

IMPLEMENTASI DAN ANALISIS ALGORITMA A* DAN *JUMP POINT SEARCH* UNTUK MENENTUKAN JALUR TERPENDEK PADA *NON PLAYER CHARACTER*

Diajukan Sebagai Syarat Untuk Menyelesaikan
Pendidikan Program Strata-1 Pada
Jurusan Teknik Informatika



Oleh :

DENI APRILYADI
09021381320034

**Jurusan Teknik Informatika
FAKULTAS ILMU KOMPUTER UNIVERSITAS SRIWIJAYA
2020**

LEMBAR PENGESAHAN TUGAS AKHIR

**IMPLEMENTASI DAN ANALISIS ALGORITMA A*
DAN *JUMP POINT SEARCH* UNTUK MENENTUKAN
JALUR TERPENDEK PADA *NON PLAYER CHARACTER***

Oleh :

Deni Aprilyadi
09021381320034

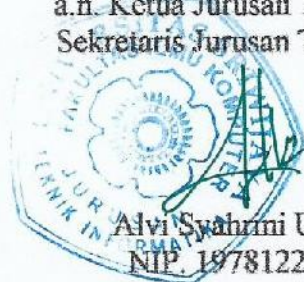
Palembang, Februari 2020

Mengetahui
Pembimbing I,



Yoppy Bazaki, M.T
NIPUS. 197406062012101201

Menyetujui,
a.n. Ketua Jurusan Teknik Informatika,
Sekretaris Jurusan Teknik Informatika,



Alvi Syahrini Utami, M.Kom.
NIP. 197812222006042003

TANDA LULUS UJIAN SIDANG TUGAS AKHIR

Pada hari Jum'at tanggal 31 Januari 2020 telah dilaksanakan Ujian Sidang Tugas Akhir oleh Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Sriwijaya.

Nama : Deni Aprilyadi

NIM : 09021381320034

Judul : Implementasi Dan Analisis Algoritma A* Dan Jump Point Search Untuk Menentukan Jalur Terpendek Pada Non Player Character

1. Pembimbing I

Yoppy Sazaki, M.T.

NIPUS. 197406062012101201



2. Penguji I

M. Fachrurrozi, M.T

NIP. 198005222008121002



3. Penguji II

Yunita, M.Cs.

NIP. 198306062015042002



Mengetahui,
a.n. Ketua Jurusan Teknik
Informatika,
Sekretaris Jurusan Teknik
Informatika,



Alvi Syahrini Utami, M.Kom.
NIP. 197812222006042003

HALAMAN PERNYATAAN BEBAS PLAGIAT

Yang bertanda tangan di bawah ini :

Nama : Deni Aprilyadi
NIM : 09021381320034
Program Studi : Teknik Informatika
Judul Skripsi : Implementasi Dan Analisis Algoritma A* Dan *Jump Point Search* Untuk Menentukan Jalur Terpendek Pada *Non Player Character*
Hasil Pengecekan Software *iThenticate/Turnitin* : 15 %

Menyatakan bahwa Laporan Proyek saya merupakan hasil karya sendiri dan bukan hasil penjiplakan/plagiat. Apabila ditemukan unsur penjiplakan/plagiat dalam laporan proyek ini, maka saya bersedia menerima sanksi akademik dari Universitas Sriwijaya sesuai dengan ketentuan yang berlaku.

Demikian, pernyataan ini saya buat dengan sebenarnya dan tidak ada paksaan oleh siapapun.

Palembang, Januari 2020



Deni Aprilyadi
NIM. 09021381320034

MOTTO DAN PERSEMBAHAN

- Allah tidak memikulkan beban kepada seseorang melainkan sekedar apa yang Allah berikan kepadanya. Allah kelak akan memberikan kelapangan sesudah kesempitan. – *Q.S Ath-Thalaq:7*
- Tidak ada balasan kebaikan kecuali kebaikan (pula). – *Q.S Ar-Rahman:60*
- Never forget what you are. The rest of the world will not. Wear it like armor, and it can never be used to hurt you – *Tyrion Lannister*
- *Valar Morghulis*
- Be kind, be honest

Kupersembahkan karya tulisi ini kepada:

- Kedua Orang Tuaku
- Keluarga Besarku
- Sahabat-sahabatku
- Dosen Pembimbingku
- Almamaterku

