

**PEMILIHAN JALUR TERPENDEK PADA *GAME* CANDI BOROBUDUR  
MENGUNAKAN ALGORITMA A\***

**SKRIPSI**

**Oleh:  
ANNISA DINAR FARAZIZAH  
NIM. 16650065**



**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2021**

**PEMILIHAN JALUR TERPENDEK PADA *GAME* CANDI BOROBUDUR  
MENGUNAKAN ALGORITMA A\***

**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:  
ANNISA DINAR FARAZIZAH  
NIM. 16650065**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2021**

**HALAMAN PERSETUJUAN**

**PEMILIHAN JALUR TERPENDEK PADA *GAME* CANDI BOROBUDUR  
MENGUNAKAN ALGORITMA A\***

**SKRIPSI**

**Oleh:**  
**ANNISA DINAR FARAZIZAH**  
**NIM. 16650065**

Telah Diperiksa dan Disetujui untuk Diuji:  
Tanggal: 17 Desember 2021

Pembimbing I,



Yunifa Miftachul Arif, M.T  
NIP. 19830616 201101 1 004

Pembimbing II,



Roro Inda Melani, M.T., M.Sc  
NIP. 19780925 200501 2 008

Mengetahui,  
Ketua Jurusan Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Dr. Fachrul Kurniawan, M.MT  
NIP. 19771020 200912 1 001

**HALAMAN PENGESAHAN**

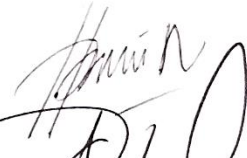



**PEMILIHAN JALUR TERPENDEK PADA *GAME* CANDI BOROBUDUR  
MENGUNAKAN ALGORITMA A\***

**SKRIPSI**

**Oleh:**  
**ANNISA DINAR FARAZIZAH**  
**NIM. 16650065**

Telah Dipertahankan di Depan Dewan Penguji Skripsi  
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan  
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)  
Tanggal: 17 Desember 2021

**Susunan Dewan Penguji**

Penguji Utama	: <u>Hani Nurhayati, M.T</u> NIP. 19780625 200801 2 006	(  )
Ketua Penguji	: <u>Fresy Nugroho, M.T</u> NIP. 19710722 201101 1 001	(  )
Sekretaris Penguji	: <u>Yunifa Miftachul Arif, M.T</u> NIP. 19830616 201101 1 004	(  )
Anggota Penguji	: <u>Roro Inda Melani, M.T, M.Sc</u> NIP. 19780925 200501 2 008	(  )

Mengetahui,  
Ketua Jurusan Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang

  
Dr. Fachrul Kurniawan, M.MT  
NIP. 19771020 200912 1 001

NIP. 19771020 200912 1 001  
**HALAMAN PERNYATAAN KEASLIAN TULISAN**

Saya yang bertanda tangan dibawah ini:

Nama : Annisa Dinar Farazizah

NIM : 16650065

Jurusan : Teknik Informatika

Fakultas : Sains dan Teknologi

Judul Skripsi : **PEMILIHAN JALUR TERPENDEK PADA *GAME* CANDI  
BOROBUDUR MENGGUNAKAN ALGORITMA A\***

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 17 Desember 2021  
Penulis,



Annisa Dinar Farazizah  
NIM. 16650065

## **MOTTO**

~ *Kecantikan bukan dengan pakaian yang menghiasi kita, tetapi kecantikan adalah kebagusan ilmu dan etika.*

*-Abu Hurairah RA-*

~ *Barang siapa yang menginginkan dunia maka mesti dengan ilmu, barang siapa yang menginginkan akhirat mesti dengan ilmu, dan barang siapa yang menginginkan kedua-duanya maka mesti pula dengan ilmu*

*-Imam Syafi'i-*

## HALAMAN PERSEMBAHAN

Alhamdulillah, puji dan syukur kehadiran Allah SWT yang telah memberikan nikmat dan hidayah sehingga penulis bisa menyelesaikan pendidikan S1 jurusan Teknik Informatika di Universitas Islam Negeri Maulana Malik Ibrahim Malang. Shalawat serta salam selalu tercurahkan kepada junjungan Nabi Agung Muhammad SAW yang telah membimbing umatnya dari jaman jahiliyah menuju jaman yang terang benderang dan dipenuhi dengan ilmu pengetahuan seperti saat ini.

Terima kasih kepada Ibu dan Ayah yang selalu membuatku termotivasi dan selalu memberikan kasih sayang, mendoakanku, serta menasehatiku menjadi lebih baik. Terima kasih Ibu dan Ayah atas semua yang telah engkau berikan semoga Ibu dan Ayah diberi kesehatan dan panjang umur agar dapat menemani langkahku bersama adikku Syadad dalam mencapai kesuksesan.

Untuk Bapak Yunifa Miftachul Arif, M.T dan Ibu Roro Inda Melani, M.T, M.Sc selaku pembimbing, terima kasih sudah membimbing serta memberikan arahan dengan tulus, sabar, dan ikhlas dalam penyelesaian skripsi ini. Tidak lupa ucapan terima kasih kepada Bapak Juniardi Nur Fadila, M.T yang telah membantu serta memberikan arahan dalam pengerjaan skripsi ini saat mengalami permasalahan *device*.

Skripsi ini juga tidak akan selesai tanpa adanya dukungan dari Mas Aldi dan juga anakku tercinta Ghifarini. Terima kasih atas dukungan, kebaikan, serta perhatian kalian dalam pengerjaan skripsi ini dengan cara yang sangat menghibur.

Tidak lupa ucapan terima kasih kepada teman seperjuangan ANDROMEDA TI'16 yang selalu meluangkan waktu serta memberikan semangat dalam penyelesaian skripsi. Begitupun teman-teman serta semua pihak yang tidak bisa disebutkan satu persatu. Semangat untuk teman-temanku yang masih berjuang untuk mengerjakan tugas akhir ini. Terima kasih teman-teman tercinta semoga kita terus terhubung melalui doa yang mengiringi kesuksesan kita. Aamiin.

## KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Alhamdulillah, puji dan syukur penulis panjatkan kepada Allah SWT Dzat yang hanya kepada-Nya memohon pertolongan, rahmat, dan kasih sayang sehingga penulis dapat menyelesaikan penyusunan skripsi yang berjudul Pemilihan Jalur Terpendek pada *Game* Candi Borobudur Menggunakan Algoritma A\*. Shalawat serta salam kepada Rasulullah SAW yang senantiasa menjadi sumber inspirasi dan teladan terbaik untuk umat manusia.

