan, M.MT</u>
NIP. 19771020 200901 1 001 | (|) |

Mengetahui,
Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdiyan
NIP. 19740424 200901 1 008

HALAMAN PERSEMBAHAN

Dengan rasa syukur seraya mengharap ridho ilahi

Kupersembahkan karya ini kepada :

Bapak, ibu dan adik tercinta:

H.Kamaluddin, Hj.Nafsiah

(Alm.) M.Rizal Ramadhani dan M.Zulfikar Akbar

Atas segalanya.

KEPADA SEGENAP KELUARGA BESAR MBAH H.CHANDIR SHIRADJ DAN KELUARGA BESAR MBAH ABDULLAH, YANG TELAH MEMBERIKAN DUKUNGAN KEPADA PENULIS UNTUK DAPAT MENYELESAIKAN SKRIPSI INI.

KEPADA SAHABAT “PLETOG” RISTIAN HARDIANAS, ERMA MUSTIKAYANI, FANI IKHA ANGGRAINI, M.IRFAN, LAILA FITROTUZ ZAHROH, MOCH FAHRI, SONI ARIFIN, SHOFWAN YUSUF, MUNA LAILAL HANA, DAN SHEILA ROSEYLA DEWI.

KEPADA SAHABAT “MMM FAMILY” ALDITA RAHMA, PUSPA SAFITRI, DEWI CHUM, PUSPITA DEWI, ADE DUROTUN, GERY ARIES, SYFEI KARIM, NOVIANTO, VIVID ROHMANIYAH, ZAENAL ABIDIN, MOH.DZIKRULLAH, ALIF HARIS, AGUS FADLUN DAN JUNIARDI.

KEPADA SAHABAT “SARJANA KUCLUK” : ALFIAN, GUS MAIMUN, ALPAN, CINAP, RONI, AHONG, ARIES, BANG SYUKRON, UWAIS, SADAD, BAHTIAR, SYAHID, FIRMAN, AGUS CAHYONO DAN SA’AD.

TEMAN-TEMAN “PUTRA-PUTRI KARTINI JEPARA MALANG (PAPIKAMA)”: AMANG, WULAN, LILY, CAK ANAM, CAK BUDI DAN CAK LUBAB

TEMAN-TEMAN “PPSS NURUL HUDA ONOSOGREM MALANG”, “CREW SETUNGGAL COFFEE”, “KOST 672C”, “4GTI” DAN “INFINITY’10” YANG TIDAK BISA SAYA SEBUTKAN SATU PERSATU.

TERIMA KASIH ATAS PELAJARAN YANG TELAH DIBERIKAN, BAIK ITU SECARA LANGSUNG MAUPUN TIDAK LANGSUNG. SEMOGA ALLAH SWT MELINDUNGI, MENYAYANGI DAN MENEMPATKAN MEREKA SEMUANYA PADA SURGANYA KELAK DAN MELIMPahkan REZEKI KEPADA MEREKA SEMUA...

Motto :

Kejarlah apa yang bermanfaat bagimu,
dan mintalah pertolongan hanya kepada Allah,
jangan mudah menyerah dan jangan pernah
berkata

“Kalau saja aku melakukan begini, pasti akan
jadi begini,”

Tapi katakanlah “Allah telah mentakdirkan dan
apa yang
Dia kehendaki pasti akan Dia lakukan”

HALAMAN PERNYATAAN
ORISINALITAS PENELITIAN

Saya yang bertandatangan di bawah ini:

Nama : Naufal Wafiqurrahman
NIM : 10650110
Fakultas/Jurusan : Sains Dan Teknologi / Teknik Informatika
Judul Penelitian : Penerapan Algoritma A*(A-Star) untuk penentuan rute terpendek game pramuka berbasis Android

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

Malang, 12 Januari 2015

Yang Membuat Pernyataan,

Naufal Wafiqurrahman

10650110

KATA PENGANTAR



Segala puji bagi Allah SWT yang telah melimpahkan rahmat serta karuniaNya kepada penulis sehingga bisa menyelesaikan skripsi dengan judul “ Penerapan Algoritma A*(A-Star) untuk penentuan rute terpendek game pramuka berbasis Android” dengan baik.

Shalawat serta salam semoga tercurah kepada Nabi Agung Muhammad SAW yang telah membimbing umatnya dari gelapnya kekufuran menuju cahaya Islam yang terang benderang.

Penulis menyadari keterbatasan pengetahuan yang penulis miliki, karena itu tanpa keterlibatan dan sumbangsih dari berbagai pihak, sulit bagi penulis untuk menyelesaikan skripsi ini. Maka dari itu dengan segenap kerendahan hati patutlah penulis ucapkan terima kasih kepada:

1. Bapak dan Ibu yang selalu memotivasi dan selalu mendoakan yang terbaik dalam pengerjaan skripsi ini.
2. Hani Nurhayati, M.T, selaku dosen pembimbing I yang telah meluangkan waktu untuk membimbing, memotivasi, mengarahkan dan memberi masukan dalam pengerjaan skripsi ini.
3. Fachrul Kurniawan, M.MT, selaku dosen pembimbing II, yang selalu memberikan masukan, nasehat serta petunjuk dalam penyusunan laporan skripsi ini.
4. Dr. Cahyo Crys dian, selaku Ketua Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang, yang mendukung dan mengarahkan dalam pengerjaan skripsi ini.

5. Segenap Dosen Teknik informatika yang telah memberikan bimbingan keilmuan kepada penulis selama masa studi.
6. Semua pihak yang tidak mungkin penulis sebutkan satu-persatu, atas segala yang telah diberikan kepada penulis dan dapat menjadi pelajaran.

Sebagai penutup, penulis menyadari dalam skripsi ini masih banyak kekurangan dan jauh dari sempurna. Semoga apa yang menjadi kekurangan bisa disempurnakan oleh peneliti selanjutnya. Apa yang menjadi harapan penulis, semoga karya ini bermanfaat bagi kita semua. Amin.

Malang, 12 Januari 2015

Penulis

DAFTAR ISI

HALAMAN PENGAJUAN.....	ii
HALAMAN PERSETUJUAN.....	iii
HALAMAN PENGESAHAN.....	iv
HALAMAN PERSEMBAHAN.....	v
HALAMAN MOTTO.....	vi
HALAMAN PERNYATAAN.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xii
DAFTAR TABEL.....	xiii
ABSTRAK BAHASA INDONESIA.....	xiv
ABSTRAK BAHASA INGGRIS.....	xv
ABSTRAK BAHASA ARAB.....	xvi
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
BAB II Landasan Teori.....	7
2.1 Kajian Pustaka.....	7
2.1.1 Permainan (Game).....	12
2.1.2 Komponen game	17

2.1.3 Platfrom Android	20
2.1.4 AndEngine	25
2.1.5 Path Finding	28
2.1.6 Kecerdasan Buatan.....	27
2.1.7 Algoritma A*(AStar).....	31
2.2 Penelitian Terkait.....	33
2.3 Metode Penelitian.....	35
BAB III ANALISA DAN PERANCANGAN SISTEM.....	35
3.1 Deskripsi Sistem.....	35
3.2 Story Line	35
3.3 Story Board.....	35
3.4 Konten Aplikasi.....	37
3.5 Skenario Game	38
3.6 Penerapan Metode A*(A-Star).....	40
BAB IV HASIL DAN PEMBAHASAN.....	56
4.1 Implementasi.....	56
4.1.1 Perangkat Yang Digunakan	56
4.2 Implementasi Aplikasi.....	57
4.3 Pengujian Algoritma A*.....	61
4.4 Implementasi Algoritma A*(A-Star).....	63
4.5 Uji Coba Pada Device.....	64
4.6 Integrasi Game Pramuka Dengan Islam.....	66

BAB V PENUTUP	71
5.1 Kesimpulan.....	71
5.2 Saran.....	71
DAFTAR PUSTAKA.....	73



DAFTAR GAMBAR

Gambar 3.1 Sketsa Peta.....	36
Gambar 3.2 Sketsa pos.....	36
Gambar 3.3 FSM level 1.....	38
Gambar 3.4 FSM level 2.....	39
Gambar 3.5 Flowchart A*.....	42
Gambar 3.6 Posisi awal musuh.....	43
Gambar 3.7 Pergerakan musuh 1.....	48
Gambar 3.8 Pergerakan musuh 2.....	51
Gambar 3.9 Hasil Akhir pergerakan musuh.....	55
Gambar 4.1 Splash Screen.....	57
Gambar 4.2 Menu utama.....	58
Gambar 4.3 Tampilan Level 1.....	59
Gambar 4.4 Tampilan Level 2.....	59
Gambar 4.5 Sourcode Ujicoba.....	62

DAFTAR TABEL

Tabel 4.1 Hasil ujicoba pada Netbeans.....	62
Tabel 4.2 Uji Coba pada device.....	64
Tabel 4.3 Presentase hasil ujicoba.....	66



ABSTRAK

Wafiqurrahman, Naufal. 2014. PENERAPAN ALGORITMA A*(A-Star) UNTUK PENENTUAN RUTE TERPENDEK PADA GAME PRAMUKA BERBASIS ANDROID. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing : (I) Hani Nurhayati, M.T (II) Fachrul Kurniawan, M.MT

Kata Kunci : *Astar, Non Player Character, Pramuka, Path finding, Android*

Pendidikan kepramukaan adalah proses pendidikan yang praktis, di luar lingkungan sekolah dan di luar lingkungan keluarga yang dilakukan di alam terbuka dalam bentuk kegiatan yang menarik, menantang, menyenangkan, sehat, teratur dan terarah dengan menerapkan Prinsip Dasar kepramukaan dan Metode Kepramukaan, yang sasaran akhirnya adalah terbentuknya watak kepribadian dan akhlak mulia. Implementasi dari akhlak mulia dalam pramuka terdapat pada dasadharma pramuka. Dasadharma memuat pokok-pokok moral yang harus ditanamkan kepada anggota Pramuka agar mereka dapat berkembang menjadi manusia yang memiliki watak yang baik, berbudi tinggi, berakhlak mulia dan menjadi warga negara Republik Indonesia yang setia. Game pramuka ini adalah sebuah permainan yang menganut tipe permainan adventure yaitu sebuah game petualangan. Musuh yang datang selalu menggunakan lintasan jalur terpendek untuk mencapai player. Jika Non Player Character sudah mencapai posisi target maka permainan berakhir, player harus sebisa mungkin menghindari enemy yang akan menyerangnya. Ada beberapa metode untuk path finding. Salah satunya adalah A*(A-Star), metode ini diterapkan untuk kecerdasan Non Player Character dalam pencarian jalur terpendek. Pengujian dilakukan pada perangkat mobile dengan menggunakan *platform* Android.

ABSTRACT

Wafiqurrahman, Naufal. 2014. Implementation Algorithm A * (A-Star) Shortest Route For Determination Based On Android Scout Game. Skripsi. Departement Of Informatic Engineering Faculty Of Science And Technology Maulana Malik Ibrahim State Islamic University Of Malang. Adviser : (I) Hani Nurhayati, M.T (II) Fachrul Kurniawan, M.MT.

Keywords : *Astar, Non Player Character, Scout, Path finding, Android*

Scouting Education is a process of practical education, outside the school and outside the family who performed in the open air in the form of activity that is interesting, challenging, fun, healthy, organized and directed by implementing the Basic Principles and Methods of Scouting. the ultimate goal is the creation of the character's personality and noble character. Implementation of noble character in there on dasadharna scout scouts. Dasadharna load moral points that must be imparted to scouts so that they can develop into a human who has a good character, virtuous, noble and become a citizen of the Republic of Indonesia are loyal. This scout game is a game that embraces type adventure game is an adventure game. Enemies who come always use the shortest path trajectory to reach the player. If the Non-Player Character has reached the target position, the game is over, the player should as much as possible to avoid enemy to be attacked. There are several methods for path finding. One is A * (A-Star), this method is applied to the intelligence of the Non-Player Character in a shortest path search. Tests performed on mobile devices using the Android platform.

الخلاصة

وفق الرحمان نوفل. ٢٠١٥. تطبيق خوارزمية أقصر الطرق لتحديد القائمين على الروبوت. المقالة. قسم المعلوماتية في كلية العلوم والتكنولوجيا في جامعة الحكومية الإسلامية مولانا مالك إبراهيم مالانج. المشرف ١. هاني نور حاباتي ٢. مخزل كورنيوان

كلمات البحث : أستار، غير لاعب الأحرف، الكشفية وتقصي المسار،

الكشفية التعليم هو عملية التعليم العملي، خارج المدرسة وخارج نطاق الأسرة الذي أجرى في الهواء الطلق في شكل من أشكال النشاط التي هي مثيرة للاهتمام، مما يشكل تحديا، والمرح، وصحية، نظمت وإخراج تنفيذ المبادئ الأساسية والأساليب الكشفية الكشفية، وهذا هو الهدف النهائي تشكيل شخصية الطابع والطابع النبيل. تنفيذ الطابع النبيل في هناك يوم الكشفية الكشفية. تحميل نقاط الأخلاقية التي يجب الكشف عنها للكشفية بحيث يمكن أن تتطور إلى الإنسان الذي لديه شخصية جيدة، الفاضلة، نبيلة ويصبح مواطنا في جمهورية اندونيسيا هم من الموالين. هذه اللعبة هي لعبة الكشفية الذي يحتضن المباراة نوع المغامرة هي لعبة المغامرة. أعداء الذين يأتون دائما استخدام أقصر مسار الطريق للوصول إلى لاعب. إذا وصلت الأحرف غير لاعب موقف الهدف، ان اللعبة قد انتهت، لاعب يجب قدر الإمكان لتجنب العدو أن تتعرض للهجوم. هناك عدة طرق لاكتشاف المسار يتم تطبيق هذه الطريقة لذكاء الأحرف غير لاعب في بحث مسار أقصر. الاختبارات التي أجريت على الأجهزة النقالة باستخدام منصة أندرويد.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Gerakan Pramuka Indonesia adalah nama organisasi pendidikan nonformal yang menyelenggarakan pendidikan kependuan yang dilaksanakan di Indonesia. Kata Pramuka merupakan singkatan dari Praja Muda Karana yang artinya pemuda yang suka berkarya. Pramuka mempunyai kegiatan yang dinamakan kepramukaan. Pendidikan kepramukaan adalah proses pendidikan yang praktis, di luar lingkungan sekolah dan di luar lingkungan keluarga yang dilakukan di alam terbuka dalam bentuk kegiatan yang menarik, menantang, menyenangkan, sehat, teratur dan terarah dengan menerapkan Prinsip Dasar kepramukaan dan Metode Kepramukaan, yang sasaran akhirnya adalah terbentuknya watak kepribadian dan akhlak mulia.