KATA PENGANTAR



Alhamdulillah Robbil'Alamin, segala puji bagi dan syukur bagi Allah SWT karena berkat rahmat dan karunia-Nya. Alhamdulillah Djazakumullahu Khaira, segala syukur, puji dan sholawat teruntuk Nabi besar, Nabi Muhammad SAW karena berkat perjuangan dan tuntunan belliau penulis dapat menyelesaikan tugas akhir ini dengan manisnya keimanan. Pada kesempatan ini penulis ingin mengucapkan terima kasih yang setulusnya kepada:

1. Kedua Orang Tua terkasih, Nawawi dan Suryani yang telah senantiasa mendoakan, menasehati, memberikan dukungan yang luar biasa dan rela bersabar kepada penulis;
2. Adik-adik ku tersayang, Ayu Febriyanti, Septina Khoirunnisah, dan Rahmat Rizki Ramadan yang selalu menjadi alasan dan motivasi untuk menyelesaikan tugas akhir ini serta kepada seluruh keluarga besar yang juga memberikan dukungan serta doa;
3. Bapak Jaidan Jauharim, S.Pd., M.T. selaku Dekan Fakultas Ilmu Komputer;
4. Bapak Rifkie Primartha, M.T. selaku Ketua Jurusan Teknik Informatika
5. Ibu Novi Yusliani, M.T. selaku dosen pembimbing akademik yang telah memberikan bimbingan secara akademik dan non akademik;
6. Bapak Yoppy Sazaki, M.T. dan Ibu Anggina Primanita, M.IT selaku dosen pembimbing tugas akhir yang telah memberikan bimbingan, ilmu pengetahuan, nasihat serta mempermudah penulis untuk menyelesaikan tugas akhir ini. Terima kasih banyak Bapak dan Ibu;
7. Bapak M. Fachrurrozi, M.T. dan Ibu Yunita, M.Cs selaku dosen penguji yang telah menguji serta memberikan banyak saran dan ilmu pengetahuan kepada penulis;
8. Seluruh Bapak dan Ibu Dosen Jurusan Teknik Informatika, Serta Staff Tata usaha Fakultas Ilmu Komputer Universitas Sriwijaya yang telah memberikan ilmu pengetahuan kepada penulis selama kegiatan akademik berlangsung;
9. Teman-teman seperjuangan Teknik Informatika Angkatan 2013;

10. Teman-teman organisasi BEM KM Unsri;
11. Seluruh rekan-rekan tim project pak yoppy yang telah memberikan semangat dan motivasi bagi penulis;
12. Seluruh rekan-rekan di PudingLab yang telah membantu penulis dan memberikan semangat, pengalaman, dan cuan;
13. Seluruh anggota SETARBAK ILKOM yang sudah berbagi canda tawa selama masa perkuliahan;
14. Sahabat seperjuangan ku Andi, Ipin, dan Rezi yang telah banyak membantu penulis dalam melewati masa perkuliahan dan kesediannya untuk menjadi tempat menumpang penulis;
15. Sahabat berceritaku Fiqih dan Faris yang telah banyak memberikan nasihat dan ruang bercerita kepada penulis;
16. Sahabat SUPI (Arif, Dita, Opi, Lisa, Angga, Meti, dan Mana) yang telah memberikan canda tawa dan kebersamaan kepada penulis;
17. Adik-adik perkuliahanku Andini dan Mutia yang telah membuat hari-hari perkuliahan penulis menjadi ceria dan berkesan;

Penulis menyadari dalam penyusunan Tugas Akhir ini masih terdapat banyak kekurangan disebabkan keterbatasan pengetahuan dan pengalaman, oleh karena itu kritik dan saran yang membangun sangat diharapkan untuk kemajuan penelitian selanjutnya.

Akhir kata semoga Tugas Akhir ini dapat berguna dan bermanfaat bagi kita semua.