Penulis menyadari perjalanan panjang telah penulis lalui dalam rangka menyelesaikan penulisan skripsi ini. Banyak kendala yang dihadapi, namun berkat bantuan, bimbingan, kerjasama dari berbagai pihak dan berkah dari Allah SWT sehingga kendala-kendala yang dihadapi tersebut dapat diatasi sehingga penulis berhasil menyelesaikan penulisan skripsi ini.

Selanjutnya penulis mengucapkan terima kasih dan penghargaan yang setinggi-tingginya kepada semua pihak yang telah membantu dalam proses penulisan skripsi ini diantaranya adalah:

1. Prof. Dr. H. M. Zainuddin, MA selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim.
2. Dr. Sri Hariani, M.Si selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim.
3. Dr. Fachrul Kurniawan, M.MT selaku ketua jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim.

4. Yunifa Miftachul Arif, M.T, selaku dosen pembimbing I yang telah memberikan bimbingan, arahan dan masukan kepada penulis dalam pengerjaan skripsi hingga akhir.
5. Roro Inda Melani, M.T, M.Sc, selaku dosen pembimbing II yang telah memberikan masukan kepada penulis dalam pengerjaan skripsi.
6. Segenap dosen Teknik Informatika yang telah mendidik dan memberikan ilmu selama kuliah di Universitas Islam Negeri Maulana Malik Ibrahim Malang dan seluruh staf yang selalu sabar melayani segala administrasi selama proses penelitian ini.
7. Serta semua pihak yang telah banyak membantu dalam pengerjaan skripsi ini baik secara langsung maupun tidak langsung.

Penulis menyadari bahwa penelitian ini masih jauh dari kata sempurna. Oleh karena itu, segala kritik dan saran penulis harapkan demi tercapainya hal terbaik dalam penelitian ini. Besar harapan penulis, semoga penelitian ini dapat memberikan manfaat sekaligus menambah pengetahuan bagi berbagai pihak. Amin.  
Wassalamu' alaikum Wr. Wb.

Malang, 17 Desember 2021

Penulis

## DAFTAR ISI

<b>HALAMAN JUDUL .....</b>	<b>i</b>
<b>HALAMAN PERSETUJUAN.....</b>	<b>ii</b>
<b>HALAMAN PENGESAHAN.....</b>	<b>iii</b>
<b>HALAMAN PERNYATAAN KEASLIAN TULISAN .....</b>	<b>iv</b>
<b>MOTTO .....</b>	<b>v</b>
<b>HALAMAN PERSEMBAHAN.....</b>	<b>vi</b>
<b>KATA PENGANTAR.....</b>	<b>viii</b>
<b>DAFTAR ISI.....</b>	<b>x</b>
<b>DAFTAR GAMBAR.....</b>	<b>xii</b>
<b>DAFTAR TABEL.....</b>	<b>xiii</b>
<b>ABSTRAK .....</b>	<b>xiv</b>
<b>ABSTRACT .....</b>	<b>xv</b>
<b>مستخلص البحث.....</b>	<b>xvi</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Pernyataan Masalah.....	4
1.3 Tujuan Penelitian.....	4
1.4 Batasan Penelitian .....	4
1.5 Manfaat Penelitian.....	5
<b>BAB II STUDI PUSTAKA .....</b>	<b>6</b>
2.1 Penelitian Terkait.....	6
2.2 Simulasi .....	9
2.3 Algoritma A*.....	10
2.4 AHP (Analytical Hierarchy Process).....	13
2.5 <i>Tools</i> Pembuat <i>Game</i> .....	16
2.5.1 Unity 3D .....	16
2.5.2 Blender.....	16
<b>BAB III DESAIN DAN IMPLEMENTASI .....</b>	<b>17</b>
3.1 Pengumpulan Data.....	17
3.2 Pembuatan Desain <i>Asset</i> 3D dan UI.....	18
3.3 Desain <i>Interface</i> .....	19
3.3.1 Desain 3D .....	22
3.3.2 FSM ( <i>Finish State Machine</i> ).....	25
3.3 Rencana Skenario <i>Game</i> .....	25
3.4 Perhitungan A* .....	29

3.5 Penerapan AHP (Analytical Hierarchy Process) .....	35
3.5.1 AHP Variabel Tetap.....	37
3.5.2 AHP Variabel Tidak Tetap .....	40
3.6 Proses <i>Scoring</i> .....	41
3.6 Rencana Pengujian .....	43
<b>BAB IV IMPLEMENTASI DAN PENGUJIAN .....</b>	<b>45</b>
4.1 Implementasi .....	45
4.1.1 Perangkat Keras yang Digunakan.....	45
4.1.2 Perangkat Lunak yang Digunakan.....	45
4.1.3 Implementasi Skenario <i>Game</i> .....	46
4.2 Pengujian A* Pada <i>Game</i> .....	51
4.3 Pengujian AHP Pada <i>Game</i> .....	57
4.4 Integrasi Islam .....	63
<b>BAB V PENUTUP .....</b>	<b>65</b>
5.1 Kesimpulan.....	65
5.2 Saran .....	65
<b>DAFTAR PUSTAKA</b>	