Implementasi dari akhlak mulia dalam pramuka terdapat pada dasadharma pramuka. Dasadharma memuat pokok-pokok moral yang harus ditanamkan kepada anggota Pramuka agar mereka dapat berkembang menjadi manusia yang memiliki watak yang baik, berbudi tinggi, berakhlak mulia dan menjadi warga negara Republik Indonesia yang setia.

Kegiatan kepramukaan ada bermacam-macam antara lain adalah perkemahan dan jelajah alam. Dengan berkemah yang diadakan di alam terbuka maka seseorang bisa merasakan berbagai cuaca, hujan, panas atau angin yang bertiup kencang dan ketika malam tiba ditemani suara serangga, burung atau binatang lainnya hal ini dapat menumbuhkan rasa natural

seseorang. Ketika seseorang memperhatikan sungai atau air terjun, maka di sana ada sebuah ritme, kesejukan, dan keindahan yang menciptakan suasana hati yang sempurna. Memperhatikan indahnya alam seperti ini akan berbuah syukur pada ciptaan Allah. Rasa syukur ini akan memupuk cinta kita kepada alam dan meningkatkan rasa sayang terhadap sesama. Allah berfirman dalam surat Ali ‘Imran ayat 190.

QS Ali Imran 190 yang berbunyi:

إِنِّ فِي خَلْقِ السَّمَوَاتِ وَالْأَرْضِ وَأَخْتِلَافِ اللَّيْلِ وَالنَّهَارِ لآيَاتٍ
لِّأُولِي الْأَلْبَابِ ﴿١٩٠﴾

Yang mempunyai arti: “Sesungguhnya dalam penciptaan langit dan bumi, dan silih bergantinya malam dan siang terdapat tanda-tanda bagi orang-orang yang berakal.

Menurut *Kakwarcab (Ketua Kwartir Cabang)* Pangkal Pinang Suparyono, Kegiatan kepramukaan di zaman modern ini mulai ditinggalkan oleh anak-anak usia sekolah, hal ini dikarenakan kegiatan kepramukaan hanya sebagai ekstrakurikuler bukan sebagai mata pelajaran wajib di sekolah. Tidak hanya penggunaan seragam pramuka di hari jum’at dan sabtu tanpa menjiwai dan mengikuti pelatihan pramuka itu sendiri. Dalam perkembangannya mulai dari kwartir nasional (tingkat nasional) samapai dengan kwartir ranting (tingkat kecamatan) mulai mencari solusi agar minat siswa terhadap pramuka meningkat, tidak hanya dilihat dari segi kuantitas tetapi juga dari segi kualitas. (Radar bangka, 2011)

Sebuah *game* dalam masa kini sering dipandang sebelah mata oleh para orang tua. Mengapa demikian ?. Hal ini karena *game* saat ini dianggap kurang memberikan kontribusi untuk pembelajaran anak-anak. Bermain *game* dipandang oleh orang tua sebagai pekerjaan yang tidak bermanfaat dan hanya membuang-buang uang saja. Berdasarkan pengalaman disekitar penulis, bermain *game* memang membuat orang ketagihan dan rela membuang-buang uang untuk hal itu.

Berkembangnya teknologi yang sangat pesat seperti saat ini menjadikan banyak sekali *game* yang menggunakan teknologi modern. Salah satu contohnya adalah *smartphone*. *Smartphone* pada saat ini sudah menjadi kebutuhan penting bagi masyarakat untuk menunjang aktifitas.

Dari sanalah muncul kepedulian dan tercetus ide untuk membuat *game* jelajah alam pramuka menjadi permainan pendidikan. Permainan pendidikan adalah permainan yang dirancang untuk pengayaan pendidikan (mendukung pengajaran dan pembelajaran) menggunakan teknologi multimedia interaktif (Higgins, dkk, 2010). *Game* edukasi unggul dalam beberapa aspek jika dibandingkan dengan metode pembelajaran konvensional. Salah satu keunggulan yang signifikan adalah adanya animasi yang dapat meningkatkan daya ingat sehingga anak dapat menyimpan materi pelajaran dalam waktu yang lebih lama dibandingkan dengan metode pengajaran konvensional. Permainan pramuka ini akan dibuat menjadi permainan berbasis *smartphone*. Keunggulan dari permainan berbasis *smartphone* ini adalah dapat dimainkan kapan saja dan dimana saja. Sistem operasi android adalah sistem operasi

yang bersifat terbuka dan para developer dapat berkesempatan mengembangkan aplikasi-aplikasinya sesuai keinginannya sendiri. Pada penelitian ini, penulis ingin mencoba membuat permainan bernama *Game Pramuka*.

Permainan *Game Pramuka* ini merupakan sebuah permainan bergenre Adventure tetapi akan dimodifikasi menjadi sebuah *game* edukasi untuk pembelajaran materi-materi Pramuka. Dalam *game* ini terdapat pos sebagai titik aman, setelah sampai titik aman maka muncul pertanyaan tentang kepramukaan, dalam perjalanan menuju pos akan ada gangguan dari seekor kera yang akan mengejar *Player*. Bagaimana musuh bisa mengejar *player* secara otomatis dan mencari posisi *Player*?. Maka musuh tersebut harus memiliki kecerdasan untuk mencari jalur agar sampai pada posisi *player*. Pencarian jalur oleh musuh inilah yang akan mengaplikasikan algoritma A*(A-Star).

Algoritma A* (disebut “A-star”) adalah sebuah *graph* atau metode *tree search* yang digunakan untuk mencari jalan dari sebuah node awal ke node tujuan (*goal node*) yang telah ditentukan, metode ini menggunakan “estimasi *heuristic*” $h(x)$ pada setiap node untuk mengurutkan setiap node x berdasarkan estimasi rute terbaik yang melalui node tersebut. Dalam prosesnya metode ini akan mengunjungi setiap node berdasarkan urutan yang dihasilkan dari estimasi *heuristic* ini.

Menurut *Munir (2009)*, Metode A* tanpa fungsi *heuristic* yang baik akan memperlambat pencarian dan dapat menghasilkan rute yang tidak tepat.

Fungsi *heuristic* yang sempurna akan membuat metode A* langsung menuju final node tanpa harus mencari kearah lain. Sehingga jika fungsi *heuristic*-nya terlalu *underestimate* akan menyebabkan algoritma ini beranggapan bahwa ada rute lain yang lebih baik.

Dalam masalah pencarian rute di mana metode A* sering digunakan, A* secara bertahap membangun semua rute yang mengarah mulai dari titik awal sampai akhirnya mencapai titik akhir. Metode A* hanya membangun rute yang mungkin digunakan untuk mencapai tujuan. Untuk mengetahui rute mana yang memungkinkan mengarah ke titik akhir, A* menggunakan estimasi *heuristic* jarak dari sembarang node ke node tujuan. Dalam kasus pencarian rute, ini bisa jadi sama dengan jarak lurus antara dua titik, di mana biasanya merupakan perkiraan dari jarak jalan.

Aplikasi permainan ini akan diimplementasikan pada Android OS mobile dengan harapan dapat dimainkan dimana saja dan kapan saja.

1.2 Rumusan Masalah

Bagaimana mengimplementasikan algoritma A*(A-Star) untuk penentuan arah gerak musuh pada *game* pramuka.

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Aplikasi dibangun berbasis *mobile* 2 Dimensi
2. *Game* yang dibuat adalah single player
3. Konten pramuka adalah materi pramuka untuk tingkatan siaga dan penggalang

1.4 Tujuan Penelitian

Dapat mengimplementasikan Algoritma A*(A-Star) kedalam aplikasi permainan *game* pramuka untuk pembelajaran tingkat siaga dan penggalang.

1.5 Manfaat Penelitian

Dari penelitian ini diharapkan dapat memberikan manfaat terhadap kepramukaan dan perkembangan *game* di Indonesia, antara lain:

1. Meningkatkan kecintaan terhadap pramuka.
2. Membuat media pembelajaran berbasis *game* yang menarik untuk siswa tingkat siaga dan penggalang.

BAB II

LANDASAN TEORI

2.1 Kajian Pustaka

2.1.1 Permainan (*Game*)

Game merupakan kata dalam bahasa Inggris yang berarti permainan. Permainan adalah sesuatu yang dapat dimainkan dengan aturan tertentu sehingga ada yang menang dan ada yang kalah, biasanya dalam konteks tidak serius atau bertujuan refreshing. Teori permainan pertama kali ditemukan oleh sekelompok ahli Matematika pada tahun 1944. Teori itu dikemukakan oleh John von Neumann and Oskar Morgenstern. Sebuah permainan adalah sebuah sistem dimana pemain terlibat dalam konflik buatan, disini pemain berinteraksi dengan sistem dan konflik dalam permainan merupakan rekayasa atau buatan, dalam permainan terdapat peraturan yang bertujuan untuk membatasi perilaku pemain dan menentukan permainan. *Game* banyak disukai oleh anak – anak hingga orang dewasa. *Game* sebenarnya penting untuk perkembangan otak, untuk meningkatkan konsentrasi dan melatih untuk memecahkan masalah dengan tepat dan cepat karena dalam *game* terdapat berbagai konflik atau masalah yang menuntut kita untuk menyelesaikannya dengan cepat dan tepat. Tetapi *game* juga bisa merugikan karena apabila kita sudah kecanduan *game* kita akan lupa waktu dan akan mengganggu kegiatan atau aktifitas yang sedang kita lakukan. Permainan terdiri atas sekumpulan peraturan yang membangun situasi

bersaing dari dua sampai beberapa orang atau kelompok dengan memilih strategi yang dibangun untuk memaksimalkan kemenangan sendiri atau pun untuk meminimalkan kemenangan lawan. Peraturan-peraturan menentukan kemungkinan tindakan untuk setiap pemain, sejumlah keterangan diterima setiap pemain sebagai kemajuan bermain, dan sejumlah kemenangan atau kekalahan dalam berbagai situasi. (Neuman, 2007).

Pengertian *game* menurut beberapa ahli:

1. Menurut Clark C.Abt *Game* adalah kegiatan yang melibatkan keputusan pemain, berupaya mencapai tujuan dengan dibatasi oleh konteks tertentu. (Misal oleh aturan)
2. Menurut Greg Costikyan, *Game* adalah sebarang karya seni dimana peserta, yang disebut player, membuat keputusan untuk mengelola sumberdaya yang dimilikinya melalui benda didalam *game* dalam mencapai tujuan.

Game ditinjau dari platform atau alat yang digunakan :

1. *Arcade games*

Yaitu yang sering disebut ding-dong di Indonesia, biasanya berada di daerah / tempat khusus dan memiliki box atau mesin yang memang khusus di design untuk jenis video *games* tertentu dan tidak jarang bahkan memiliki fitur yang dapat membuat pemainnya lebih merasa “masuk” dan “menikmati”, seperti pistol, kursi khusus, sensor gerakan, sensor injakkan dan stir mobil (beserta transmisinya tentunya).

2. *PC Games*

Yaitu video *game* yang dimainkan menggunakan Personal Computers..

3. *Console games*

Yaitu video *games* yang dimainkan menggunakan console tertentu, seperti Playstation 2, Playstation 3, XBOX 360, dan Nintendo Wii..

4. *Handheld games*

Yaitu yang dimainkan di console khusus video *game* yang dapat dibawa kemana-mana, contoh Nintendo DS dan Sony PSP..

5. *Mobile games*

Yaitu yang dapat dimainkan atau khusus untuk mobile phone atau PDA..

Game ditinjau dari genre permainannya dapat dibagi menjadi beberapa pengklarifikasian. Diantaranya adalah:

1. *Fighting* (pertarungan) Ada yang mengelompokan video *game fighting* di bagian Aksi, namun penulis berpendapat berbeda, jenis ini memang memerlukan kecepatan refleks dan koordinasi mata-tangan, tetapi inti dari *game* ini adalah penguasaan jurus (hafal caranya dan lancar mengeksekusinya), pengenalan karakter dan timing sangatlah penting, o iya,

combo-pun menjadi esensial untuk mengalahkan lawan secepat mungkin. Dan berbeda seperti *game* Aksi pada umumnya yang umumnya hanya melawan Artificial Intelligence atau istilah umumnya *melawan komputer* saja, pemain jenis *fighting game* ini baru teruji kemampuan sesungguhnya dengan melawan pemain lainnya. *Seri Street Fighter, Tekken, Mortal Kombat, Soul Calibur dan King of Fighter* adalah contohnya.

2. *Adventure game* (Petualangan). Bedanya dengan jenis video *game* aksi-petualangan, refleks dan kelihaian pemain dalam bergerak, berlari, melompat hingga memecut atau menembak tidak diperlukan di sini. Video *Game* murni petualangan lebih menekankan pada jalan cerita dan kemampuan berpikir pemain dalam menganalisa tempat secara visual, memecahkan teka-teki maupun menyimpulkan rangkaian peristiwa dan percakapan karakter hingga penggunaan benda-benda tepat pada tempat yang tepat. Termasuk didalamnya:
 - a. *Petualangan dengan teks* atau sistem tunjuk dan klik, contoh: *Kings Quest, Space Quest, Heroes Quest, Monkey Island, Sam and Max,*
 - b. *Novel atau film interaktif*, seperti *game “dating”* yang banyak beredar di Jepang, *Dragons Lair dan Night Trap*
3. *Role Playing*. Video *game* jenis ini sesuai dengan terjemahannya, bermain peran, memiliki penekanan pada tokoh/peran perwakilan pemain di dalam

permainan, yang biasanya adalah tokoh utamanya, dimana seiring kita memainkannya, karakter tersebut dapat berubah dan berkembang ke arah yang diinginkan pemain (biasanya menjadi semakin hebat, semakin kuat, semakin berpengaruh, dll) dalam berbagai parameter yang biasanya ditentukan dengan naiknya level, baik dari status kepintaran, kecepatan dan kekuatan karakter, senjata yang semakin sakti, ataupun jumlah teman maupun mahluk peliharaan. Secara kebudayaan, pengembang *game* Jepang biasanya membuat *Role Playing Game (RPG)* ke arah cerita linear yang diarahkan seolah karakter kita adalah tokoh dalam cerita itu, seperti *Final Fantasy*, *Dragon Quest* dan *Xenogears*. Sedangkan pengembang *game* RPG Eropa, cenderung membuat karakter kita bebas memilih jalan cerita sendiri *secara non-linear*, seperti *Ultima*, *Never Winter Nights*, *baldurs gate*, *Elder Scroll*, dan *Fallout*.