Indralaya, Februari 2020

Penulis

IMPLEMENTASI DAN ANALISIS ALGORITMA A* DAN *JUMP POINT SEARCH* UNTUK MENENTUKAN JALUR TERPENDEK PADA *NON PLAYER CHARACTER*

Deni Aprilyadi (09021381320034)

Jurusan Teknik Informatika Fakultas Ilmu Komputer

Universitas Sriwijaya

ABSTRAK

Pathfinding adalah cara untuk menemukan rute terpendek antara dua titik. Algoritma A* dan *Jump Point Search* menggunakan jumlah *node* yang harus dilalui untuk menentukan jalur terpendek yang dapat dilalui oleh NPC. Untuk mencapai tujuan, NPC harus melalui *node* yang bukan merupakan hambatan sehingga dapat membentuk suatu jalur yang akan diambil sebagai jalur terpendek. Algoritma A* menggabungkan *Uniform Cost Search* dan *Greedy-Best First Search*, dengan biaya yang diperhitungkan didapat dari biaya sebenarnya ditambah dengan biaya perkiraan, sedangkan algoritma JPS menggunakan *pruning rules* dan *jumping rules* untuk melakukan pemangkasan sederhana, dengan tujuan untuk melompati banyak tempat yang hanya dianggap perlu untuk dilewati. Dari 5 level pengujian yang dilakukan, didapatkan hasil bahwa algoritma JPS mampu mereduksi jumlah *node* yang digunakan untuk melakukan pencarian jalur dengan selisih jarak 533% sampai 1700% lebih pendek dibandingkan dengan algoritma A*.

Kata kunci: *Pathfinding*, *Non Player Character*, Jalur Terpendek, Algoritma A*, Algoritma *Jump Point Search*

IMPLEMENTATION AND ANALYSIS OF A* AND JUMP POINT SEARCH ALGORITHM TO DETERMINE THE SHORTEST PATH OF A NON PLAYER CHARACTER

Deni Aprilyadi (09021381320034)

Jurusan Teknik Informatika Fakultas Ilmu Komputer

Sriwijaya University

ABSTRACT

Pathfinding is the way to find the shortest route between two points. The A* and Jump Point Search algorithms use the number of nodes that must be taken to determine the shortest path that can be traversed by the NPC. To reach the goal, the NPC must go through a node that is not an obstacle so it can form a path that will be taken as the shortest path. A * algorithm combines Uniform Cost Search and Greedy-Best First Search, with the calculated costs derived from actual costs plus estimated costs, whereas the JPS algorithm uses pruning rules and jumping rules to do a simple pruning, with the aim of jumping over many places that only considered necessary. From the 5 levels of testing that has been conducted, the result shows that JPS algorithm can reduce the amount of nodes that will be used to find the path with the range of 533% until 1700% shorter compared to A* algorithm.

Keywords: Pathfinding, Non Player Character, Shortest Path, A* Algorithm, Jump Point Search Algorithm

DAFTAR ISI

Halaman

HALAMAN JUDUL	i
LEMBAR PENGESAHAN TUGAS AKHIR	ii
TANDA LULUS UJIAN SIDANG TUGAS AKHIR	iii
HALAMAN PERNYATAAN.....	iv
MOTTO DAN PERSEMBAHAN.....	v
KATA PENGANTAR	vi
ABSTRAKSI	vii
ABSTRACT	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xii
DAFTAR TABEL	xvi

BAB I PENDAHULUAN

1.1 Latar Belakang.....	I-1
1.2 Rumusan Masalah	I-4
1.3 Tujuan Penelitian.....	I-4
1.4 Manfaat.....	I-4
1.5 Batasan Masalah.....	I-5

BAB II TINJAUAN PUSTAKA

2.1 Algoritma A* (A-Star)	II-1
2.1.1 <i>Jump Point Search</i>	II-7
2.1.2 Fungsi Heuristik <i>Manhattan Distance</i>	II-9
2.2 <i>Non Player Character</i>	II-10
2.3 <i>Side Scrolling</i>	II-11
2.4 Metode Pengembangan Perangkat Lunak	II-12
2.4.1 <i>Agile Software Development</i>	II-13
2.4.2 <i>Scrum</i>	II-14

2.4.2.1 <i>Scrum Role</i>	II-14
2.4.2.2 <i>Development Process</i>	II-15

BAB III ANALISIS PERANCANGAN

3.1 Analisis Perangkat Lunak	III-1
3.1.1 Analisis Masalah	III-1
3.1.1.1 Analisis Data	III-2
3.1.1.2 Analisis Unit-Unit Permainan	III-2
3.1.1.3 Analisis Algoritma A*	III-5
3.1.1.4 Analisis <i>Jump Point Search</i>	III-6
3.1.2 Rekayasa Perangkat Lunak.....	III-6
3.1.2.1 Deskripsi Umum Sistem	III-6
3.1.2.2 Analisis Kebutuhan Perangkat Lunak.....	III-8
3.2 Perancangan Perangkat Lunak.....	III-9
3.2.1 Pemodelan <i>Use Case</i>	III-9
3.2.1.1 Diagram <i>Use Case</i>	III-9
3.2.1.2 Definisi <i>Use Case</i>	III-10
3.2.1.3 Definisi Aktor	III-11
3.2.1.4 Skenario <i>Use Case</i>	III-12
3.2.3 Diagram Sekuensial.....	III-20
3.2.4 Diagram Kelas	III-26
3.2.5 Perancangan Antarmuka.....	III-28
3.2.5.1 Antarmuka Permainan.....	III-28