## DAFTAR GAMBAR

Gambar 2. 1 Struktur Hierarki AHP (Ansah et al, 2015).....	13
Gambar 3. 1 Desain Penelitian.....	17
Gambar 3. 2 Proses Desain UI.....	19
Gambar 3. 3 Menu Utama.....	19
Gambar 3. 4 Pemilihan Mode.....	20
Gambar 3. 5 Mode Algoritma Penerapan A*.....	20
Gambar 3. 6 <i>Starting Point</i> .....	21
Gambar 3. 7 <i>Minimap</i> .....	21
Gambar 3. 8 <i>Gameplay</i> .....	22
Gambar 3. 9 <i>Player</i> Sampai Tujuan.....	22
Gambar 3. 10 Proses Pembuatan Karakter dengan <i>Software</i> Blender 3D.....	23
Gambar 3. 11 Desain 3D Mobil.....	23
Gambar 3. 12 Desain 3D Motor.....	24
Gambar 3. 13 Desain 3D Kota.....	24
Gambar 3. 14 <i>Flow Chart</i> Skenario <i>Game</i> .....	26
Gambar 3. 15 <i>Indeks</i> Poin Terminal Madu.....	30
Gambar 3. 16 <i>Indeks</i> Poin Kebon Agung.....	33
Gambar 3. 17 Diagram Blok Proses AHP.....	36
Gambar 4. 1 <i>Main Menu</i> .....	46
Gambar 4. 2 Pemilihan Mode.....	47
Gambar 4. 3 Mode Penerapan Algoritma A*.....	47
Gambar 4. 4 Pemilihan Kendaraan.....	48
Gambar 4. 5 Pemilihan <i>Starting Point</i> .....	49
Gambar 4. 6 Doa Sebelum Perjalanan.....	49
Gambar 4. 7 Pemilihan Rute.....	50
Gambar 4. 8 Animasi <i>Traveling</i> .....	50
Gambar 4. 9 <i>Player</i> Mencapai Tujuan Akhir.....	51
Gambar 4. 10 Visualisasi Jalur Rute dari <i>Starting Point</i> Terminal Madu.....	52
Gambar 4. 11 Grafik Total Jarak dari Terminal Madu.....	54
Gambar 4. 12 Visualisasi Jalur Rute dari <i>Starting Point</i> Kebon Agung.....	55
Gambar 4. 13 Grafik Total jarak dari Kebon Agung.....	57
Gambar 4. 14 Grafik Perbandingan <i>Score</i> Skenario dengan <i>Score Game</i> .....	63

## DAFTAR TABEL

Tabel 2. 1 <i>Comparative Judgement</i> .....	14
Tabel 2. 2 Matriks Perbandingan Berpasangan.....	15
Tabel 2. 3 Nilai Random Indeks ( <i>RI</i> ) .....	16
Tabel 3. 1 Perhitungan Algoritma A* Jalur 1 Terminal Madu .....	31
Tabel 3. 2 Perhitungan Algoritma A* Jalur 2 Terminal Madu .....	31
Tabel 3. 3 Perhitungan Algoritma A* Jalur 3 Terminal Madu .....	32
Tabel 3. 4 Perhitungan Algoritma A* Jalur 1 Kebon Agung.....	34
Tabel 3. 5 Perhitungan Algoritma A* Jalur 2 Kebon Agung.....	34
Tabel 3. 6 Perhitungan Algoritma A* Jalur 3 Kebon Agung.....	35
Tabel 3. 7 Matrik Perbandingan Berpasangan .....	37
Tabel 3. 8 Normalisasi Matriks Keputusan.....	37
Tabel 3.9 Matrik Perbandingan Alternatif Terhadap Kecepatan. ....	38
Tabel 3.10 Normalisasi Matrik Perbandingan Alternatif Terhadap Kecepatan. ...	38
Tabel 3.11 Matrik Perbandingan Alternatif Terhadap Kenyamanan. ....	38
Tabel 3. 12 Normalisasi Matrik Perbandingan Alternatif Terhadap Kenyamanan. ....	38
Tabel 3. 13 Matrik Perbandingan Alternatif Terhadap Konsumsi BBM. ....	39
Tabel 3. 14 Normalisasi Matrik Perbandingan Alternatif Terhadap Konsumsi BBM .....	39
Tabel 3. 15 List Kondisi Awal Permainan .....	40
Tabel 3. 16 Tabel Bobot Akhir AHP .....	41
Tabel 4. 1 Tabel Uji Pencarian Jalur dari Terminal Madu.....	53
Tabel 4. 2 Tabel Uji Pencarian Jalur dari Kebon Agung .....	56
Tabel 4. 3 Tabel Uji Metode AHP .....	59

## ABSTRAK

Farazizah, Annisa Dinar. 2021. **Pemilihan Jalur Terpendek pada *Game Candi Borobudur Menggunakan Algoritma A\****. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing (I) Yunifa Miftachul Arif, MT, (II) Roro Inda Melani, M.T, M.Sc.

---

Kata Kunci: *Game* Simulasi, Algoritma A\* (*A-Star*), AHP (*Analytical Hierarchy Process*), *Pathfinding*, Rekomendasi.

Kemacetan yang ada di sepanjang rute perjalanan Candi Borobudur menjadi masalah untuk wisatawan yang belum pernah berkunjung. Permasalahan ini dapat diatasi dengan *game* simulasi, simulasi dapat memberikan pengalaman hampir menyerupai keadaan aslinya. Pada *game* ini terdapat algoritma A\* (*A-Star*) untuk *pathfinding* dan metode AHP (*Analytical Hierarchy Process*) untuk rekomendasi konfigurasi berdasarkan beberapa kondisi. Uji coba menggunakan algoritma A\* dilakukan 20 kali di 2 *starting point* yang berbeda dan menghasilkan nilai akurasi yang sangat baik yaitu 100% untuk keduanya. Pengujian menggunakan metode AHP dilakukan dengan simulasi perjalanan dengan kondisi yang berbeda sebanyak 54 kali dan menghasilkan nilai error sebesar 0,095% dan nilai akurasi yang didapatkan dari rata rata error adalah sebesar 99,905%.

## ABSTRACT

Farazizah, Annisa Dinar. 2021. **Selection of the Shortest Path in the Borobudur Temple Game Using the A\* Algorithm**. Undergraduate Thesis. Department of Informatics Engineering Faculty of Science and Technology . Islamic State University of Maulana Malik Ibrahim Malang. Counselor (I) Yunifa Miftachul Arif, MT, (II) Roro Inda Melani, M.T, M.Sc.

---

Keyword: Simulation Game, A\* (A-Star) Algorithm, AHP (Analytical Hierarchy Process), Pathfinding, Recommendation.

Traffic jam along the route of Borobudur Temple is a problem for tourists who have never visited. This problem can be solved with simulation games, simulations can provide an experience that is almost like the real situation. In this game there is the A\* (A-Star) algorithm for pathfinding and the AHP (Analytical Hierarchy Process) method for configuration recommendations based on several conditions. The experiment using the A\* algorithm was carried out 20 times at 2 different starting points and produced a very good accuracy value of 100% for both. Testing using the AHP method is carried out by simulating trips with different conditions as much as 54 times and producing an error value of 0.095% and the accuracy value obtained from the average error is 99.905% .