4. *Puzzle*. Video *game* jenis ini sesuai namanya berintikan mengenai pemecahan teka-teki, baik itu menyusun balok, menyamakan warna bola, memecahkan perhitungan matematika, melewati labirin, sampai mendorong-dorong kota masuk ke tempat yang seharusnya, itu semua termasuk dalam jenis ini. Sering pula permainan jenis ini adalah juga unsur permainan dalam video *game* petualangan maupun *game* edukasi. *Tetris*, *Minesweeper*, *Bejeweled*, *Sokoban* dan *Bomberman*.
5. *Olahraga*. Singkat padat jelas, bermain sport di PC atau konsol anda. Biasanya permainannya diusahakan serealistik mungkin walau kadang ada

yang menambah unsur fiksi seperti NBA JAM. Contohnya pun jelas, Seri *Winning Eleven*, seri *NBA*, seri *FIFA*

6. Aksi – Petualangan. Memasuki gua bawah tanah, melompati bebatuan di antara lahar, bergelayutan dari pohon satu ke pohon lain, bergulat dengan ular sambil mencari kunci untuk membuka pintu kuil legendaris, atau sekedar mencari telepon umum untuk mendapatkan misi berikutnya, itulah beberapa dari banyak hal yang karakter pemain harus lakukan dan lalui dalam video *game* jenis ini. Menurut penulis, *game* jenis ini sudah berkembang jauh hingga menjadi genre campuran *action beat-em up* juga, dan sekarang, di tahun 2000 an, jenis ini cenderung untuk memiliki visual 3D dan sudut pandang orang ke-tiga. *Prince Tomb of Rider*, *Grand Theft Auto* dan *Persia* termasuk didalamnya
7. Simulasi, Konstruksi dan manajemen. Video *Game* jenis ini seringkali menggambarkan dunia di dalamnya sedekat mungkin dengan dunia nyata dan memperhatikan dengan detil berbagai faktor. Dari mencari jodoh dan pekerjaan, membangun rumah, gedung hingga kota, mengatur pajak dan dana kota hingga keputusan memecat atau menambah karyawan. Dunia kehidupan rumah tangga sampai bisnis membangun konglomerasi, dari jualan limun pinggir jalan hingga membangun laboratorium cloning. Video *Game* jenis ini membuat pemain harus berpikir untuk mendirikan, membangun dan mengatasi masalah dengan menggunakan dana yang terbatas. Contoh: *Sim City*, *The Sims*, *Tamagotchi*

8. Strategi. Kebalikan dari video *game* jenis action yang berjalan cepat dan perlu refleks secepat kilat, video *game* jenis strategi, layaknya bermain catur, justru lebih memerlukan keahlian berpikir dan memutuskan setiap gerakan secara hati-hati dan terencana. Video *game* strategi biasanya memberikan pemain atas kendali tidak hanya satu orang tapi minimal sekelompok orang dengan berbagai jenis tipe kemampuan, sampai kendaraan, bahkan hingga pembangunan berbagai bangunan, pabrik dan pusat pelatihan tempur, tergantung dari tema ceritanya. Pemain *game* strategi melihat dari sudut pandang lebih meluas dan lebih kedepan dengan waktu permainan yang biasanya lebih lama dan santai dibandingkan *game* action. Unsur-unsur permainannya biasanya berkisar sekitar, prioritas pembangunan, peletakan pasukan, mencari dan memanfaatkan sumberdaya (uang, besi, kayu,minyak,dll), hingga ke pembelian dan peng-upgrade-an pasukan atau teknologi. *Game* jenis ini terbagi atas:

- a. *Real time Strategy, game* berjalan dalam waktu sebenarnya dan serentak antara semua pihak dan pemain harus memutuskan setiap langkah yang diambil saat itu juga berbarengan mungkin saat itu pihak lawan juga sedang mengeksekusi strateginya. Contoh: *Starcraft, Warcraft* , dan *Command and Conquer*.
- b. *Turn based Strategy* , *game* yang berjalan secara bergiliran, saat kita mengambil keputusan dan menggerakkan pasukan, saat itu pihak lawan menunggu, begitu pula sebaliknya, layaknya catur. Contoh: *Front Mission*,

Super robot wars, Final Fantasy tactics, Heroes of might and magic, Master of Orion

9. Simulasi kendaraan. Video *Game* jenis ini memberikan pengalaman atau interaktifitas sedekat mungkin dengan kendaraan yang aslinya, meskipun terkadang kendaraan tersebut masih eksperimen atau bahkan fiktif, tapi ada penekanan khusus pada detil dan pengalaman realistik menggunakan kendaraan tersebut. Terbagi atas beberapa jenis:
 - a. Perang. Video *game* simulasi kendaraan yang sempat tenar di tahun 90-an ini mengajak pemain untuk menaiki kendaraan dan berperang melawan kendaraan lainnya. Dan kebanyakan diantaranya memiliki judul sama dengan nama kendaraannya. Contoh : *Apache 64, Comanche, Abrams, YF-23, F-16 fighting eagle.*
 - b. Balapan. Dari namanya sudah jelas, siapa sampai duluan di garis finish dialah pemenangnya! Terkadang malah pemain dapat memilih kendaraan, mendandani, upgrade mesin bahkan mengecatnya. Contoh: *Top Gear, Test Drive, Sega Rally Championship, Daytona, Grand Turismo, Need For Speed, Mario Cart, ManXTT.*
 - c. Luar Angkasa. Walau masih dapat dikategorikan simulasi kendaraan perang, tetapi segala unsur fiksi ilmiah dan banyaknya judul yang beredar membuat subgenre ini pantas dikategorikan diluar simulasi kendaraan perang. Jenis ini memungkinkan pemain untuk menjelajah luar angkasa, berperang dengan makhluk alien, mendarat di planet antah

berantah atau sekedar ingin merasakan bagaimana menjadi kapten di film fiksi ilmiah kesayangan kamu. Contoh: *Wing Commander*, *Freelancer*, *Star Wars X-Wing*, *Star Wars Tie Fighter*, dll.

- d. *Mecha*. Pendapat bahwa hampir tidak ada orang yang terekspos oleh film robot jepang saat kecilnya tidak memimpikan ingin mengendalikan robot, memang sulit dibantah. Dipopulerkan oleh serial *Mechwarrior* oleh Activision, subgenre Simulasi Mecha ini memungkinkan pemainnya untuk mengendalikan robot dan menggunakannya untuk menghancurkan gedung, helikopter dan tentu saja robot lainnya. Contoh: *Mechwarrior*, *Gundam Last war Chronicles*, dan *Armored Core*

10. Aksi – *Shooting*, (tembak-tembakan, atau hajar-hajaran bisa juga tusuk-tusukan, tergantung cerita dan tokoh di dalamnya), video *game* jenis ini sangat memerlukan kecepatan refleks, koordinasi mata-tangan, juga timing, inti dari *game* jenis ini adalah tembak, tembak dan tembak. Termasuk didalamnya :

- a. *First person shooting (FPS)* seperti *Counter Strike* dan *Call of Duty*
- b. *Drive n' shoot*, menggunakan unsur simulasi kendaraan tetapi tetap dengan tujuan utama menembak dan menghancurkan lawan, contoh : *Spy Hunter*, *Rock and Roll Racing*, *Road Rash*.
- c. *Shoot em' up*, seperti *Raiden*, *1942*, dan *gradius*.

- d. *Beat 'em up* (tonjok hajar) seperti *Double Dragon* dan *Final Fight*, lalu *hack and slash* (tusuk tebas) seperti *Shinobi* dan *Legend of Kage*.
- e. *Light gun shooting*, yang menggunakan alat yang umumnya berbentuk seperti senjata, seperti *Virtua Cop* dan *Time Crisis*

Menurut Teresa Dillon elemen-elemen dasar sebuah *game* adalah :

1. *Game Rule*

Game rule merupakan aturan perintah, cara menjalankan, 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*.

2. Plot

Plot berisi informasi tentang hal-hal yang akan dilakukan oleh *player* dalam *game* atau perintah tentang hal yang harus dicapai dalam *game*

3. Thema

Dalam *game*, thema *game* lebih cenderung kepada *genre game*, yaitu berisikan informasi mengenai jenis *game*.

4. Karakter

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

5. Objek

Hal digunakan pemain untuk memecahkan masalah, adakalanya pemain harus punya keahlian dan pengetahuan untuk bisa memaninkannya.

6. Text, Grafik dan Sound

Game merupakan kombinasi dari media teks, grafik maupun suara, meskipun ada beberapa *game* tidak menggunakan ketiganya.

7. Animasi

Animasi ini selalu melekat pada dunia *game*, khususnya untuk gerakan karakter, properti, dan objek.

8. User Interface

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

2.1.2 Komponen Permainan (*Component Games*)

Dalam pembuatan *game* terdapat beberapa komponen penting yang harus diperhatikan. Berikut komponen dalam membangun *game* itu sendiri yaitu:

Opening (Splash) Screen (Pembukaan)

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

Menu Screen (Tampilan Menu)

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

Music (Musik)

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

Sound Effects (Efek Suara)

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

Time (Waktu)

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

Lives (Nyawa)

Game harus mempunyai tantangan agar menjadi menyenangkan, sehingga pemain dapat gagal dalam bermain. Membunuh *player* adalah cara yang pantas dalam

memberikan kegagalan. Beberapa *game* memberikan pemain beberapa nyawa per sesi da nada juga yang hanya memiliki nyawa.

Obstacles (Rintangan)

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

Levels (Tingkatan)

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

Adversaries (Lawan/Musuh)

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

Player (Pemain)

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

Scenes (Adegan)

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

2.1.3 Platform Android

Android adalah system operasi berbasis linux yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan computer tablet. Fungsinya sama seperti sistem operasi Symbian di Nokia, iOS di Apple dan BlackBerry OS di BlackBerry. Keunggulan utama Android adalah gratis dan *open source*, yang membuat smartphone Android dijual lebih murah dibandingkan dengan Blackberry atau iPhone meski fitur (hardware) yang ditawarkan Android lebih baik.

Sejarah singkat Android, Pada Juli 2000, Google bekerjasama dengan Android Inc., perusahaan yang berada di Palo Alto, California Amerika Serikat. Para pendiri Android Inc. bekerja pada Google, di antaranya Andy Rubi, Rich Miner, Nick Sears, dan Chris White. Saat itu banyak yang menganggap fungsi Android Inc. hanyalah sebagai perangkat lunak pada telepon seluler. Sejak saat itu muncul rumor bahwa Google hendak memasuki pasar telepon seluler. Di perusahaan Google, tim yang dipimpin Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh kernel Linux. Hal ini menunjukkan indikasi bahwa Google sedang bersiap menghadapi persaingan dalam pasar telepon seluler. Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk

mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada saat perilisan perdana Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler.

Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau Google Mail Services (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai Open Handset Distribution (OHD).

Beberapa versi android yang sudah diluncurkan antara lain:

1. Android versi 1.1 Dirilis pada : 9 Maret 2009.

Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, voice search (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

2. Android versi 1.5 (Cupcake) Dirilis pada : pertengahan Mei 2009

Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (Software Development Kit) dengan versi 1.5 (Cupcake). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan Bluetooth A2DP, kemampuan terhubung secara otomatis ke headset Bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan sistem.

3. Android versi 1.6 (Donut) Dirilis pada : September

Menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, camcorder dan galeri yang diintegrasikan; CDMA / EVDO, 802.1x, VPN, Gestures, dan Text-to-speech engine; kemampuan dial kontak; teknologi text to change speech (tidak tersedia pada semua ponsel; pengadaan resolusi VWGA.

4. Android versi 2.0/2.1 (Eclair) Dirilis pada : 3 Desember 2009

Perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan

HTML5, daftar kontak yang baru, dukungan flash untuk kamera 3,2 MP, digital Zoom, dan Bluetooth 2.1.

5. Android versi 2.2 (Froyo: Frozen Yoghurt) Dirilis pada : 20 Mei 2010

Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi V8 JavaScript engine yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD Card, kemampuan WiFi Hotspot portabel, dan kemampuan auto update dalam aplikasi Android Market.

6. Android versi 2.3 (Gingerbread) Dirilis pada : 6 Desember 2010

Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (gaming), peningkatan fungsi copy paste, layar antar muka (User Interface) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (reverb, equalization, headphone virtualization, dan bass boost), dukungan kemampuan Near Field Communication (NFC), dan dukungan jumlah kamera yang lebih dari satu.

7. Android versi 3.0/3.1 (Honeycomb)

Android Honeycomb dirancang khusus untuk tablet. Android versi ini mendukung ukuran layar yang lebih besar. User Interface pada Honeycomb juga berbeda karena sudah didesain untuk tablet. Honeycomb juga mendukung multi prosesor dan juga akselerasi perangkat keras (hardware) untuk grafis. Tablet pertama yang dibuat dengan menjalankan Honeycomb adalah Motorola Xoom. Perangkat tablet dengan platform Android 3.0 akan segera hadir di Indonesia. Perangkat tersebut bernama Eee Pad Transformer produksi dari Asus.

8. Android versi 4.0 (ICS :Ice Cream Sandwich) Diumumkan pada tanggal 19 Oktober 2011.

Membawa fitur Honeycomb untuk smartphone dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara offline, dan berbagi informasi dengan menggunakan NFC.