BAB IV IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Perangkat Lunak	IV-1
4.1.1 Lingkungan Implementasi	IV-1
4.1.2 Implementasi Kelas	IV-2
4.1.3 Implementasi Antarmuka	IV-7
4.2 Pengujian Perangkat Lunak	IV-9
4.2.1 Lingkungan Pengujian.....	IV-10

4.2.2 Rencana Pengujian Tiap <i>Use Case</i>	IV-10
4.2.3 Kasus Uji Tiap Use Case.....	IV-13
4.2.4 Pengujian Pathfinding	IV-20
4.2.5 Analisa Hasil Pengujian	IV-25

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan	V-1
5.2 Saran	V-2

DAFTAR PUSTAKA.....xv

LAMPIRAN.....L-1

DAFTAR GAMBAR

	Halaman
Gambar II-1 Contoh Pencarian Jalur Terpendek	II-3
Gambar II-2 Hasil Langkah Awal.....	II-4
Gambar II-3 Hasil Langkah 4,5, dan 6.....	II-5
Gambar II-4 Hasil Langkah 4,5, dan 6 yang di ulangi.....	II-5
Gambar II-5 Jalur solusi yang ditemukan	II-6
Gambar II-6 Ilustrasi <i>Straight Move</i>	II-8
Gambar II-7 Ilustrasi <i>Diagonal Move</i>	II-8
Gambar II-8 <i>Super Mario Bros</i>	II-12
Gambar II-9 <i>Castlevania 2D: Symphony of the Night</i>	II-12
Gambar II-10 Alur Proses <i>Scrum</i>	II-16
Gambar III-1 Ilustrasi Karakter Naga	III-2
Gambar III-2 Ilustrasi Karakter Manusia Pada Permainan	III-3
Gambar III-3 Ilustrasi <i>Obstacle 1</i>	III-3
Gambar III-4 Ilustrasi <i>Obstacle 2</i>	III-4
Gambar III-5 Ilustrasi <i>Obstacle 3</i>	III-4
Gambar III-6 Ilustrasi <i>Obstacle 4</i>	III-4
Gambar III-7 Ilustrasi <i>Obstacle 5</i>	III-4
Gambar III-8 Ilustrasi <i>Obstacle 6</i>	III-5
Gambar III-9 Ilustrasi <i>Obstacle 7</i>	III-5
Gambar III-10 <i>Flowchart</i> Algoritma A*	III-7
Gambar III-11 <i>Flowchart</i> Algoritma A* dan JPS.....	III-8

Gambar III-12	Diagram <i>Use Case</i>	III-10
Gambar III-13	Diagram Analisis Memainkan Permainan Dengan Algoritma A*	III-17
Gambar III-14	Diagram Analisis Memainkan Permainan Dengan Algoritma JPS	III-18
Gambar III-15	Diagram Analisis <i>Pause</i> Permainan.....	III-18
Gambar III-16	Diagram Analisis Menjalankan Simulasi Algoritma A*	III-19
Gambar III-17	Diagram Analisis Menjalankan Simulasi Algoritma A* dan JPS	III-19
Gambar III-8	Ilustrasi <i>Obstacle 6</i>	III-5
Gambar III-9	Ilustrasi <i>Obstacle 7</i>	III-5
Gambar III-10	<i>Flowchart</i> Algoritma A*	III-7
Gambar III-11	<i>Flowchart</i> Algoritma A* dan JPS.....	III-8
Gambar III-12	Diagram <i>Use Case</i>	III-10
Gambar III-12	Diagram Analisis Memainkan Permainan Dengan Algoritma A*	III-17
Gambar III-14	Diagram Analisis Memainkan Permainan Dengan Algoritma JPS	III-18
Gambar III-15	Diagram Analisis <i>Pause</i> Permainan.....	III-18
Gambar III-16	Diagram Analisis Menjalankan Simulasi Algoritma A*	III-19
Gambar III-17	Diagram Analisis Menjalankan Simulasi Algoritma A.....	III-19
Gambar III-18	Diagram Analisis Menampilkan Bantuan Permainan.....	III-20