## مستخلص البحث

فرزينة، أنيسة دينار. ٢٠٢١. خيار أقصر المسار في لعبة معبد أي هيكل بروبودور باستخدام خوارزمية  $A^*$  (A-Star). مقالة  
جامعية. قسم هندسة المعلوماتية بكلية العلوم و التكنولوجيا جامعة الدولة الإسلامية مولانا مالك إبراهيم مالانج. المشرف :  
(الأول) يونيفا مفتاح العارف، ماجستير. (الثاني) رورو إيند ميلاني، ماجستير.

---

الكلمات الرئيسية : لعبة المحاكاة، خوارزمية  $A^*$  (A-Star) ، (عملية التسلسل الهرمي التحليلي) AHP، إكتشاف المسار،  
التوصيات.

الإزدحام الذي وجد على طول طرق بروبودور للسياح الذين لم يسبق لهم الزيارة. يمكن بألعاب المحاكاة، و يتمكّن هذه  
المحاكاة بتوفير تجربة تشبه الوضع الحقيقي تقريبا. في هذه اللعبة، توجد خوارزمية  $A^*$  (A-Star) لتحديد المسار و طريقة (عملية  
التسلسل الهرمي التحليلي) AHP لتوصيات التكوين بناءً على عدّة شروط. تم إجراء التجربة باستخدام خوارزمية  $A^*$  ٢٠ مرة في  
نقطتي بداية مختلفتين و انتجت قيمة دقّة جيّدة جدًّا بنسبة ١٠٠٪ لكلّيهما. و تم إجراء الإختبارات باستخدام طريقة AHP عن  
طريق محاكاة الرحلات بشروط مختلفة ٥٤ مرّة و نتج قيمة ٠،٠٩٥٪ و قيمة الدقّة التي تم الحصول عليها من متوسطة الخطأ ٠،٩٩،٩٠٥٪.

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Setiap daerah di Indonesia memiliki beberapa objek wisata yang menarik salah satunya adalah Candi Borobudur. Candi Borobudur adalah candi peninggalan Buddha terbesar di dunia yang dibangun pada abad ke-9. Candi Borobudur mampu menjadi magnet yang menarik jutaan wisatawan setiap tahunnya dan menjadi wisata yang wajib dikunjungi saat berwisata ke Yogyakarta, meskipun letaknya berada di Magelang Jawa Tengah. Banyak wisatawan yang datang pada hari biasa maupun hari libur untuk melihat monumen Buddha terbesar di dunia, terlebih candi ini pernah menjadi 7 keajaiban dunia.

Indonesia adalah negara berkembang yang memiliki banyak kota dengan destinasi wisata yang potensial. Beberapa di antaranya belum menerapkan konsep manajemen pariwisata dengan benar. Destinasi pariwisata adalah daerah yang memiliki pengaruh signifikan terhadap peningkatan perekonomian di kota-kota pariwisata (Arif et al, 2019).

Dalam ayat suci Al-Qur'an terdapat beberapa seruan untuk melakukan wisata seperti pada firman Allah SWT dalam surat Al-Mulk ayat 15 yang berbunyi:

هُوَ الَّذِي جَعَلَ لَكُمُ الْأَرْضَ ذَلُولًا فَامْشُوا فِي مَنَاكِبِهَا وَكُلُوا مِن رِّزْقِهِ وَإِلَيْهِ النُّشُورُ ١٥

*“Dialah yang menjadikan bumi untuk kamu dalam keadaan mudah dimanfaatkan. Maka, jelajahilah segala penjurunya dan makanlah sebagian dari rezeki-Nya. Hanya kepada-Nya kamu (kembali setelah) dibangkitkan” (QS. Al-Mulk:15).*

Tafsir Jalalain:

هُوَ الَّذِي جَعَلَ لَكُمْ الْأَرْضَ

“(Dialah yang menjadikan bumi itu mudah bagi kalian) mudah untuk dipakai berjalan di atas permukaannya”

ذُلُولًا فَأَمْشُوا فِي مَنَاكِبِهَا

“(maka berjalanlah di segala penjurunya) pada semua arahnya”

وَكُلُوا مِنْ رِزْقِهِ

“(dan makanlah sebagian dari rezeki-Nya) yang sengaja diciptakan untuk kalian”

وَالِيهِ النُّشُورُ

“(dan hanya kepada-Nyalah kalian dibangkitkan) dari kubur untuk mendapatkan pembalasan”

Kepadatan jalur di Yogyakarta yang merupakan pusat wisata di Jawa menjadikan jalur menuju objek wisata ini memiliki kepadatan lalu lintas yang tinggi. Kepadatan lalu lintas yang tinggi di rute menuju objek wisata menjadi salah satu masalah bagi wisatawan yang belum pernah berkunjung ke objek wisata tersebut, karena hal ini bisa mengakibatkan terbuangnya waktu di perjalanan. Perjalanan yang dibutuhkan dari kota Yogyakarta ke Candi Borobudur kurang lebih 60 menit dengan kendaraan pribadi maupun kendaraan umum. Terdapat dua *starting point* menuju Candi Borobudur, dengan pemilihan *starting point* yang tepat waktu dalam perjalanan dapat diminimalisir dengan pemilihan jalur dengan kepadatan lalu lintas yang rendah.

Dengan adanya permasalahan kepadatan jalur menuju objek wisata maka perlu adanya optimalisasi jalur *guide tour* menuju Candi Borobudur. Dalam

optimasi ini ada beberapa faktor yang bisa dijadikan indikator dalam pemilihan jalur diantaranya, kepadatan jalur, jenis kendaraan, jarak, dan waktu.

Alternatif yang bisa digunakan untuk mengatasi kejadian yang menyerupai aslinya adalah simulasi. Simulasi memberikan pengalaman interaksi seperti keadaan sebenarnya kepada *user*, diharapkan pengalaman yang di dapat di simulasi dapat bermanfaat saat user berada di situasi yang sama. Simulasi jalur menuju Candi Borobudur bisa menjadi salah satu solusi dalam permasalahan ini, dengan adanya simulasi masyarakat bisa mengetahui visualisasi jalur sebenarnya. Pemilihan *game* simulasi sebagai media pemilihan rute ini bisa digunakan sebagai referensi Dinas Perhubungan atau instansi lainnya untuk mengatur lalu lintas dan bisa digunakan sebagai pemilihan jalur wisata untuk *user*. Hal ini bisa memudahkan *user* ketika berkunjung ke tempat tersebut suatu saat karena sudah familiar dengan jalur perjalanan berdasarkan *starting point* yang dipilih.