9. Android 4.1-4.3 “Jelly Bean” (27 Juni 2012)

Dirilis tahun 2012 ini. Android Jelly Bean ini diperuntukkan untuk komputer tablet dan memungkinkan untuk digunakan pada sistem operasi PC

atau Komputer. Sehingga rumornya kemunculan Android Jelly Bean ini untuk menyaingi rilis terbaru Windows 8 yang juga akan segera dirilis. Karena kita ketahui bersama perbincangan versi Android sebelumnya yaitu Android Ice Cream Sandwhich pun masih hangat di telinga.

10. Android versi 4.4 “KitKat” (31 Oktober 2013)

Sistem operasi Android 4.4 KitKat ini bisa dijalankan pada smartphone yang memiliki spesifikasi yang rendah, oleh sebab itu untuk kerja prosesor dari smartphone tidak terlalu berat sehingga menjadikan perangkat yang telah menggunakan sistem operasi Android 4.4 KitKat ini lebih menghemat baterai jika dibanding dengan sistem operasi Android yang menginginkan spesifikasi tinggi dan pasti kerjanya lebih berat. Untuk bisa menjalankan sistem operasi ini pengguna hanya menggunakan RAM minimum dengan kapasitas 512 MB.

2.1.4 AndEngine

Andengine merupakan suatu library kecil yang dibuat untuk memudahkan kita membuat *game* pada device Android. Pada library ini sudah terdapat kelas-kelas yang dapat mempermudah pembuatan *game*, misal kelas Animasi, Sound, Physic, dsb. Sebenarnya, ada banyak library yang dapat anda gunakan selain andengine, misal rokon, libgdx, dan masih banyak lagi. Library ini bersifat free open source, jadi tidak

perlu khawatir tentang biaya pembuatan *game*. Seperti layaknya pemrograman android, pengembangan *game* dan aplikasi menggunakan library ini menggunakan bahasa pemrograman java.

Beberapa keunggulan AndEngine adalah :

- Free.
- Relatif mudah digunakan.
- Terdapat berbagai extension yang bisa digunakan, misal Physics extension, multitouch extension, multiplayer extension, bahkan AR extension. Selain itu masih banyak extension yang lainnya.
- Komunitas yang memakai AndEngine lumayan banyak, jadi lebih mudah untuk mencari referensi di komunitas ini.

Selain berbagai kelebihan di atas, ternyata AndEngine juga memiliki berbagai kelemahan. Diantara kelemahan AndEngine adalah sbb :

- Hanya suport *game* 2D.
- Dokumentasi sangat kurang, namun hal ini dapat diatasi karena banyaknya komunitas Andengine yang siap menjawab pertanyaan anda.
- Terkadang masih ditemui bug dalam library ini, jadi jangan heran jika program anda error padahal script anda sudah benar.

2.1.5 Pathfinding

Pathfinding dalam konteks *games* adalah cara menggerakkan pemain melalui suatu jalan terpendek tanpa menabrak rintangan. Penggunaan metode pathfinding paling sering adalah pada *game* real-time (berorientasi waktu nyata) dimana seorang pemain pada titik awal menentukan titik akhir/tujuan. Pathfinding saat ini telah menjadi elemen penting di dunia *games*. Dimana semua *games* 2D/3D pasti memiliki metode pathfinding yang berbeda. Sesuai dengan tingkat kerumitan dari *games* 2D/3D yang diusung. Semakin rumit map dari suatu *game* maka semakin rumit pula metode pathfinding yang digunakan (Fitra Haryadi, 2012)

2.1.6 Artificial Intelligence (Kecerdasan Buatan)

Kecerdasan Buatan atau *Artificial Intelligence* merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan manusia. Pada awal diciptakannya, komputer hanya difungsikan sebagai alat hitung saja. Namun seiring dengan perkembangan jaman, maka peran komputer semakin mendominasi kehidupan umat manusia. Komputer tidak lagi hanya digunakan sebagai alat hitung, lebih dari itu, komputer diharapkan untuk dapat diberdayakan untuk mengerjakan sesuatu yang bisa dikerjakan oleh manusia.

Manusia bisa menjadi pandai dalam menyelesaikan segala permasalahan di dunia ini karena manusia mempunyai pengetahuan dan pengalaman. Pengetahuan diperoleh dari cara mempelajarinya. Semakin banyak bekal pengetahuan yang dimiliki oleh seseorang tentu saja diharapkan akan lebih mampu dalam menyelesaikan permasalahan. Namun bekal pengetahuan saja tidak cukup, manusia juga diberi akal untuk melakukan penalaran, mengambil kesimpulan berdasarkan pengetahuan dan pengalaman yang mereka miliki. Tanpa memiliki kemampuan untuk menalar dengan baik, manusia dengan segudang pengalaman dan pengetahuan tidak akan dapat menyelesaikan masalah dengan baik. Demikian juga dengan kemampuan menalar yang sangat baik, namun tanpa dibekali pengetahuan dan pengalaman yang memadai, manusia juga tidak akan bisa menyelesaikan masalah dengan baik (Sri Kusumadewi, 2003).

Dengan adanya kecerdasan buatan, diharapkan tidak menutup kemungkinan hanya dengan data pengetahuan yang terbatas, sebuah komputer dapat berpikir seperti manusia dalam menghadapi masalah.

Implementasi kecerdasan buatan dapat diterapkan pada bidang:

Visualisasi computer

Kecerdasan buatan pada bidang visualisasi komputer ini memungkinkan sebuah sistem komputer mengenali gambar sebagai *input*. Contohnya mengenali sebuah pola pada suatu gambar.

Pengenalan Suara

Kecerdasan buatan pada pengenalan suara ini dapat mengenali suara manusia. Cara mengenali suara ini dengan mencocokkannya pada acuan yang telah diprogramkan terlebih dahulu. Contohnya perintah komputer dengan menggunakan suara *user*.

Sistem Pakar

Kecerdasan buatan pada Sistem Pakar ini memungkinkan sebuah sistem komputer memiliki cara berpikir dan penalaran seorang ahli dalam mengambil keputusan, untuk memecahkan masalah yang ada pada saat itu. Contohnya program komputer yang dapat mendiagnosa penyakit dengan memasukan gejala-gejala yang dialami pasien.

Permainan

Kecerdasan buatan pada permainan ini memungkinkan sebuah sistem komputer untuk memiliki cara berpikir manusia dalam bermain. Contohnya permainan yang memiliki fasilitas orang melawan komputer. Komputer sudah di program sedemikian rupa agar memiliki cara bermain seperti seorang manusia bahkan bisa melebihi seorang manusia.

2.1.7 Algoritma A*(A-Star)

A* merupakan algoritma yang pertama kali dikembangkan oleh Nils Nilsson pada 1964 berdasarkan algoritma Dijkstra. Saat itu, algoritma ini dinamakan algoritma A1. Pada 1967 Bertram Raphael mengembangkan lebih jauh algoritma ini dan menyebutnya A2, namun tidak dapat membuktikan keunggulannya dibandingkan algoritma sebelumnya. Kemudian pada 1968 Peter E. menunjukkan bukti keoptimalan algoritma A2 dibandingkan dengan A1. Kemudian algoritma A2 dinyatakan sebagai algoritma paling optimal untuk kasus tersebut, dan diganti namanya menjadi A*. Berdasarkan waktu, algoritma ini lebih baik daripada algoritma Dijkstra dengan pencarian heuristik.

A*(A-star) memiliki 2 fungsi utama dalam menentukan solusi terbaik. Fungsi pertama disebut sebagai $g(x)$ merupakan fungsi yang digunakan untuk menghitung total cost yang dibutuhkan dari node awal menuju node tertentu. Fungsi kedua yang biasa disebut sebagai $h(x)$ merupakan fungsi perkiraan total cost yang diperkirakan dari suatu node ke node akhir.

$$G(n) = \sqrt{Xn^2 + Yn^2}$$

$$H(n) = |X(\text{target}) - X(n)| + |Y(\text{target}) - Y(n)|$$

$$F(n) = G(n) + H(n)$$

Prinsip algoritma ini adalah mencari jalur terpendek dari sebuah simpul awal menuju simpul tujuan dengan memperhatikan harga (F) terkecil. Diawali dengan

menenmpatkan A pada starting point, kemudian memasukkan seluruh simpul yang bertetangga dan tidak memiliki atribut rintangan dengan A ke dalam open list (Tempat menyimpan data simpul yang mungkin diakses dari starting point maupun simpul yang sedang dijalankan). Kemudian mencari nilai F terkecil dari simpul-simpul dalam open list tersebut. Kemudian memindahkan A kesimpul yang memiliki nilai F terkecil. Simpul sebelum A disimpan sebagai parent dari A dan dimasukkan kedalam closed list (tempat menyimpan data simpul sebelum A yang juga merupakan bagian dari jalur terpendek yang telah berhasil didapatkan).

Jika terdapat simpul lain yang bertetangga dengan A (yang sudah berpindah) namun belum termasuk kedalam anggota open list, maka masukkan simpul-simpul tersebut kedalam open list. Setelah itu, bandingkan nilai G yang ada dengan nilai G sebelumnya(pada langkah awal, tidak perlu dilakukan perbandingan nilai G). jika nilai G sebelumnya lebih kecil maka A kembali ke posisi awal. Simpul yang pernah dicoba dimasukkan kedalam close list. Hal tersebut dilakukan berulang-ulang hingga terdapat solusi atau tidak ada simpul lain yang berada dalam open list.

2.2 Penelitian Terkait

Ada beberapa penelitian yang terkait dengan penelitian yang dilakukan oleh penulis, beberapa diantaranya adalah :

Aghus Sofwan, ST., MT., R.Rizal Isnanto, ST., M.M, MT. dan Pasca Maulana (2008) menyatakan dalam jurnal yang berjudul "*Kecerdasan buatan dalam*

permainan snake 3D menggunakan visual basic.net dan DirectX” pencarian jalur terpendek menggunakan Algoritma A* dalam Keandalan Algoritma A Star (A*) pada permainan ini dalam hal efisiensi pencarian komputer(Non Player Character) mendapatkan makanan melalui beberapa pengujian adalah 100%. Persentase mutlak ini adalah benar selama makanan yang dicari terdapat jalan yang dapat dilalui.

Freddi Yonathan mahasiswa Institut Teknologi Bandung dalam makalahnya yang berjudul “Metode Path Finding pada *Game* 3D Menggunakan Algoritma A* dengan Navigation Mesh” Metode Path Finding merupakan metode yang sangat penting dalam *game* 3D untuk menunjang cara bergerak objek, terutama objek yang digerakkan oleh AI. Metode Path Finding perlu dilakukan sangat cepat, sebab setiap pergerakan yang ada pada *game* tidak hanya soal bergerak ke suatu lokasi tertentu, tetapi juga melakukan hal lainnya bersamaan dengan bergerak. Selain itu, sering kali perubahan tujuan gerak kurang dalam 1 detik, misalkan dalam permainan Defense of the Ancients, pemain biasanya akan terus menerus memerintahkan hero yang dimainkan untuk terus berpindah tempat secara cepat. Maka dari itu, algoritma A* dan Navigation Mesh diperlukan supaya dapat melakukan arah pergerakan ke tempat tujuan dengan sangat cepat.

Nur Fadhila, Juniardi dalam penelitian yang berjudul “Aplikasi permainan asteroid shooter menggunakan MCRNG dan A* sebagai algoritma randoming spawn dan pencarian objek berbasis mobile”. Dari penelitian tersebut penulis mengambil

refrensi bagaimana penerapan algoritma A*(A-Star) yang digunakan dalam objek asteroid dalam mengejar player.

Johannes Ridho Tumpuan Parlindungan dalam paper yang berjudul “*Penerapan Algoritma A* dalam Penentuan Lintasan Terpendek*” Untuk memecahkan problem lintasan terpendek dapat digunakan algoritma A*. Algoritma ini memperhitungkan semua kemungkinan lintasan yang ada, lalu kemudian membandingkan kemungkinan tersebut satu demi satu. Pada problem yang telah dibahas diperoleh lintasan II sebagai lintasan terpendek.

Nelly Indriani Widyastuti dan Irwan Setiawan dalam jurnal yang berjudul “*Membangun Game Edukasi Sejarah Walisongo dengan menggunakan Algoritma A*(A-Star)*”. *Game* edukasi dapat menarik minat siswa kelas V Madrasah Ibtida’iyah mempelajari sejarah Walisongo dan dapat membuat suasana pelajaran yang menyenangkan, selain itu tenaga pengajar juga dapat terbantu dalam menyampaikan materi tersebut kepada siswa yang sesuai kurikulum sekolah dasar.

2.3 Metode Penelitian

Tahap-tahap dalam penelitian ini adalah:

1. Studi Literatur

- a. Mengumpulkan informasi tentang materi kepramukaan
- b. Pengumpulan informasi tentang algoritma yang akan digunakan

2. Perancangan dan Desain Aplikasi

Perancangan aplikasi terdiri atas perancangan algoritma A-Star untuk menentukan langkah-langkah yang harus ditempuh dari keadaan awal sampai mendapatkan tujuan atau goal. Desain aplikasi adalah pembuatan design menu *game*, karakter karakter yang dibutuhkan dalam *game*, pembuatan peta dan lain lain.

3. Pembuatan Aplikasi

Pada proses pembuatan aplikasi ini, dibangunlah *game* sesuai rancangan yang kita inginkan dan ditambah dengan pengimplementasian algoritma yang akan digunakan.

4. Tahap Ujicoba dan Evaluasi

Uji coba dilakukan sampai sistem benar – benar *ready to use*, kekurangan yang terjadi diperbaiki dalam lingkup batasan masalah. Evaluasi dilakukan untuk mengetahui apakah sistem yang dibangun sudah mendekati yang diharapkan.

5. Dokumentasi

Penulisan laporan skripsi merupakan dokumentasi dari keseluruhan pelaksanaan penelitian. Diharapkan dokumentasi penelitian berguna dan bermanfaat untuk penelitian dan pengembangan lebih lanjut.