Gambar III-19	Diagram Sekuensial Memainkan Permainan Dengan Algoritma A*	III-21
Gambar III-20	Diagram Sekuensial Memainkan Permainan Dengan Algoritma JPS	III-10
Gambar III-21	Diagram Analisis <i>Pause</i> Permainan.....	III-23
Gambar III-22	Diagram Sekuensial Menjalankan Simulasi Algoritma A*	III-24
Gambar III-23	Diagram Sekuensial Menjalankan Simulasi Algoritma JPS...	III-25
Gambar III-24	Diagram Sekuensi Menampilkan Bantuan.....	III-23
Gambar III-25	Diagram Kelas	III-27
Gambar III-26	Gambar Antarmuka Menu Utama.....	III-28
Gambar III-27	Gambar Antarmuka Memainkan Permainan	III-29
Gambar III-28	Gambar Antarmuka <i>Pause</i> Permainan.....	III-30
Gambar III-29	Gambar Antarmuka Menu Simulasi	III-30
Gambar III-30	Gambar Antarmuka Simulasi	III-31
Gambar III-31	Gambar Antarmuka Help.....	III-31
Gambar IV-1	Antarmuka Menu Utama	IV-7
Gambar IV-2	Gambar Antarmuka Memainkan Permainan	IV-8
Gambar IV-3	Gambar Antarmuka Pause Permainan.....	IV-8
Gambar IV-4	Gambar Antarmuka Menu Simulasi	IV-8
Gambar IV-5	Gambar Antarmuka Simulasi	IV-9
Gambar IV-6	Gambar Antarmuka <i>Help</i>	IV-9
Gambar IV-7	Hasil Pencarian A* Pada Level 0	IV-21
Gambar IV-8	Hasil Pencarian A* Pada Level 1	IV-21

Gambar IV-9 Hasil Pencarian A* Pada Level 2	IV-22
Gambar IV-10 Hasil Pencarian A* Pada Level 3	IV-22
Gambar IV-11 Hasil Pencarian A* Pada Level 4	IV-23
Gambar IV-12 Hasil Pencarian JPS Pada Level 0	IV-23
Gambar IV-13 Hasil Pencarian JPS Pada Level 1	IV-24
Gambar IV-14 Hasil Pencarian JPS Pada Level 2	IV-24
Gambar IV-15 Hasil Pencarian JPS Pada Level 3	IV-25
Gambar IV-16 Hasil Pencarian JPS Pada Level 4	IV-25
Gambar IV-17 Grafik Pengujian Jarak	IV-27
Gambar IV-18 Grafik Pengujian Cost.....	IV-27

DAFTAR TABEL

	Halaman
Tabel III-1 Tabel Kebutuhan Fungsional.....	III-9
Tabel III-2 Kebutuhan Nonfungsional.....	III-9
Tabel III-3 Definisi <i>Use Case</i>	III-10
Tabel III-4 Definisi Aktor	III-11
Tabel III-5 Skenario <i>Use Case</i> Memainkan Permainan Dengan Algoritma A*	III-13
.....	III-13
Tabel III-6 Skenario <i>Use Case</i> Memainkan Permainan Dengan Algoritma JPS.....	III-15
.....	III-15
Tabel III-7 Skenario <i>Use Case</i> Menjalankan Simulasi Algoritma A.....	III-15
Tabel III-8 Skenario <i>Use Case</i> Menjalankan Simulasi Algoritma JPS.....	III-15
Tabel IV-1 Implementasi Perangkat Keras	IV-1
Tabel IV-2 Implementasi Kelas dalam Bahasa Pemrograman C#.....	IV-2
Tabel IV-3 Rencana Pengujian <i>Use Case</i> Memainkan Permainan Dengan Algoritma A*	IV-10
Tabel IV-4 Rencana Pengujian <i>Use Case</i> Memainkan Permainan Dengan Algoritma JPS	IV-11
Tabel IV-5 Rencana Pengujian <i>Use Case</i> Melakukan Pause Permainan.....	IV-12
Tabel IV-6 Rencana Pengujian <i>Use Case</i> Menjalankan Simulasi Algoritma A*	IV-12
Tabel IV-7 Rencana Pengujian <i>Use Case</i> Menjalankan Simulasi Algoritma JPS	IV-13