Pemilihan simulasi sebagai media pemilihan jalur membutuhkan kecerdasan di dalamnya, kecerdasan ini digunakan untuk penentuan jalur terbaik. Simulasi ini menggunakan algoritma A\* (A-Star). Algoritma A\* adalah algoritma pencarian *graf* yang mencari jalur dari titik awal yang diberikan menuju titik tujuan. Algoritma A\* ini memberikan solusi yang optimal dan komplit, optimal disini dapat menghasilkan rekomendasi jalur terbaik yaitu jalur terpendek dan komplit disini mencapai tujuan yang diharapkan. Hasil dari penelitian menggunakan algoritma A\* ini akan dikombinasikan lagi dengan beberapa faktor seperti kepadatan jalur, jenis kendaraan dan waktu menggunakan metode AHP (*Analytic Hierarchy Process*). Metode AHP mampu mengatasi kompleksitas dari beberapa

faktor yang ada sehingga menghasilkan alternatif terbaik berdasarkan tujuan serta menyediakan skala pengukuran untuk menghasilkan prioritas.

## 1.2 Pernyataan Masalah

Berikut pernyataan masalah yang peneliti buat berdasarkan latar belakang yang telah dipaparkan diatas.

1. Seberapa tinggi *accuracy* rekomendasi jalur yang dihasilkan dari algoritma A\*?
2. Apakah algoritma A\* yang dikombinasikan dengan AHP dapat menghasilkan pemilihan jalur terbaik menuju Candi Borobudur?

## 1.3 Tujuan Penelitian

Berikut tujuan penelitian berdasarkan pernyataan masalah yang telah dipaparkan.

1. Untuk mengukur *accuracy* pemilihan jalur yang dihasilkan dari permainan simulasi menggunakan algoritma A\*.
2. Untuk menganalisa algoritma A\* yang dikombinasikan dengan AHP dalam pemilihan jalur terbaik menuju Candi Borobudur.

## 1.4 Batasan Penelitian

Batasan dalam penelitian adalah sebagai berikut :

1. Penelitian yang dilakukan adalah pencarian jalur yang tepat menuju wisata Candi Borobudur dari *starting point* yang telah ditentukan menggunakan peta Yogyakarta.
2. Menggunakan metode A\* untuk pencarian rute tercepat.

3. Menggunakan metode AHP untuk pencarian kendaraan yang paling tepat berdasarkan kondisi perjalanan.
4. Input kriteria yang dibutuhkan pada penelitian ini yaitu jenis kendaraan, kenyamanan, kecepatan, kondisi cuaca, kepadatan lalu lintas, *budget* perjalanan, dan jarak.
5. Output dari *game* ini adalah nilai *score* yang menggunakan skala 1-5 bintang bergantung pada pencapaian pemain.
6. *Starting point* yang digunakan yaitu, Terminal Madu dan Kebon Agung
7. Data jalur dan jarak tempuh dari titik n ke titik tujuan diperoleh dari *Google Maps*.

### 1.5 Manfaat Penelitian

Penelitian dengan algoritma A\* berbasis AHP diharapkan dapat menghasilkan *accuracy* jalur terpendek menuju Candi Borobudur yang tepat. Penelitian ini diharapkan dapat memberikan manfaat untuk beberapa pihak antara lain:

1. *User game*, dengan adanya permainan simulasi ini *user* bisa mengetahui jalur mana yang akan dipilih ketika berkunjung ke wisata Candi Borobudur berdasarkan pilihan yang dihasilkan dari permainan ini.
2. Manfaat bagi *developer game*, algoritma yang ada pada permainan simulasi ini dapat menjadi masukan untuk dikembangkan dalam pembuatan *game* yang lebih kompleks. Begitu juga algoritma yang digunakan pada permainan simulasi ini bisa dijadikan pembanding untuk pengembangan *game* yang telah dibuat.

## BAB II

### STUDI PUSTAKA

#### 2.1 Penelitian Terkait

Berikut beberapa penelitian terkait yang dijadikan referensi dalam penelitian ini.

- a. Pencarian Jalur Terpendek pada *Snake Game* menggunakan Algoritma A\*.

Penelitian yang dilakukan oleh Masri dan Ari Tri Mukti pada tahun 2014 menjelaskan bahwa algoritma A\* adalah algoritma pencarian jalur terbaik untuk pencarian jalur terpendek dengan hasil perhitungan yang kecil dari simpul awal menuju simpul tujuan. Pada penelitian ini algoritma A\* teruji efektifitasnya dalam mendapatkan makanan dengan jalur terpendek.

Kekurangan pada penelitian ini adalah hanya dicari restaurant terdekat dengan titik awal *user*, namun tidak dilakukan spesifikasi dari *user* menginginkan jenis makanan seperti apa. Pada penelitian yang dilakukan pada skripsi ini memanfaatkan metode AHP untuk mengatasi kekurangan tersebut, dengan AHP dapat memungkinkan *user* mendapatkan restaurant terdekat dengan menu makanan yang diinginkan.

- b. Penerapan Algoritma A Star untuk Penentuan Jarak Terdekat Wisata Kuliner di Kota Bandar Lampung.

Penelitian yang dilakukan oleh Purnama et al menjelaskan bahwa pada tahun 2016 Dinas Pariwisata dan Ekonomi Kreatif Provinsi Lampung mencatat kurang lebih 194 rumah makan yang terdapat di Bandar Lampung. Minimnya

informasi yang akurat mengenai jalur terdekat menuju lokasi wisata kuliner dan informasi wisata kuliner yang terdekat membuat masyarakat mengalami kesulitan dalam mencari dan menentukan wisata kuliner mana yang tepat. Permasalahan di atas dapat diselesaikan dengan algoritma A\* (A Star). Hasil dari penelitian ini adalah aplikasi *mobile* android yang memberikan informasi mengenai jalur terdekat menuju lokasi wisata kuliner dan juga informasi mengenai wisata kuliner yang terdapat di sekitar.

Kekurangan pada penelitian ini adalah hanya dicari restaurant terdekat dengan titik awal *user*, namun tidak dilakukan spesifikasi dari *user* menginginkan jenis makanan seperti apa. Pada penelitian yang dilakukan pada skripsi ini memanfaatkan metode AHP untuk mengatasi kekurangan tersebut, dengan AHP dapat memungkinkan user mendapatkan restaurant terdekat dengan menu makanan yang diinginkan.

c. Pencarian Jalur Terpendek Pengiriman Barang Menggunakan Algoritma A\* Studi Kasus Kantor Pos Besar Medan.