BAB III

DESAIN DAN RANCANGAN SISTEM

3.1 Deskripsi Sistem

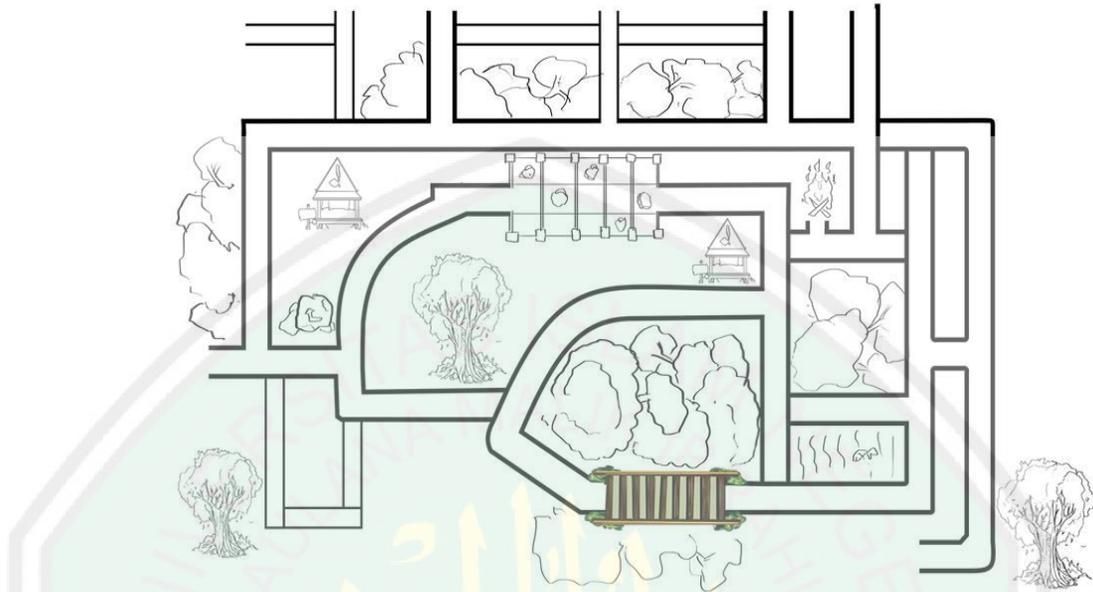
Konsep permainan game ini adalah mencari pos sebagai titik aman, berjalan melewati segala rintangan yang ada dan player harus menghindari enemy. Point akan didapatkan jika player sudah mencapai pos dan mampu menjawab pertanyaan yang diberikan. Game berakhir saat player gagal menghindari musuh.

3.2 Storyline

Game ini berlatar disebuah hutan, dalam hutan tersebut seorang anggota pramuka harus menemukan api unggun sebagai garis finish. Dalam pencarian api unggun ini akan ada banyak pos sebagai titik aman, ada banyak jalan sebelum menuju ke pos diperjalanan akan ada rintangan yang akan dilalui anggota pramuka tersebut. Rintangan tersebut adalah musuh yang digambarkan sebagai seekor monyet dan jalan buntu.

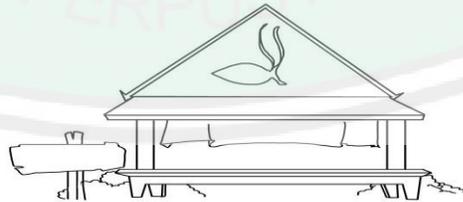
3.3 Storyboard

Storyboard adalah sketsa gambar yang disusun berurutan sesuai naskah, *storyboard* dapat menyampaikan ide cerita kepada orang lain dengan lebih mudah, karena dapat menggiring khayalan seseorang mengikuti gambar-gambar yang tersaji, sehingga menghasilkan persepsi yang sama pada ide cerita.



Gambar 3.1 : Sketsa Peta

Pada gambar 2 Peta tersebut adalah sketsa jalan didalam hutan yang akan dilalui oleh anggota pramuka tersebut, dijalan akan ada baanyak halang rintang dan petunjuk arah untuk mencapai ke pos sebagai titik aman.



Gambar 3.2: Sketsa pos sebagai titik aman

Pada gambar 3.2 adalah pos yang digunakan sebagai titik aman sebelum melanjutkan ke perjalanan menuju ke pos berikutnya, setelah sampai di pos akan ada pertanyaan pertanyaan tentang materi kepramukaan. Menurut rencana akan ada 2 pos yang akan dilalui sebelum mencapai ke garis finish.

3.4 Konten Aplikasi

Konten-konten yang terdapat pada aplikasi ini antara lain:

a) Materi pramuka dasar

Materi yang akan diajukan sebagai pertanyaan disetiap pos adalah materi kepramukaan dasar untuk tingkat siaga dan penggalang. Setiap pos ditentukan soal nya tetapi yang dikeluarkan secara acak atau random.

b) Background

Latar belakang dari game ini menggunakan setting di tengah hutan yang mempunyai banyak jalan dan rintangan.

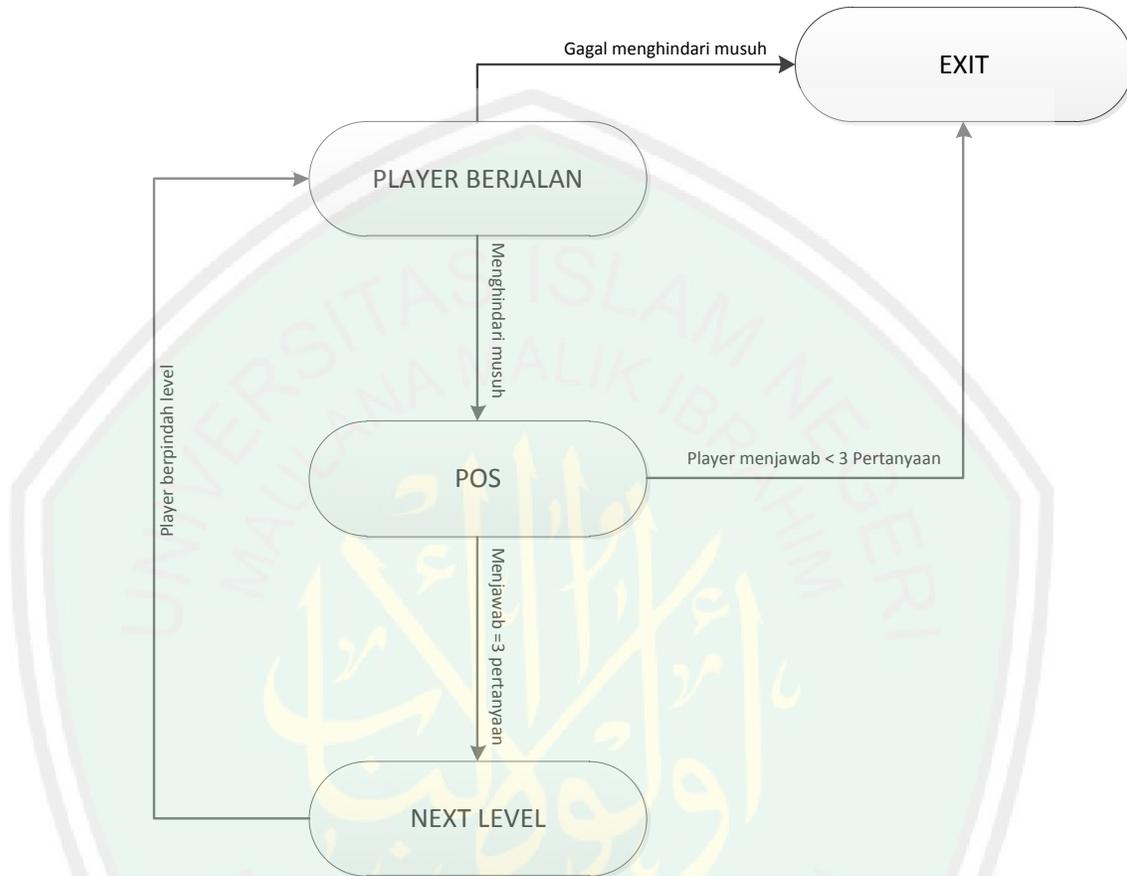
c) Tokoh Utama

Tokoh utama dalam game ini adalah seorang anggota pramuka yang mencari api unggun sebagai garis finish.

d) Tokoh Musuh

Musuh atau Enemy digambarkan sebagai seekor monyet yang akan mengganggu anggota pramuka tersebut dalam mencapai garis finish.

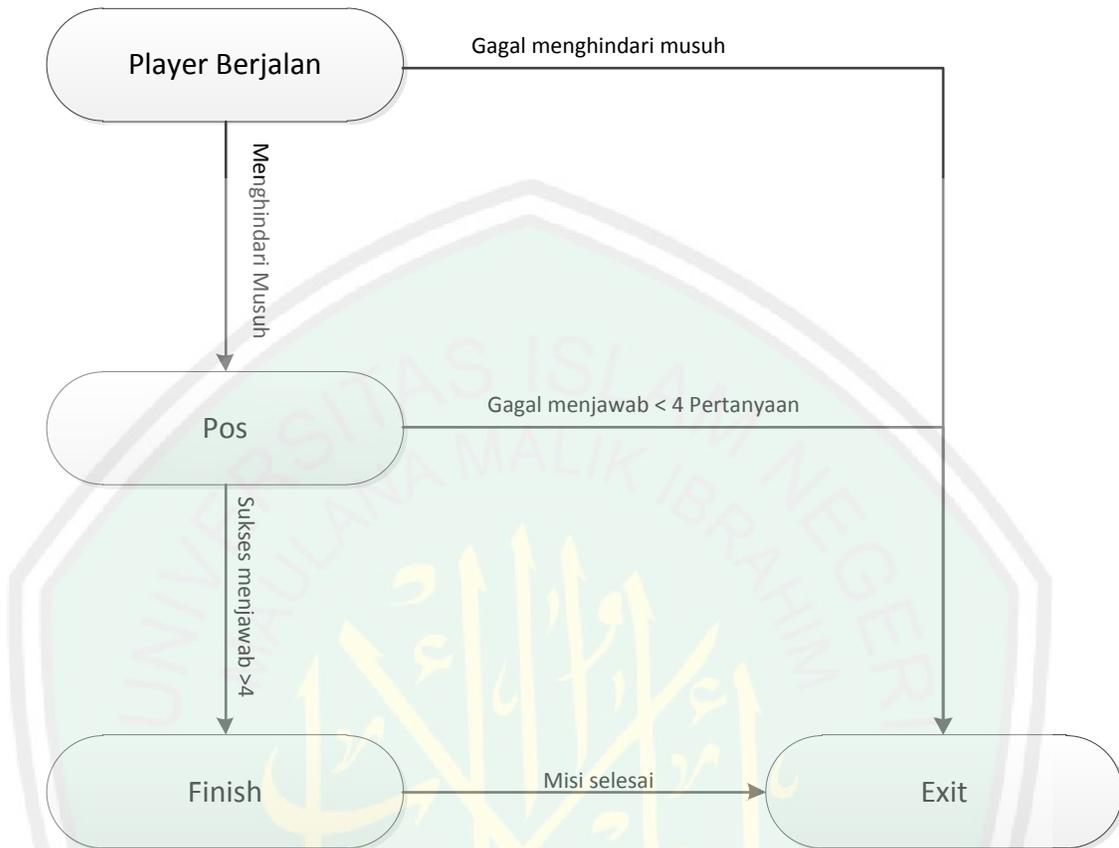
3.5 Skenario game



3.3 FSM Permainan level 1

Pada perjalanan awal, player berjalan menuju pos atau titik aman dalam perjalanan akan ada obstacle dan enemy yang akan mengganggu player dalam mencapai tujuan, jika player gagal menghindari enemy maka otomatis akan mati.

Jika telah mencapai pos atau titik aman maka akan ada pertanyaan dan jika pertanyaan mampu dijawab dengan benar sesuai syarat atau syarat untuk naik ke level selanjutnya terpenuhi.



Gambar 3.4 FSM level 2

Setelah sukses melewati level pertama, perjalanan selanjutnya adalah menuju level 2, dengan pola yang hampir sama dengan level 1, level 2 lebih menarik dikarenakan ada penambahan jumlah enemy pada level ini. Jika pada level 1 hanya menghindari 1 enemy maka level 2 akan bertambah menjadi 2 enemy dan juga kecepatan enemy nya naik menjadi kecepatan sedang.

3.5 Leveling

Dalam permainan ini tingkat kesulitan dibagi menjadi beberapa tingkatan. Tingkatan ini berdasarkan pada kesulitan pertanyaan yang akan diajukan pada setiap pos yang akan dilewati.