Tabel IV-8 Rencana Pengujian <i>Use Case</i> Menampilkan Bantuan Permainan	IV-13
.....	
Tabel IV-9 Rencana Pengujian <i>Use Case</i> Memainkan Permainan Dengan Algoritma A*	IV-13
Tabel IV-10 Rencana Pengujian <i>Use Case</i> Memainkan Permainan Dengan Algoritma JPS	IV-16
Tabel IV-11 Rencana Pengujian <i>Use Case</i> Melakukan Pause Permainan.....	IV-18
Tabel IV-12 Rencana Pengujian <i>Use Case</i> Menjalankan Simulasi Algoritma A*	IV-19
Tabel IV-13 Rencana Pengujian <i>Use Case</i> Menjalankan Simulasi Algoritma JPS	IV-19
Tabel IV-14 Rencana Pengujian <i>Use Case</i> Menampilkan bantuan Permainan	IV-20
.....	
Tabel IV-5 Pengujian Pathfinding	IV-26

BAB I

PENDAHULUAN

1.1. Latar Belakang

Non Player Character (NPC) adalah salah satu komponen terpenting dalam sebuah permainan, dan bahkan terkadang jalannya sebuah cerita dalam permainan dapat sangat di pengaruhi oleh hal ini. NPC memainkan peranan penting dalam banyak permainan, menghadirkan alur cerita dan berfungsi sebagai pemberi misi untuk pemain dalam permainan (Al-Gharaibeh & Jeffery, 2010). Pada banyak permainan diperlukan penentuan jalur yang baik untuk dapat mengontrol satu atau lebih karakter *NPC* agar dapat mendeteksi keberadaan musuh dan bergerak menuju musuh tersebut tanpa harus terhenti oleh hambatan yang ada sehingga perilaku dari *NPC* akan lebih terlihat alami.

Side Scrolling didefinisikan sebagai permainan dimana pengguna mengontrol karakter dalam permainan untuk dapat bergerak dari satu sisi layar ke sisi lain. Karakter pengguna bergerak menuju tepi layar (biasanya dari kiri ke kanan), pergeseran *playfield* atau lingkungan permainan di arah tertentu (biasanya kiri) untuk menyesuaikan gerakan dari karakter yang digunakan atau dimainkan. Dalam permainan bergenre *side scrolling* dibutuhkan penerapan algoritma *pathfinding* yang bertujuan untuk menemukan rute atau jalur terpendek antara dua buah titik dari suatu lingkungan yang pada umumnya memiliki hambatan – hambatan dari lingkungan tersebut. Salah satu algoritma yang dapat digunakan untuk penyelesaian masalah *pathfinding* adalah Algoritma A*.

Algoritma A* merupakan metode algoritma yang digunakan dalam melakukan pencarian jalur yang optimal yang menghubungkan dua titik pada lingkungan (grafik) dari permainan yang ada (Hermawan & Bellanar, 2015). Seperti dalam penelitian Taufiq, Wibowo, & Septiana (2015) algoritma A* telah berhasil diimplementasikan untuk menentukan jalur dengan *multiple goal* pada pergerakan NPC menggunakan algoritma A* yang dapat menghasilkan jalur *complete* dan optimal pada kasus *multiple destination*, sehingga NPC dapat menentukan jalur mana yang akan di lalui untuk mencapai seluruh tujuannya.

Penelitian lainnya oleh Suhaili, Irawan, Fahrizal, & Herusutopo (2014) yang melakukan analisis perbandingan algoritma *pathfinding Greedy Best-First Search* dengan A* dalam menentukan lintasan peta dimana dalam pengujian yang dilakukan mendapatkan kesimpulan bahwa algoritma *greedy best-first search* lebih unggul dalam waktu pencarian, sedangkan algoritma A* lebih unggul dalam pencarian lintasan terpendek yang dilalui. Hal ini diakibatkan oleh *heuristic cost* dari setiap node yang dilalui saja yang dilihat oleh algoritma *greedy best-first search*, sedangkan *heuristic cost* dan *actual cost* dari setiap *node* yang dilalui dihitung oleh algoritma A*, sehingga waktu yang dibutuhkan algoritma A* lebih lama dari pada algoritma *Greedy Best-first search* dalam menentukan lintasan yang dilalui. Namun algoritma A* dapat memberikan lintasan yang lebih optimal daripada algoritma *greedy best-first search* dengan menghasilkan panjang lintasan optimal sebanyak 31,32 % , panjang lintasan yang sama antara algoritma A* dan *Greedy Best-first Search* sebanyak 68,68%, dan panjang lintasan optimal dari *Greedy Best-first Search* sebanyak 0%.