Penelitian yang dilakukan oleh Alfarisi menyatakan bahwa kurir pengiriman barang yang ada di kantor pos melakukan pengiriman barang dari kantor pos pusat ke kantor pos cabang yang ada di kota Medan dengan melalui jalur yang sama. Untuk meminimalisir jarak dan tujuan yang akan dituju maka diperlukan pencarian jalur terpendek dari kantor pos pusat ke beberapa kantor pos cabang yang ada di kota Medan. Pada penelitian ini digunakan algoritma A\* untuk pencarian jalur terpendek, dengan menggunakan algoritma ini diharapkan kurir pengiriman barang yang ada di kantor pos pusat bisa lebih

cepat menemukan jalur mana yang paling singkat dan cepat untuk mencapai tujuan pada jalur pengiriman barang ke beberapa kantor pos cabang Medan.

Penelitian ini memiliki kekurangan hanya menemukan jalan tercepat dengan menggunakan metode A\* namun pada kenyataan di lapangan, tidak selalu jalur terpendek menjadi pilihan terbaik dan tercepat, tetapi ada faktor lain seperti kepadatan lalu lintas, cuaca, dan banyak faktor lain yang mempengaruhi. Kekurangan tersebut dapat diselesaikan dengan menggunakan metode AHP yang sudah diaplikasikan pada skripsi ini, yang diharapkan selain mendapatkan jarak terdekat, aplikasi ini juga bisa memberikan rekomendasi berdasarkan faktor-faktor tertentu.

- d. Aplikasi Sistem Pencarian Halte BRT Terdekat Kota Semarang Menggunakan Metode A\* Berbasis Android.

Penelitian yang dilakukan oleh Abimanyu dan Muljono telah berhasil mengembangkan sistem informasi halte terdekat di kota Semarang menggunakan Algoritma A\*. Penelitian ini menyatakan bahwa BRT (*Bus Rapid Transit*) adalah salah satu transportasi umum yang dapat digunakan di Kota Semarang. Pada penelitian ini digunakan algoritma yang digunakan *Best First Search* yang menggabungkan *Uniform Cost Search* dan *Greedy Best-First Search*. Dimana harga yang dipertimbangkan  $f(n)$  didapat dari harga sesungguhnya  $g(n)$  ditambah dengan harga perkiraan  $h(n)$ . Aplikasi yang dihasilkan memberikan informasi halte terdekat dengan menggunakan algoritma A\*. Perhitungan yang dihasilkan menggunakan algoritma A\* dengan

uji coba sebanyak 20 kali pada lokasi yang berbeda menghasilkan nilai akurasi 100% dalam menentukan halte naik dan turun terdekat.

Kekurangan pada penelitian ini hanya menentukan jarak terdekat terhadap halte bis, namun tidak menyajikan berapa lama waktu yang dibutuhkan user untuk menunggu kedatangan bis pada halte terpilih. Kekurangan tersebut dapat diselesaikan dengan kolaborasi A\* dengan AHP untuk menemukan rekomendasi halte dengan jarak terdekat dari *user* dan waktu tunggu yang paling singkat.

## 2.2 Simulasi

Menurut Shanon (1975) simulasi adalah operasi peniruan dari proses kejadian yang terjadi di dunia nyata yang mensimulasikan sesuatu hal, pertamanya membutuhkan model yang dikembangkan, model ini mewakili karakteristik atau kebiasaan tertentu atau sistem abstrak atau proses. Menurut Simon & Emshoff (1970) simulasi merupakan model sistem yang komponennya diuraikan dengan proses aritmatika dan logika untuk memperkirakan sifat-sifat dinamis sistem yang dijalankan oleh komputer.

Menurut Udin Syaefudin Sa'ud (2005) simulasi adalah sebuah replikasi atau visualisasi dari perilaku sebuah sistem dan juga sebuah model yang berisi seperangkat variabel yang menampilkan ciri utama dari sistem kehidupan yang sebenarnya. Simulasi memungkinkan keputusan-keputusan yang menentukan bagaimana ciri-ciri utama itu bisa dimodifikasi secara nyata.

Dikutip dari Harrel et al (2004) di dalam buku *Simulation Using Promodel* bahwa selain dapat membuat desain untuk mengambil keputusan simulasi juga dapat digunakan untuk melakukan validasi terhadap keputusan yang dilakukan.

### 2.3 Algoritma A\*

Menurut Suyanto (2014) algoritma A\* merupakan algoritma komputer yang digunakan secara luas dalam mencari jalur (*pathfinding*) dan grafik melintang (*graph traversal*), proses *plotting* sebuah jalur melintang secara efisien antara titik-titik, disebut *node*.

Menurut Russel & Norvig (2003) algoritma A Star (A\*) memiliki lima komponen utama, yaitu: *node* awal, *node* goal, *open list*, *closed list* dan *cost*. *Node* awal merupakan titik awal dari posisi saat ini, sedangkan *node goal* merupakan titik akhir atau dapat juga disebut titik tempat tujuan. *Cost* merupakan nilai dari jarak yang telah ditempuh untuk sampai ke tempat tujuan. *Open list* ini berupa sebuah *priority queue*, dimana setiap *node* yang masuk pertama akan dikeluarkan pertama dengan syarat tertentu. *Closed list* ini berupa sebuah *stack* (tumpukan), dimana *node* yang terakhir dimasukkan akan dikeluarkan pertama kali. Selain sebagai penampung *node* yang telah dilewati, *closed list* ini juga digunakan untuk mendapatkan jarak terdekat saat *node goal* sudah dicapai.