Pos 1

- Materi tri satya dan dasa dharma
- Menjawab minimal 2 pertanyaan dengan benar
- Kecepatan musuh lambat

Pos 2

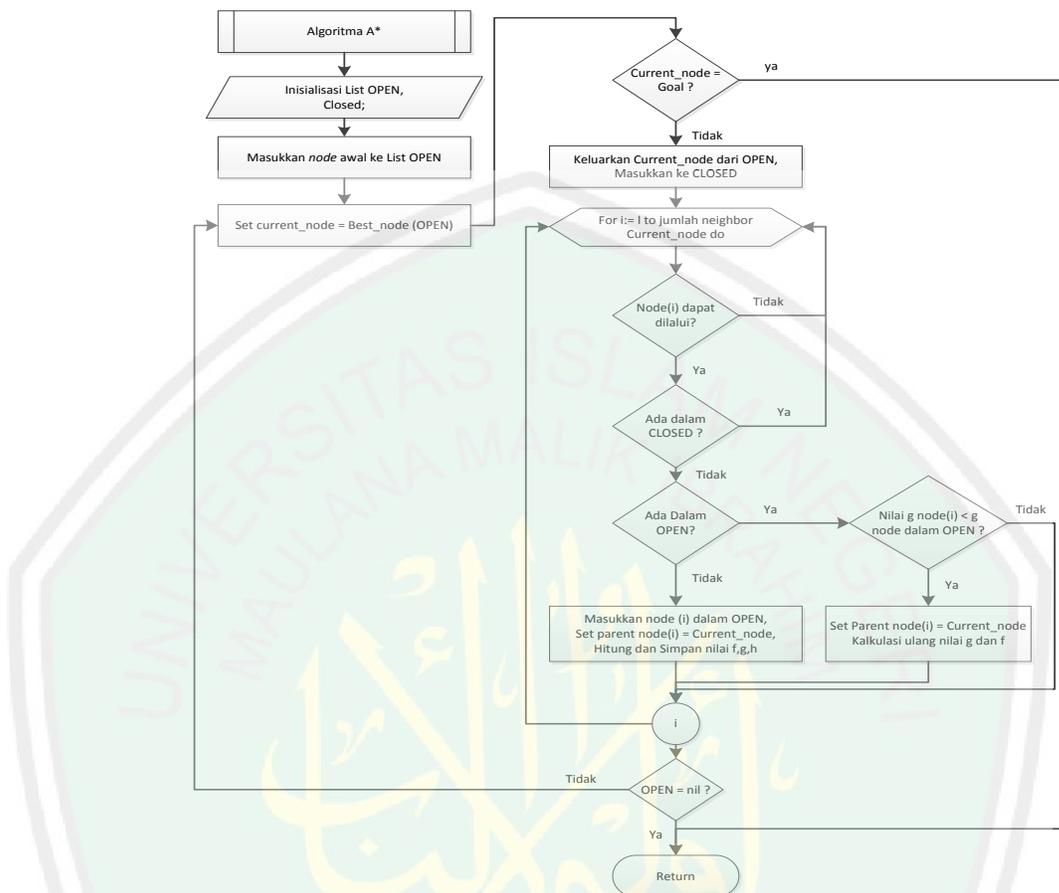
- Materi sejarah kepramukaan
- Menjawab minimal 3 pertanyaan dengan benar
- Kecepatan musuh meningkat menjadi sedang

3.6 Penerapan Metode A*(A-Star)

Algoritma A* dalam permainan ini akan diterapkan kedalam gerakan musuh untuk mengejar pemain melalui kedelapan arah yakni kiri,kanan,atas,bawah dan arah diagonal. Perancangan algoritma A* dalam aplikasi dapat di jelaskan melalui tahapan sebagai berikut. Langkah-langkah implementasi algoritma A* dalam penentuan gerak musuh dapat di jelaskan sebagai berikut berikut.

1. Masukkan node awal ke openlist
2. Loop Langkah – langkah di bawah ini :

- a. Cari node (n) dengan nilai $f(n)$ yang paling rendah dalam open list. Node ini sekarang menjadi current node.
 - b. Keluarkan current node dari openlist dan masukan ke close list
 - c. Untuk setiap tetangga dari current node lakukan berikut :
 - Jika tidak dapat dilalui atau sudah ada dalam close list, abaikan.
 - Jika belum ada di open list . Buat current node parent dari node tetangga ini. Simpan nilai f,g dan h dari node ini.
 - Jika sudah ada di open list, cek bila node tetangga ini lebih baik, menggunakan nilai g sebagai ukuran. Jika lebih baik ganti parent dari node ini di openlist menjadi current node, lalu kalkulasi ulang nilai g dan f dari node ini.
 - d. Hentikan loop jika :
 - Node tujuan telah ditambahkan ke openlist, yang berate rute telah ditemukan.
 - Belum menemukan node goal sementara open list kosong atau berarti tidak ada rute.
3. Simpan rute. Secara 'backward', urut mulai dari node goal ke parent-nya terus sampai mencapai node awal sambil menyimpan node ke dalam sebuah array.



Gambar 3.5: Flowchart Algoritma A*(A-Star)

Contoh perhitungan:

Misalkan terdapat kondisi, enemy berada pada koordinat (7,3) dan target berada pada koordinat (9,9). Maka langkah pertama yaitu, deteksi semua node disekitar enemy yang memungkinkan untuk enemy lalui. Hitung nilai G yaitu nilai yang dibutuhkan untuk berjalan satu node, H yaitu nilai heuristic atau perkiraan dari node tersebut menuju target dan F jumlah dari G dan H seperti rumus dibawah ini:

$$G(n) = \sqrt{Xn^2 + Yn^2}$$

$$H(n) = |X(\text{target}) - X(n)| + |Y(\text{target}) - Y(n)|$$

$$F(n) = G(n) + H(n)$$

9									P	
8										
7										
6										
5										
4						G=7,2 H=8 F=15,2	G=8,06 H=7 F=15,06	G=8,9 H=6 F=14,9		
3						G=6,7 H=9 F=15,7	E	G=8,5 H=7 F=15,5		
2						G=6,3 H=10 F=16,3	G=7,3 H=9 F=16,3	G=8,2 H=8 F=16,2		
1										
Y/X	1	2	3	4	5	6	7	8	9	10

Gambar 3.6 posisi awal enemy

Seperti pada gambar 3.6, kita membangkitkan semua node yang berada disekitar enemy.

- Node ke-1 pada koordinat (6,4) mempunyai nilai F sebesar 15,2

$$\begin{aligned} G_n &= \sqrt{6^2 + 4^2} \\ &= \sqrt{36 + 16} \\ &= \sqrt{52} \end{aligned}$$

$$= 7,2$$

$$H_n = |9-6| + |9-4|$$

$$= 3+5$$

$$= 8$$

$$F_n = 8+7,2$$

$$= 15,2$$

- Node ke-2 pada koordinat (6,3) mempunyai nilai F sebesar 15,7

$$G_n = \sqrt{6^2 + 3^2}$$

$$= \sqrt{36 + 9}$$

$$= \sqrt{45}$$

$$= 6,7$$

$$H_n = |9-6| + |9-3|$$

$$= 3+6$$

$$= 9$$

$$F_n = 6,7 + 9$$

$$= 15,7$$

- Node ke-3 pada koordinat (6,2) mempunyai nilai F sebesar 16,3

$$G_n = \sqrt{6^2 + 2^2}$$

$$= \sqrt{36 + 4}$$

$$= \sqrt{40}$$

$$= 6,3$$

$$\begin{aligned} H_n &= |9-6| + |9-2| \\ &= 3+7 \\ &= 10 \end{aligned}$$

$$\begin{aligned} F_n &= 6,3 + 10 \\ &= 16,3 \end{aligned}$$

- Node ke-4 pada koordinat (7,4) mempunyai nilai F sebesar 15,06

$$\begin{aligned} G_n &= \sqrt{7^2 + 4^2} \\ &= \sqrt{49 + 16} \\ &= \sqrt{65} \\ &= 8,06 \end{aligned}$$

$$\begin{aligned} H_n &= |9-7| + |9-4| \\ &= 2+5 \\ &= 7 \end{aligned}$$

$$\begin{aligned} F_n &= 8,06+7 \\ &= 15,06 \end{aligned}$$

- Node ke-5 pada koordinat (7,2) mempunyai nilai F sebesar 16,3

$$\begin{aligned} G_n &= \sqrt{7^2 + 2^2} \\ &= \sqrt{49 + 4} \\ &= \sqrt{53} \end{aligned}$$

$$= 7,3$$

$$H_n = |9-7| + |9-2|$$

$$=2+7$$

$$=9$$

$$\mathbf{F_n} =7,3+9$$

$$=16,3$$

- Node ke-6 pada koordinat (8,4) mempunyai nilai F sebesar 14,9

$$\mathbf{G_n} =\sqrt{8^2 + 4^2}$$

$$=\sqrt{64 + 16}$$

$$=\sqrt{80}$$

$$=8,9$$

$$\mathbf{H_n} =|9-8|+|9-4|$$

$$=1+5$$

$$=6$$

$$\mathbf{F_n} =8,9+6$$

$$=14,9$$

- Node ke-7 pada koordinat (8,3) mempunyai nilai F sebesar 15,5

$$\mathbf{G_n} =\sqrt{8^2 + 3^2}$$

$$=\sqrt{64 + 9}$$

$$=\sqrt{75}$$

$$=8,5$$

$$\mathbf{H_n} =|9-8|+|9-3|$$

$$=1+6$$

$$=7$$

$$F_n = 8,5+7$$

$$=15,7$$

- Node ke-8 pada koordinat (8,2) mempunyai nilai F sebesar 16,2

$$G_n = \sqrt{8^2 + 2^2}$$

$$= \sqrt{64 + 4}$$

$$= \sqrt{68}$$

$$=8,2$$

$$H_n = |9-8|+|9-2|$$

$$=1+7$$

$$=8$$

$$F_n = 8,2+8$$

$$=16,2$$

Dari hasil diatas dapat diketahui bahwa node ke 6 mempunyai nilai F yang terendah jika dibandingkan node lainnya, sehingga posisi enemy bergeser menuju node ke-6.

10										
9									P	
8										
7										
6										
5							G= 8,6 H=6 F=14,6	G= 9,4 H= 5 F= 14,4	G= 10,2 H=4 F=14,2	
4						G=7,2 H=8 F=15,2	G=8,06 H=7 F= 15,06	E Now	G= 9,8 H= 5 F= 14,8	
3						G= 6,7 H= 9 F= 15,7	E	G=8,5 H=7 F= 15,5	G= 9,4 H= 6 F= 16,4	
2						G= 6,3 H= 10 F= 16,3	G= 7,3 H= 9 F= 16,3	G=8,2 H=8 F=16,2		
1										
Y/X	1	2	3	4	5	6	7	8	9	10

Gambar 3.7 pergerakan enemy pada koordinat (8,4)

Setelah berpindah tempat maka hitung semua node baru disekitar enemy tersebut untuk mendapatkan nilai F terendah sebagai jalur yang akan dilalui enemy untuk mencapai posisi target. Node-node yang sudah pernah dihitung dimasukkan kedalam Closelist. Agar tidak dihitung kembali, seperti pada node (7,4), (7,3) dan (8,3).

Node ke-1 pada koordinat (7,5) mempunyai nilai F sebesar 14,6

$$\begin{aligned}
 G_n &= \sqrt{7^2 + 5^2} \\
 &= \sqrt{49 + 25} \\
 &= \sqrt{74}
 \end{aligned}$$

$$=8,6$$

$$H_n = |9-7|+|9-5|$$

$$=2+4$$

$$=6$$

$$F_n = 8,6+6$$

$$=14,6$$

- Node ke-2 pada koordinat (8,5) mempunyai nilai F sebesar 14,4

$$G_n = \sqrt{8^2 + 5^2}$$

$$= \sqrt{64 + 25}$$

$$= \sqrt{89}$$

$$=9,4$$

$$H_n = |9-8|+|9-5|$$

$$=1+4$$

$$=5$$

$$F_n = 9,4+5$$

$$=14,4$$

- Node ke-3 pada koordinat (9,5) mempunyai nilai F sebesar 14,2

$$G_n = \sqrt{9^2 + 5^2}$$

$$= \sqrt{81 + 25}$$

$$= \sqrt{106}$$

$$=10,2$$

$$\begin{aligned} H_n &= |9-9| + |9-5| \\ &= 0+4 \\ &= 4 \end{aligned}$$

$$\begin{aligned} F_n &= 10,2+4 \\ &= 14,2 \end{aligned}$$

- Node ke-4 pada koordinat (9,4) mempunyai nilai F sebesar 14,8

$$\begin{aligned} G_n &= \sqrt{9^2 + 4^2} \\ &= \sqrt{81 + 16} \\ &= \sqrt{97} \\ &= 9,8 \end{aligned}$$

$$\begin{aligned} H_n &= |9-9| + |9-4| \\ &= 0+5 \\ &= 5 \end{aligned}$$

$$\begin{aligned} F_n &= 9,8+5 \\ &= 14,8 \end{aligned}$$

- Node ke-5 pada koordinat (9,2) mempunyai nilai F sebesar 16,4

$$\begin{aligned} G_n &= \sqrt{9^2 + 2^2} \\ &= \sqrt{81 + 4} \\ &= \sqrt{85} \\ &= 9,4 \end{aligned}$$

$$H_n = |9-9| + |9-2|$$

$$=0+7$$

$$=7$$

$$F_n = 9,4+7$$

$$=16,4$$

hasil perhitungan pada gambar 3.7 menunjukkan bahwa nilai F terkecil terdapat pada iterasi ke-3 yaitu pada koordinat (9,5). Masukkan node (8,5), (8,4) dan (9,4) kedalam closelist.

10										
9								P		
8										
7										
6							G= 10 H= 4 F= 14	G= 10,8 H= 3 F= 13,8	G= 11,6 H= 4 F= 15,6	
5							G= 8,6 H=6 F=14,6	G= 9,4 H= 5 F= 14,4	E Now	G= 11,1 H=5 F= 16,1
4						G=7,2 H=8 F=15,2	G=8,06 H=7 F= 15,06	E 2	G= 9,8 H= 5 F= 14,8	G= 10,7 H= 6 F= 16,7
3						G= 6,7 H= 9 F= 15,7	E	G=8,5 H=7 F= 15,5	G= 9,4 H= 6 F= 15,4	
2						G= 6,3 H= 10 F= 16,3	G= 7,3 H= 9 F= 16,3	G=8,2 H=8 F=16,2		
1										
Y/X	1	2	3	4	5	6	7	8	9	10

Gambar 3.8 pergerakan enemy dari koordinat (8,4) ke (9,5)

- Node ke-1 pada koordinat (8,6) mempunyai nilai F sebesar 14

$$\begin{aligned} G_n &= \sqrt{8^2 + 6^2} \\ &= \sqrt{64 + 36} \\ &= \sqrt{100} \\ &= 10 \end{aligned}$$

$$\begin{aligned} H_n &= |9-8| + |9-6| \\ &= 1+3 \\ &= 4 \end{aligned}$$

$$\begin{aligned} F_n &= 10+4 \\ &= 14 \end{aligned}$$

- Node ke-2 pada koordinat (9,6) mempunyai nilai F sebesar 13,8

$$\begin{aligned} G_n &= \sqrt{9^2 + 6^2} \\ &= \sqrt{81 + 36} \\ &= \sqrt{117} \\ &= 10,8 \end{aligned}$$

$$\begin{aligned} H_n &= |9-9| + |9-6| \\ &= 0+3 \\ &= 3 \end{aligned}$$

$$\begin{aligned} F_n &= 10,8+3 \\ &= 13,8 \end{aligned}$$

- Node ke-3 pada koordinat (10,6) mempunyai nilai F sebesar 15,6

$$\begin{aligned} G_n &= \sqrt{10^2 + 6^2} \\ &= \sqrt{100 + 36} \\ &= \sqrt{136} \\ &= 11,6 \end{aligned}$$

$$\begin{aligned} H_n &= |9-10| + |9-6| \\ &= 1+3 \\ &= 4 \end{aligned}$$

$$\begin{aligned} F_n &= 11,6+4 \\ &= 15,6 \end{aligned}$$

- Node ke-4 pada koordinat (10,5) mempunyai nilai F sebesar 16,1

$$\begin{aligned} G_n &= \sqrt{10^2 + 5^2} \\ &= \sqrt{100 + 25} \\ &= \sqrt{125} \\ &= 11,6 \end{aligned}$$

$$\begin{aligned} H_n &= |9-10| + |9-5| \\ &= 1+4 \\ &= 5 \end{aligned}$$

$$\begin{aligned} F_n &= 11,6+5 \\ &= 16,6 \end{aligned}$$

- Node ke-5 pada koordinat (10,4) mempunyai nilai F sebesar 16,7

$$G_n = \sqrt{10^2 + 4^2}$$

$$\begin{aligned}
 &= \sqrt{100 + 16} \\
 &= \sqrt{116} \\
 &= 10,7 \\
 H_n &= |9-10| + |9-4| \\
 &= 1+5 \\
 &= 6 \\
 F_n &= 10,7+6 \\
 &= 16,7
 \end{aligned}$$