Algoritma A* memiliki beberapa variasi untuk mengatasi penyelesaian waktu pencarian yang lebih lama, diantaranya adalah *Iterative Deepening A** (IDA*), *Partial Expansion A** (PEA*), *Jump point search* (JPS), dimana dalam penelitian yang dilakukan oleh Delima, Indrajaya, Takaredase, Dhian, & Rachmat (2016) tentang Perbandingan Penerapan Algoritma A*, IDA* *Jump Point Search*, dan PEA* Pada Permainan Pacman. Dalam penelitian dibahas mengenai perbandingan kinerja dari keempat algoritma tersebut pada permainan Pacman, dan yang menjadi tolak ukur perbandingan melalui jumlah *open node*, *visited node*, dan panjang *path* yang dipilih oleh agen cerdas. Dari pengujian yang telah dilakukan, didapatkan hasil bahwa algoritma *Jump Point Search* memiliki rata-rata *visited node* terkecil yaitu 11,7, yang diikuti oleh algoritma IDA dengan 15,57, sementara dua algoritma lain A* dan PEA memiliki rata-rata jumlah *visited node* yang hampir sama yaitu 27,8 dan 28,63. Untuk analisis panjang *path*, didapatkan data panjang *path* yang sama, hal ini menunjukkan bahwa tiga varian A* memiliki karakteristik optimal dari A*. Untuk analisa *open list* didapatkan data bahwa jumlah *open list* antara A* dengan PEA* menunjukkan rata-rata yang hampir sama yaitu 37,5 dan 38,4. Hal ini menunjukkan bahwa kemampuan PEA untuk mengurangi jumlah *open list* tidak dapat berjalan dengan efektif pada permainan Pacman. Berdasarkan hasil implementasi dan analisis sistem pada permainan *customed* pacman dengan ukuran *grid map* 19x22 *pixels* disimpulkan bahwa algoritma *Jump Point Search* mampu menghasilkan rata-rata jumlah *visited node* terbaik dari keempat algoritma, algoritma PEA kurang sesuai untuk diimplementasikan dalam permainan pacman karena berdasarkan nilai heuristik setiap *node* yang tidak terlalu signifikan dan

sedikitnya jumlah percabangan dalam permainan, dan keempat algoritma memiliki optimalitas menemukan solusi yang sama.

Dalam penelitian ini, pengimplementasian algoritma A* dengan optimasi JPS akan diterapkan untuk pencarian jalur terpendek pada NPC.

1.2. Rumusan Masalah

Bagaimana menentukan jalur terpendek pada *Non Player Character* menggunakan algoritma A* dan algoritma *Jump Point Search*?

1.3. Tujuan Penelitian

1. Melakukan analisis terhadap algoritma A* dan *Jump Point Search* dalam melakukan pencarian jalur terpendek pada *Non Player Character*.
2. Mengetahui jarak jalur terpendek pada *Non Player Character* dengan menggunakan algoritma A*.
3. Mengetahui jarak jalur terpendek pada *Non Player Character* dengan menggunakan algoritma A* dan *Jump Point Search*.
4. Membandingkan jarak jalur terpendek pada *Non Player Character* dengan menggunakan algoritma A* dengan algoritma A* dan *Jump Point Search*.

1.4. Manfaat Penelitian

1. Menentukan jalur terpendek bagi NPC untuk menemukan *player*
2. Mendapatkan perbedaan jarak jalur terpendek pencarian algoritma A* dengan algoritma A* dan JPS pada pencarian jalur bagi NPC untuk menemukan *player*

3. Hasil dari penelitian pencarian jalur terpendek pada *Non Player Character* dapat digunakan pada penelitian pencarian jalur terpendek selanjutnya.