Algoritma A\* merupakan salah satu algoritma pencarian jalur yang umum digunakan dalam menentukan jalur terpendek dari titik awal ke titik tujuan. Pencarian jalur pada graf sering sekali digunakan dalam permainan simulasi. Secara matematis, fungsi evaluasi dan estimasi pada A\* dapat dijabarkan seperti pada persamaan (1).

$$f(n) = g(n) + h(n) \quad (1)$$

Keterangan:

$f(n)$ : biaya estimasi terendah

$g(n)$ : biaya dari *node* awal ke *node* n

$h(n)$ : perkiraan biaya dari *node* n ke *node* akhir

A\* memperhitungkan *cost* dari *current state* ke tujuan dengan fungsi heuristik. Algoritma ini juga mempertimbangkan *cost* yang telah ditempuh selama ini dari *initial state* ke *current state*. Jadi jika ada jalan yang telah ditempuh sudah terlalu panjang dan ada jalan lain yang memiliki *cost* lebih kecil tetapi memberikan posisi yang sama dilihat dari *goal*, jalan yang lebih pendek yang akan dipilih. Prinsip algoritma ini adalah mencari jalur terpendek dari sebuah simpul awal (*starting point*) menuju simpul tujuan dengan memperhatikan harga (F) terkecil. Terminologi dasar algoritma A\* terdiri dari:

- A. *Node* awal adalah *node* mula-mula.
- B. *Node* merupakan vertek-vertek sebagai representasi dari area *pathfinding*.
- C. *Open list* adalah tempat menyimpan data simpul yang akan dilakukan pengecekan.
- D. *Closed list* adalah tempat menyimpan data simpul yang telah dilakukan pengecekan dari *open list* yang memiliki nilai terkecil.
- E. *Current Node* merupakan simpul yang saat ini digunakan untuk mengecek, yang berasal dari nilai f terkecil di *open list*.
- F. *Node* tujuan adalah *node* yang dituju.
- G. Halangan adalah sebuah *node* yang tidak dapat dilalui.

Heuristik sendiri bisa diartikan sebagai suatu fungsi yang memberikan nilai berupa biaya perkiraan (estimasi) dari suatu solusi. Teknik pencarian heuristik (*heuristic searching*) merupakan suatu strategi untuk melakukan proses pencarian secara selektif dan dapat memandu proses pencarian yang memiliki kemungkinan sukses paling besar, namun dengan kemungkinan mengorbankan kelengkapan (*completeness*). Untuk menerapkan pencarian heuristik diperlukan suatu fungsi heuristik. Fungsi heuristik adalah aturan-aturan yang digunakan untuk mendapatkan solusi yang diinginkan. Terdapat beberapa fungsi heuristik yang umumnya digunakan dalam perhitungan algoritma A\*, salah satunya adalah *Euclidean distance*.

*Euclidean distance* adalah perhitungan jarak dari 2 buah titik dalam *Euclidean space*. *Euclidean space* diperkenalkan oleh Euclid, seorang matematikawan dari Yunani untuk mempelajari hubungan antara sudut dan jarak. Secara matematis jarak Euclidean dapat dijabarkan pada persamaan (2).

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2)$$

Keterangan:

$d$  : biaya estimasi terendah

$x_1$  : latitude<sub>1</sub>

$x_2$  : latitude<sub>2</sub>

$y_1$  : longitude<sub>1</sub>

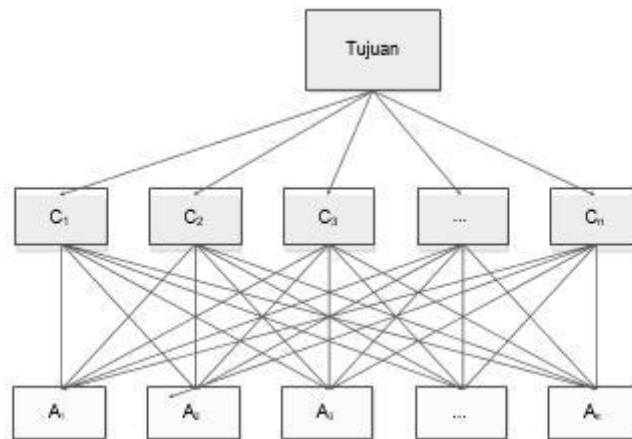
$y_2$  : longitude<sub>2</sub>

## 2.4 AHP (Analytical Hierarchy Process)

*Analytical Hierarchy Process* (AHP) diperkenalkan oleh Thomas L. Saaty pada tahun 1970, untuk menyelesaikan masalah multikriteria yang kompleks menjadi suatu hirarki. Tiga prinsip yang harus dipahami untuk menyelesaikan suatu permasalahan dengan metode AHP, yaitu: *decomposition*, *comparative judgement*, dan *logical consistency*.

### 1. *Decomposition*

*Decomposition* adalah memecah persoalan yang utuh menjadi unsur-unsur yang saling berhubungan, dalam bentuk struktur hirarki seperti tampak pada gambar berikut:



Gambar 2. 1 Struktur Hierarki AHP (Ansah et al, 2015)

### 2. *Comparative Judgement*

*Comparative judgement* dilakukan dengan memberikan penilaian tentang kepentingan relatif dua elemen pada suatu tingkat tertentu dalam kaitannya dengan tingkatan di atasnya, untuk memberikan urutan prioritas dari elemen-elemennya. Hasil penilaian ini akan lebih mudah disajikan dalam bentuk

matriks *pairwise comparison* atau matriks perbandingan berpasangan. Dalam membentuk *pairwise*, Saaty (1980) menetapkan skala kuantitatif 1 sampai dengan 9 untuk menilai perbandingan tingkat kepentingan suatu elemen terhadap elemen lain seperti pada tabel 2. 1 di bawah ini.

Tabel 2. 1 *Comparative Judgement*

Intensitas Kepentingan	Keterangan
1	Kedua elemen sama pentingnya
3	Elemen yang satu sedikit lebih penting daripada elemen yang lainnya
5	Elemen yang satu lebih penting daripada elemen yang lainnya
7	Satu elemen jelas lebih mutlak penting daripada elemen lainnya
9	Satu elemen mutlak penting daripada elemen lainnya
2, 4, 6, 8	Nilai-nilai anatar dua nilai pertimbangan yang berdekatan
Kebalikan	Jika untuk aktivitas i mendapat satu angka dibanding dengan aktivitas j, maka j mempunyai nilai kebalikannya dibanding dengan i

### 3. *Logical Consistency*

*Logical consistency* merupakan karakteristik penting AHP. Konsistensi memiliki dua makna, yang pertama adalah bahwa obyek-obyek yang serupa dapat dikelompokkan sesuai dengan keseragaman dan relevansi dan yang kedua adalah menyangkut tingkat hubungan antara objek-objek yang didasarkan pada kriteria tertentu.

Tahapan dalam proses perhitungan menggunakan metode AHP antara lain:

1. Menentukan prioritas kriteria dengan menyusun perbandingan berpasangan  
Perbandingan antar alternatif untuk sub sistem hirarki dapat dibuat dalam bentuk matriks  $n \times n$  seperti pada tabel 2.2 di bawah ini.