Pada gambar 3.8, posisi enemy telah berpindah dari posisi sebelumnya E2 telah berpindah posisi menuju E now pada koordinat (9,5). Dari hasil perhitungan yang dilakukan langkah selanjutnya yang harus ditempuh enemy adalah pada koordinat (9,6) dengan nilai F sebesar 13,8. Setelah dari koordinat (9,6) maka perjalanan enemy selanjutnya adalah ke koordinat (9,7) dengan nilai F sebesar 13,4, nilai tersebut kecil jika dibandingkan dengan nilai tetangga currentnode yang lainnya yaitu koordinat (8,7) dengan nilai F sebesar 13,6 dan koordinat (10,7) dengan nilai F sebesar 15,6. Jalur yang akan dilewati selanjutnya berdasarkan perhitungan yang telah dilakukan adalah koordinat (9,8) dengan nilai F sebesar 13,04. Nilai ini merupakan nilai yang terkecil diantara nilai F dari koordinat yang lainnya. Jika masih belum ketemu dengan target maka perhitungan akan terus dilakukan sampai posisi target dan posisi enemy bertemu.

10										
9								G= 12,04 H=1 F= 13,04	P	G= 13,4 H=1 F= 14,4
8								G=11,3 H= 2 F= 13,3	G= 12,04 H= 1 F= 13,04	G= 13,4 H= 1 F= 14,4
7								G= 10,6 H= 3 F=13,6	G= 11,4 H= 2 F= 13,4	G= 12,2 H= 3 F= 15,2
6								G= 10 H= 4 F= 14	G= 10,8 H= 3 F= 13,8	G= 11,6 H= 4 F= 15,6
5							G= 8,6 H=6 F=14,6	G= 9,4 H= 5 F= 14,4	E Now	G= 11,1 H=5 F= 16,1
4						G=7,2 H=8 F=15,2	G=8,06 H=7 F= 15,06	E 2	G= 9,8 H= 5 F= 14,8	G= 10,7 H= 6 F= 16,7
3						G= 6,7 H= 9 F= 15,7	E	G=8,5 H=7 F= 15,5	G= 9,4 H= 6 F= 15,4	
2						G= 6,3 H= 10 F= 16,3	G= 7,3 H= 9 F= 16,3	G=8,2 H=8 F=16,2		
1										
Y/X	1	2	3	4	5	6	7	8	9	10

Gambar 3.9 hasil akhir perhitungan enemy mencapai target

Dari gambar diatas dapat diketahui rute perjalanan yang enemy dari mulai awal hingga sampai pada target yang diinginkan yaitu dengan melalui koordinat-koordinat berikut ini, (8,4), (9,5), (9,6), (9,7), (9,8), dan koordinat (9,9) sebagai koordinat bertemunya enemy dan target.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi

Bab ini menjelaskan tentang pengujian metode yang digunakan dan analisisnya dalam membangun aplikasi yang telah dibuat. Serta implementasi metode tersebut di *game* yang akan dibuat. Tujuan dilakukan pengujian ini adalah untuk mengetahui apakah metode yang dipilih oleh penulis ini cocok dan dapat digunakan dalam aplikasi yang digunakan. Selain itu agar dapat diketahui kekurangannya agar dapat dikembangkan dan diperbaiki lebih lanjut.

4.1.1 Peralatan yang digunakan

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

Hardware:

Adapun hardware yang digunakan untuk pembuatan aplikasi dan pengujian metode tersebut adalah sebagai berikut :

- Personal Computer dengan spesifikasi : Processor AMD A10-5800K APU with Radeon(tm) HD Graphics 3,80GHz, RAM 8 GB, HD 500 GB
- Smartphone SM-T311 : Processor Dual-Core, RAM 1,5GB, Internal memory 16 GB

Software:

Sedangkan software yang digunakan dalam pembuatan aplikasi dan pengujian metode ialah:

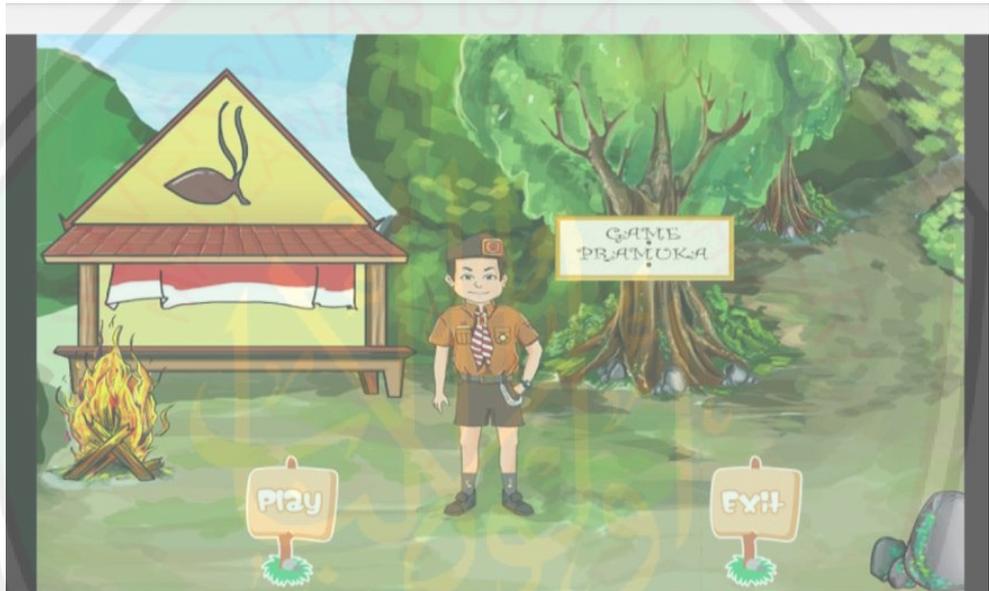
- Netbeans IDE 7.3
- Eclipse Kepler
- ADT 22.0
- Android Jelly Bean 4.2

4.2 Hasil Implementasi Aplikasi*A. Splash Screen*

Gambar 4.1 Tampilan Splash Screen

Tampilan pada gambar 4.1 merupakan tampilan splash cren dari *game* pramuka yang telah dibuat. Scene tersebut akan muncul sesaat setelah masuk kedalam aplikasi.

B. Menu Utama



Gambar 4.2 Menu Utama

Gambar 4.2 merupakan tampilan dari scene menu utama yang didalamnya terdapat dua menu utama yakni menu PLAY dan menu EXIT. Jika memilih menu PLAY maka permainan akan dimulai, dan jika memilih menu EXIT maka otomatis akan langsung keluar dari permainan

C. Permainan



Gambar 4.3 Tampilan permainan level 1



Gambar 4.4 Tampilan level 2

Tampilan pada gambar 4.3 dan gambar 4.4 merupakan tampilan dari scene permainan utama dalam permainan “*Game Pramuka*”. Dalam scene permainan ini terdapat beberapa property didalamnya diantaranya adalah karakter pemain berupa karakter player, karakter musuh dan papan tombol navigasi untuk menggerakkan player kearah atas, kanan, bawah dan kiri untuk menghindari musuh.

D. *Game Over*



Gambar 4.4 *Game Over*

Pada scene *game over* terlihat musuh dan player saling bertemu maka pada saat posisi seperti itulah *game over* terjadi. Jika player dapat menghindari musuh yang mengejar maka *game over* dapat dihindari dan melanjutkan permainan.

4.3 Pengujian algoritma A*(A-Star)

Sebelum diimplementasikan pada *game* yang akan dibuat, algoritma A-star akan diujicoba terlebih dahulu. Pada dasarnya algoritma A-Star adalah algoritma path finding dengan cara melakukan pencarian nilai heuristik terkecil dari setiap node yang ada disekitar.

Kelas *Node* tersebut digunakan untuk merepresentasikan posisi target dan posisi object yang digerakkan pada saat ini. Di dalam kelas *Node* tersebut juga memiliki object bernama *parent* yang menunjukkan bahwa node tersebut datang dari node mana.

Langkah pertama yaitu mencari node-node di sekitar *current node* kemudian kita tampung semua tersebut kedalam suatu *array*. Setelah ditampung maka kita hitung nilai F untuk setiap node yang ada dalam *array* sekitar tersebut dan kita pilih node yang memiliki nilai F terkecil sebagai *best node*. Kemudian kita pindahkan *current node* menuju ke *best node*. Lakukan hal tersebut berulang-ulang hingga nilai *current node* sama dengan nilai *target node*.

Untuk percobaannya terlebih dahulu kita tentukan *node* awal dan *node* tujuan yang akan digunakan. Misalkan *node* awal pada koordinat (3,7) dan koordinat tujuan adalah (9,9). Seperti pada source code dibawah ini.

```

public static void main(String[] args) {
    AStarTest astar = new AStarTest();
    Node a = new Node(3, 7);
    Node target = new Node(9, 9);
    Node temp;
    while (!a.equals(target)) {
        temp = astar.nextNode(a, target);
        temp.SetParent(a);
        a = temp;
        System.out.print("current X: " + a.getX() + " , current Y: " + a.getY());
        System.out.println();
        System.out.println();
    }
}

```

Gambar 4.5 Void main ujicoba pada Netbeans

Kemudian kita looping hingga nilai *current node* sama dengan *target node*. Saat memindahkan *current node* menuju *next node*, kita set parent dari next node tersebut adalah current node agar pada saat penghitungan nilai F, nilai parent tidak usah dihitung lagi agar dapat lebih optimal. Maka output yang kita dapatkan adalah seperti yang ada di bawah ini.

Langkah	Nilai G(min)	Nilai H(min)	Nilai F(min)	X,Y
1	8.94427190999916	6.0	14.94427190999916	X=4 Y=8
2	10.295630140987	4.0	14.295630140987	X=5 Y=9
3	10.816653826391969	3.0	13.816653826391969	X=6 Y=9
4	11.40175425099138	2.0	13.40175425099138	X=7 Y=9
5	12.041594578792296	1.0	13.041594578792296	X=8 Y=9

Langkah	Nilai G(min)	Nilai H(min)	Nilai F(min)	X,Y
6	12.727922061357855	0.0	12.727922061357855	X=9 Y=9

Tabel 4.1 Hasil ujicoba A-Star pada Netbeans

Dari percobaan diatas dapat diketahui bahwa current node dapat menuju target node dengan 6 iterasi menggunakan nilai F terkecil. Hasilnya, dipilih sebuah node yang berada pada koordinat (4,8) dengan nilai F paling minimum bernilai 14,9. Kemudian pindahkan current node menuju node tersebut, lalu hitung kembali nilai G, H, dan F pada lokasi *current node* yang baru. Didapatlah koordinat (5,9) dengan nilai F paling minimum yaitu 14,2. Pindahkan current node pada koordinat terbaru tersebut. Ulangi terus prosesnya hingga koordinat current node sama dengan koordinat target node.

Setelah mengetahui pola jalan algoritma A-Star, kemudian sekarang saatnya pengimplementasian pada permainan.

4.4 Implementasi algoritma A* (A-Star)

Setelah melalui ujicoba seperti diatas algoritma A-Star akan diimplementasikan pada NPC yang digunakan untuk pencarian jalur terpendek musuh menuju player. Algoritma Astar akan diterapkan pada object musuh

dimana musuh akan mencari player dan jika sampai player ditemukan (bertabrakan) maka akan *game over*.

Pertama kita mendefinisikan objek node terlebih dahulu setelah itu membuat kelas A-Star. Menginisialisasikan parameter yang menjadi inputan diantaranya Open node, Visited node, From node, to Node dan lain-lainnya. Kemudian deklarasikan nilai awalnya dari tiap-tiap parameter.

Pertama buat kondisi awal jika path yang ditemukan melebihi batas maksimum yang ditentukan maka Algoritma A-star tidak dijalankan, jika tidak melebihi A-star akan melakukan perhitungan. Node pertama pada openlist itu adalah node dengan nilai terendah. Kemudian kita cek apakah node yang dipijak sekarang merupakan tujuan akhir jika iya kita hentikan iterasi perhitungan, jika tidak perhitungan dilanjutkan. masukkan node yang kita pijak sekarang ke visited node agar tidak ikut perhitungan, kemudian kita cek data node tetangga sekitar.

Jika node tetangga diblok maka kita masukkan ke visited node supaya tidak masuk kedalam perhitungan. Kita hitung biaya yang digunakan untuk menuju tiap node tetangga tersebut yang tidak diblok. Node tetangga dengan biaya terkecil akan menjadi waypoint selanjutnya. Perhitungan tersebut akan diulang terus sampai waypoint terakhir adalah tujuannya.

Jika iterasi sudah selesai maka kita cek apakah kita menemukan path tujuan, jika path ditemukan maka kita traceback path tersebut.