1.5. Batasan Masalah

Batasan masalah dalam penelitian ini adalah :

1. Perangkat lunak permainan berbasis *desktop*
2. Perangkat lunak permainan tidak membutuhkan sambungan internet
3. Perangkat lunak permainan dengan grafis 2D
4. Menggunakan fungsi heuristik *manhattan distance*
5. Menggunakan satuan ukuran node untuk mengukur jarak

DAFTAR PUSTAKA

- Al-Gharaibeh, J., & Jeffery, C. (2010, March 20-24). Portable Non-Player Character Tutors with Quest Activities. *IEEE Virtual Reality*.
- Behrisch, M. e. (2011). SUMO - Simulation of Urban Mobility. *The Third International Conference on Advances in System Simulation (SIMUL 2011)*.
- BioWare. (2002, June 18). Neverwinter Nights. Atari.
- Delima, R., Indrajaya, G. T., Takaredase, A. K., Dhian, I., & Rachmat, A. (2016). Perbandingan Penerapan Algoritma A*, IDA*, Jump Point Search, dan PEA* Pada Permainan Pacman. *Jurnal Teknik Informatika dan Sistem Informasi*.
- Galih, B., Delima, R., & Gandang, S. (2013, Oktober). Analisis Implementasi Algoritma A*(A-Star) Pada Game RPG (Role Playing Game) 3D Sebagai Dasar Pergerakan NPC(Non-Player Character) Mendekati Player Untuk Meningkatkan Realitas Game World. *INFORMATIKA Vol. 9, No.2 Oktober 2013, 9*.
- Harabor, D. (2012). Fast Pathfinding via Symmetry Breaking.
- Harabor, D., & Grastien, A. (2011). Online Graph Pruning for Pathfinding on Grid Map. *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*.
- Hermawan, L., & Bellanar, M. (2015). Pencarian Rute Oleh Non Player Character Menggunakan Algoritma A* Berbasis 2D. *Seminar Nasional Teknologi Informasi dan Komunikasi Terapan (SEMANTIK) 2015*.
- Konami Computer Entertainment Tokyo. (1997, March 20). Castlevania: Symphony of the Night. Konami.
- Nexon. (2003). MapleStory. South Korea: Nexon.
- Nintendo. (1985, September 13). Super Mario Bros. Nintendo.
- Novak, J. (2011). *Game Development Essentials: An Introduction* (3 ed.). Cengage Learning. Diambil kembali dari <https://books.google.co.id/books?id=r36qpCjBIuMC>
- Parlindungan, J. R. (2011). Penerapan Algoritma A* dalam Penentuan Lintasan. *Makalah IF2091 Struktur Diskrit – Sem. I Tahun 2011/2012*.
- Pressman, R. S. (2010). *Software Engineering: A Practitioner's Approach* (7th ed.). Raghathan Srinivasan.

- Rare Ltd. (1991, June). *Battletoads*.
- Riwinoto, & Alfian. (2015). Implementasi Pathfinding dengan Algoritma A* pada Game Funny English Menggunakan Unity 3D Berbasis Graf Navmesh.
- Sherwood, T., Perelman, E., Hamerly, G., & Calder, B. (2002). Automatically Characterizing Large Scale Program Behavior.
- Sommerville, I. (2011). *Software Engineering, 9th Edition*. Boston, Massachusetts, Unites States of America: Pearson Education.
- Suhaili, C. Y., Irawan, M., Fahrizal, R. M., & Herusutopo, A. (2014). Analisis Perbandingan Algoritma Pathfinding Greedy Best-first Search Dengan A*(Star) Dalam Menentukan Lintasan Pada Peta. *IF Working Paper 001*.
- Suyanto. (2014). *Artificial Intelligence : Searching, Reasoning, Planning dan Learning* (2 ed.). Bandung: INFORMATIKA.
- Taufiq, P. J., Wibowo, A. T., & Septiana, G. (2015). IMPLEMENTASI DAN ANALISIS ALGORITMA A*(STAR) UNTUK MENENTUKAN JALUR DENGAN MULTIPLE GOAL PADA PERGERAKAN NPC(NON-PLAYABLE CHARACTER).
- Tilawah, H. (2011). Penerapan Algoritma A-star (A*) Untuk Menyelesaikan Masalah Maze. *Makalah IF3051 Strategi Algoritma*.
- Tong, M. (2001, mei 22). Side Scrollers: A Planer Odyssey. *History of Computer Game Design*.
- Traish, J., Tulip, J., & Moore, W. (2015). Optimization using Boundary Lookup Jump Point Search. *IEEE Transactions on Computational Intelligence and AI in Games*. doi:10.1109/TCIAIG.2015.2421493
- White Wolf Publishing, Onyx Path Publishing, By Night Studios. (1991). *Vampire: The Masquerade*.