4.5 Ujicoba aplikasi pada berbagai device

-Tabel 4.2 tabel hasil ujicoba pada device

No	Layar	Versi OS	CPU	RAM	Keterangan
1	5,3 inchi	4.2.2 (Jelly Bean)	Dual core 1,3 Ghz Chipset Mediatek MT-6572 GPU Mali 400 MP	512Mb	Tampilan menu berjalan sempurna, tombol berjalan sempurna. Tampilan <i>game</i> berjalan sempurna
2	8,0 inchi	4.4.2 (Kitkat)	Dual-core 1,5 GHz GPU Mali 400	1,5Gb	Tampilan menu berjalan sempurna, tombol berjalan sempurna. Tampilan <i>game</i> berjalan sempurna
4	3,5 inchi	4.0.3 (Ice Cream Sandwich)	800 MHz Cortex-A5	512Mb	Tampilan menu berjalan sempurna,tombol berjalan sempurna. Tampilan <i>game</i> berjalan sempurna
5	4,5 inchi	4.2.2 (Jelly Bean)	Dual-core 1.3 GHz	512Mb	Tampilan menu berjalan sempurna,tombol berjalan sempurna. Tampilan <i>game</i> berjalan sempurna
6	3,5 inchi	4.1.2 (Jelly Bean)	1GHz Cortex-A9	512Mb	Tampilan menu berjalan sempurna,tombol berjalan sempurna. Tampilan <i>game</i> berjalan sempurna

Dari pengujian yang dilakukan sebanyak 6 kali pengujian dapat diketahui bahwa *game* pramuka telah berhasil diujicoba pada berbagai device mulai ukuran layar 3,5 inchi sampai dengan ukuran layar 8.0 inchi. Dan dapat berjalan mulai dari Android Ice Cream Sandwich sampai dengan Android KitKat.

-Tabel 4.3 presentasi hasil ujicoba

No	Jenis Pengujian	Baik		Kurang Baik	
		Jumlah	%	Jumlah	%
1	Sistem	6	$(6/6) \times 100 = 100\%$	0	$(0/6) \times 100 = 0\%$
2	Tombol	4	$(4/4) \times 100 = 100\%$	0	$(0/4) \times 100 = 0\%$
3	Tampilan	5	$(5/5) \times 100 = 100\%$	0	$(0/5) \times 100 = 0\%$

Dari tabel 4.3 dapat diketahui bahwa semua sistem, tombol dan tampilan, semua dapat berjalan dengan baik pada device-device yang diujicobakan.

4.6 Integrasi Aplikasi dengan islam

Salah satu butir isi dalam dasa darma pramuka adalah mencintai alam dan kasih sayang sesama manusia. Implementasi dari mencintai alam adalah dengan merawatnya bukan malah merusak alam, seperti yang terjadi akhir-akhir ini.

Dalam hal ini dalam islam terdapat penjelasan pada Surat Ar-Rum ayat 41.

ظَهَرَ الْفَسَادُ فِي الْبَرِّ وَالْبَحْرِ بِمَا كَسَبَتْ أَيْدِي النَّاسِ لِيُذِيقَهُمْ
بَعْضَ الَّذِي عَمِلُوا لَعَلَّهُمْ يَرْجِعُونَ ﴿٤١﴾

“Telah nampak kerusakan di darat dan di laut disebabkan karena perbuatan tangan manusia, supaya Allah merasakan kepada mereka sebagian dari

(akibat) perbuatan mereka, agar mereka kembali (ke jalan yang benar)”.(QS: Ar-Rum, 41).

Menurut tafsir Ibnu Katsir adalah telah tampak kerusakan didarat dan dilaut disebabkan oleh tangan manusia. Sesungguhnya kekurangan tanaman pangan dan buah-buahan itu disebabkan oleh aneka kemaksiatan. Abu Aliyah berkata, “barang siapa yang durhaka pada Allah dimuka bumi ini, berarti dia berbuat kerusakan dibumi, hal itu karena kedamaian bumi dan langit adalah dengan ketaatan”. Dan Allah menguji manusia dengan kekurangan kekayaan diri, dan buah-buahan. Ujian ini merupakan cobaan dan balasan atas perbuatan mereka. Agar mereka kembali kejalan yang lurus.

Dari ayat dan tafsir diatas kerusakan didarat dan laut terjadi karena ulah tangan-tangan manusia yang tidak bertanggung jawab, eksploitasi secara besar-besaran tanpa memperdulikan efek lingkungan yang terjadi, dilaut misalkan, efek eksploitasi pengkapan ikan besar-besaran menggunakan bom atau obat dapat merusak ekosistem alam bawah laut, didarat, eksploitasi hutan secara besar-besaran. Hutan digunduli secara ilegal untuk dijadikan lahan-lahan industri yang tidak mempunyai resapan air yang mumpuni, maka terjadilah banjir dan longsor.

Allah telah menciptakan seluruh alam semesta yang terdiri dari manusia, binatang, tumbuhan-tumbuhan, dan benda-benda alam. Bumi, alam, hewan, dan tumbuh-tumbuhan tersebut diciptakan Allah bagi kesejahteraan manusia. Karena itu, sudah selayaknya pemberian Allah ini dikelola, dimanfaatkan, dan dibangun. Wajar dan pantaslah Pramuka, secara alamiah, melimpahkan cinta kepada alam sekitarnya (benda alam, satwa, dan tumbuh-tumbuhan), kasih sayang kepada sesama manusia dan sesama hidup serta menjaga kelestariannya.

Kelestarian benda alam, satwa, dan tumbuh-tumbuhan perlu dijaga dan dipelihara karena hutan tanah, pantai, fauna, dan flora serta laut merupakan sumber alam yang perlu dikembangkan untuk menunjang kehidupan generasi kini dan dipelihara kelestariannya untuk kehidupan generasi mendatang.

Rasulullah saw, bersabda, “Seorang muslim tidak menanam tanaman kecuali apa yang dimakan dari tanaman itu menjadi sedekah baginya. Apa yang dicuri dari tanaman itu menjadi sedekah baginya. Apa yang dimakan binatang buas menjadi sedekah baginya. Apa yang dimakan burung menjadi sedekah baginya. Dan tidaklah orang lain mengambil manfaat (dari pohon itu) kecuali menjadi sedekah bagi (penanam)nya....”(HR: Muslim dari Ibnu Numair). Hadis ini menekankan pentingnya menanam, bukan semata menikmati hasilnya, ada banyak manfaat lain dari menanam pohon antara lain:

1. Manfaat estetis (keindahan), pohon memiliki beberapa bentuk tajuk yang khas, sehingga menciptakan keindahan tersendiri. Oleh karena itu, apabila di susun secara berkelompok dengan jenis yang sama pada masing-masing kelompok dapat menciptakan keindahan atau suasana yang nyaman. Struktur bangunan tanpa di imbangi dengan pohon, akan terasa gersang, sebaliknya apa bila di sekitarnya di tanami pohon serta di tata dengan baik akan nampak hijau dan asri.

2. Manfaat Orologis, akar pohon dan tanah merupakan satu kesatuan yang kuat, sehingga mampu mencegah erosi atau pengikisan tanah

3. Manfaat Hidrologis, dalam hal ini di maksudkan bahwa tanaman pada dasarnya akan menyerap air hujan. Dengan demikian, banyaknya kelompok pohon-pohon akan menjadikan daerah sebagai daerah resapan/persediaan air tanah yang dapat memenuhi kehidupan bagi manusia dan makhluk lainnya

4. Manfaat Klimatologis, dengan banyaknya pohon, akan menurunkan suhu setempat, sehingga udara sekitarnya akan menjadi sejuk dan nyaman. Maka, kehadiran kelompok pohon-pohon pelindung sangat besar artinya.

5. Manfaat Edaphis, ini manfaat dalam kaitan tempat hidup binatang. Dilingkungan yang penuh dengan pohon, satwa akan hidup dengan tenang karena lingkungan demikian memang sangat mendukung.

6. Manfaat Ekologis, lingkungan yang baik adalah seimbang antara struktur buatan manusia dan struktur alam. Kelompok pohon atau tanaman, air, dan binatang adalah bagian dari alam yang dapat memberikan keseimbangan lingkungan.

8. Manfaat Hygienis, adalah sudah menjadi sifat pohon pada siang hari menghasilkan O₂ (oksigen) yang sangat di perlukan oleh manusia, dan sebaliknya dapat menyerap CO₂ (karbondioksida) yaitu udara kotor hasil gas buangan sisa pembakaran. Jadi secara hygienis, pohon sangat berguna bagi kehidupan manusia.

Di samping itu, sebagai Negara kepulauan pemanfaatan wilayah pesisir dan lautan yang sekaligus memelihara kelestarian sumber alam ini dengan

menanggulangi pencemaran laut, perawatan hutan, hutan bakau dan hutan payau, serta pengembangan budi daya laut menduduki tempat yang penting pula.



BAB V

PENUTUP

5.1 Kesimpulan

1. Konten edukasi pada game pramuka ini digunakan sebagai pembelajaran mengenai materi-materi kepramukaan tingkat siaga dan penggalang.
2. Algoritma A*(A-Star) berhasil diimplementasikan pada game pramuka untuk mencari jalur terpendek dari sebuah koordinat awal menuju koordinasi game pada deat tujuan dengan menggunakan nilai F terkecil.
3. Ditinjau dari implementasi A-Star pada game tersebut telah didapatkan tingkat keberhasilan sebesar 100%.
4. Ditinjau dari implementasi pada device berukuran mulai dari 3,5 inci sampai ukuran 8 inci dan berOS Android versi Ice Cream Sandwich sampai KitKat diperoleh tingkat keberhasilan sebesar 100%.

5.2 Saran

Dalam game ini masih terdapat beberapa kekurangan yang dapat disempurnakan oleh penelitian selanjutnya dan dapat dikembangkan agar semakin diminati. Penulis dapat memberikan saran antara lain:

1. Baksound kedalam permainan
2. Penambahan level, pada game pramuka ini masih ada hanya 2 level.
3. Penambahan bank soal tentang kepramukaan

4.Perbaikan tampilan sangat diperlukan

5.Game ini hanya dapat dioperasikan pada OS android, alangkah lebih baiknya jika dikembangkan ke berbagai platform, misal: iOS, Windows Phone dan lain sebagainya.



DAFTAR PUSTAKA

- Ar-Rifa'I, Muhammad Nasib. 2000. *Ibnu Katsir jilid 3*. Gema insani, Jakarta
- Abt, Clark C. 1970. *Serious Games*. New York: The Viking Press
- Haryadi, Fitra. 2012. *Metode Pathfinding pada map 2D Menggunakan Algoritma Diagonal & Bidirectional BFS IF3051 Strategi Algoritma*. Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung. Bandung
- Hapsari Tilawah, 2011, *Penerapan Algoritma A-Star(A*) Untuk Menyelesaikan Masalah Maze*, Jurnal, Teknik Informatika, Institut Teknologi Bandung, Bandung.
- Harianja, Firman. 2013, *PENERAPAN ALGORITMA A* PADA PERMASALAHAN OPTIMALISASI PENCARIAN SOLUSI DYNAMIC WATER JUG*, Pelita Informatika Budi Dharma. Medan
- Hertanto, Aditya. 2012. *Perancangan Program Aplikasi Mobile Pencarian Lokasi Dalam Gedung Menggunakan Algoritma A-Star*. Binus University. Jakarta
- Higgins, H., Ritzhaupt, A. & Allred, B. (2010). *Teacher Experiences on the Integration of Modern Educational Games in the Middle School Mathematics Classroom*. *Journal of Computers in Mathematics and Science Teaching*, 29(2), 189-216. Chesapeake, VA: AACE.

Ian, 2011. Minat Siswa ke pramuka masih rendah, <http://www.radarbangka.co.id/rubrik/detail/edukasi/2193/minat-siswa-ke-pramuka-masih-rendah.html> , Diakses pada 16 September 2014, 21.30 WIB

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

Kusumadewi, Sri.2003.*Artificial Intelligence (Teknik dan Aplikasinya)*, Yogyakarta: Graha Ilmu

Missyamel, 2010. Pengertian Teknologi Game, <http://www.joey-amel.blogspot.com/2010/04/pengertian-teknologi-game.html>, diakses pada 16 September 2014, 22.00 WIB

Munir, Rinaldi.2009.*Diktat Kuliah IF3051 Strategi Algoritma*.Bandung:Program Studi Teknik Informatika ITB

Nurfadila, juniardi. 2014. *Aplikasi permainan Asteroid Shooter Menggunakan MCRNG dan A* Sebagai Algoritma Randoming Spawn dan Pencarian Objek Berbasis Mobile*. Jurusan Teknik Informatika Universitas Islam Maulana Malik Ibrahim Malang, Malang

Parlindungan, Johannes Ridho Tumpuan.2011. *Penerapan Algoritma A* Dalam Penentuan Lintasan Terpendek*. Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung. Bandung

Patrick Lester, A* pathfinding for Beginner.

http://www.gamedev.net/page/resources/_/technical/artificial-intelligence/a-pathfinding-for-beginners-r2003 diakses pada 23 Desember 2014

Ramadhani, Aristam. 2008, *Menggerakkan Karakter Game Menggunakan Algoritma Breadth-First Search (BFS) dan Algoritma A* (A Star)*. Makalah IF2251Strategi Algoritmik Tahun 2008. Program Studi Teknik Informatika Sekolah Tinggi Elektro dan Informatika, Institut Teknologi Bandung, Jalan Ganesha 10, Bandung , Indonesia.

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

Setiawan, Irwan dan Nelly Indriani Widyastuti.2012. *Membangun Game Edukasi Sejarah Wali Songo dengan Menggunakan Algoritma A**. Jurusan Teknik Informatika FTIK UNIKOM. Bandung

Setyawan M, Bayu. *Optimasi Rute Perjalanan Ambulance Menggunakan Algoritma A**. Jurusan Teknik Elektro Institute Sepuluh November. Surabaya

Sofwan, Agus dkk. 2008. *Kecerdasan Buatan Dalam Permainan Snake 3D menggunakan Visual Basic.net dan DirectX*. Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro (UNDIP) Semarang. Semarang

Widyastuti, Nelly Indriani dan Irwan Setyawan. 2012. *Membangun Game Edukasi Sejarah Walisongo*. Komputa. Bandung

Yonathan, Freddi.2011. *Metode Path Finding pada game 3D Menggunakan Algoritma A* Dengan Navigation Mesh*. Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung. Bandung