Tabel 2. 2 Matriks Perbandingan Berpasangan

C	A1	A2	...	A <sub>n</sub>
A1	a <sub>11</sub>	a <sub>12</sub>	...	a <sub>1n</sub>
A2	a <sub>21</sub>	a <sub>22</sub>	...	a <sub>2n</sub>
...	...	...	...	...
A <sub>m</sub>	a <sub>m1</sub>	a <sub>m2</sub>	...	a <sub>mn</sub>

## 2. Normalisasi Matriks Keputusan

Untuk menentukan nilai dari masing-masing matriks  $m \times n$  maka, nilai setiap kolom matriks dibagi dengan nilai total kolom. Setelah itu, menentukan rata-rata matriks pada setiap baris. Nilai total dari setiap kolom matriks harus berjumlah 1.

## 3. Uji Konsistensi

Penentuan konsistensi dari matriks didasarkan pada *eigen value* maksimum yang diperoleh dengan persamaan (3).

$$CI = \frac{\lambda_{\max} - n}{n-1} \quad (3)$$

$CI$  : Rasio penyimpangan konsistensi

$\lambda_{\max}$  : Nilai *eigen* terbesar dari matriks berordo  $n$

$n$  : Orde matriks

Jika nilai  $CI = 0$ , maka matriks *pairwise comparison* tersebut konsisten. Batas *inconsistency* yang telah ditetapkan oleh Thomas L. Saaty ditentukan dengan menggunakan Rasio Konsistensi ( $CR$ ), yaitu perbandingan indeks konsistensi dengan random indeks ( $RI$ ). Rasio konsistensi terdapat dalam persamaan (4).

$$CR = \frac{CI}{RI} \quad (4)$$

$CR$  : Rasio konsistensi

$CI$  : Rasio penyimpangan konsistensi

$RI$  : Random indeks

Nilai  $RI$  dapat dihitung dengan persamaan (5) atau dengan melihat tabel 2.3 di bawah ini.

$$RI = \frac{1.98 \times (n-1)}{n} \quad (5)$$

Tabel 2. 3 Nilai Random Indeks ( $RI$ )

N	1	2	3	4	5	6	7	8	9	10
$RI$	0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.48

## 2.5 Tools Pembuat Game

### 2.5.1 Unity 3D

Menurut Sudarmilah et al (2013) unity 3D merupakan suatu aplikasi untuk membuat *game*, simulasi, dan bangunan arsitektur. Dalam unity 3D, terdapat banyak fitur yang menunjang pembuatan suatu *game* menjadi menarik, diantaranya seperti bahasa pemrograman yang dapat digunakan antara lain Boo, javascript, dan C#. Selain itu unity 3D adalah *game* multi *platform*, sehingga dapat digunakan untuk PC, Mac, Android, *browser*, dan lainnya. Fitur lainnya yang terdapat pada unity 3D seperti *sky box*, *rendering*, *lighting*, *sound effect*, *physic game*.

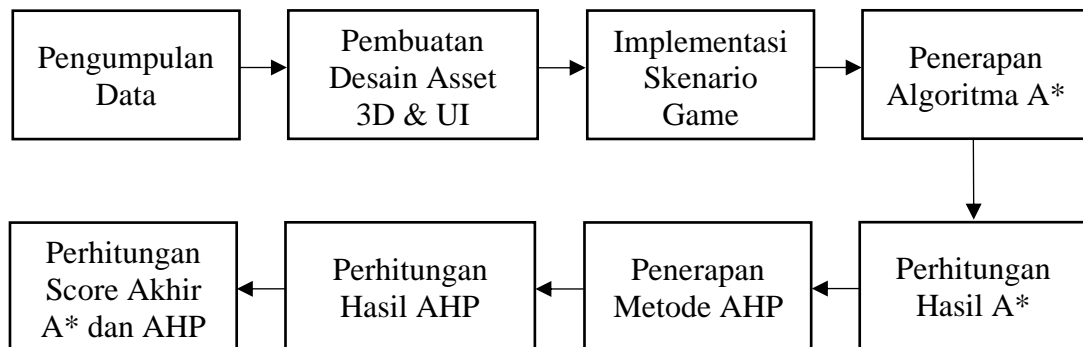
### 2.5.2 Blender

Menurut Ardianto et al (2012) blender merupakan aplikasi yang digunakan untuk membuat perancangan suatu multimedia khususnya 3 dimensi, diantaranya seperti *asset* sebuah *game*, animasi 3D, *visual effect* dan model cetak 3D. Blender merupakan *software open source*, sehingga aplikasi ini dapat digunakan untuk beberapa *Operation System* di antaranya Mac, Windows, dan Linux.

## BAB III

### DESAIN DAN IMPLEMENTASI

Pada bab ini dijelaskan beberapa tahapan dalam desain sistem dan implementasi yang digunakan pada pembuatan *game*.



Gambar 3. 1 Desain Penelitian

Gambar 3.1 menggambarkan alur desain dalam penelitian ini. Seperti penjelasan sebelumnya, jalur menuju Candi Borobudur memiliki tingkat kepadatan yang tinggi karena banyak wisatawan domestik dan mancanegara yang berkunjung setiap harinya. Masalah yang ditimbulkan adalah kemacetan maka dari itu perlu adanya rekomendasi jalur yang tepat berdasarkan jarak, kepadatan jalur, waktu dan jenis kendaraan. Kepadatan jalur akan mempengaruhi estimasi waktu kedatangan menuju Candi Borobudur, dalam penelitian ini digunakan algoritma A\* dan AHP untuk mengatasi permasalahan tersebut.

#### 3.1 Pengumpulan Data

Pengumpulan data dilakukan selama dua hari di Yogyakarta dan Magelang, data yang didapat digabungkan dengan data dari Google *Maps*. Dari

pengumpulan data diperoleh dua *starting point* yaitu Terminal Madu dan Kebon Agung. Data yang digunakan sebagai parameter pemilihan jalur antara lain:

- *Starting point* sebagai posisi awal perhitungan
- Candi Borobudur sebagai tujuan akhir
- Kepadatan lalu lintas yang dilewati antara posisi awal sampai tujuan
- Jarak antara posisi awal sampai tujuan akhir
- Jenis kendaraan yang digunakan
- Budget Perjalanan

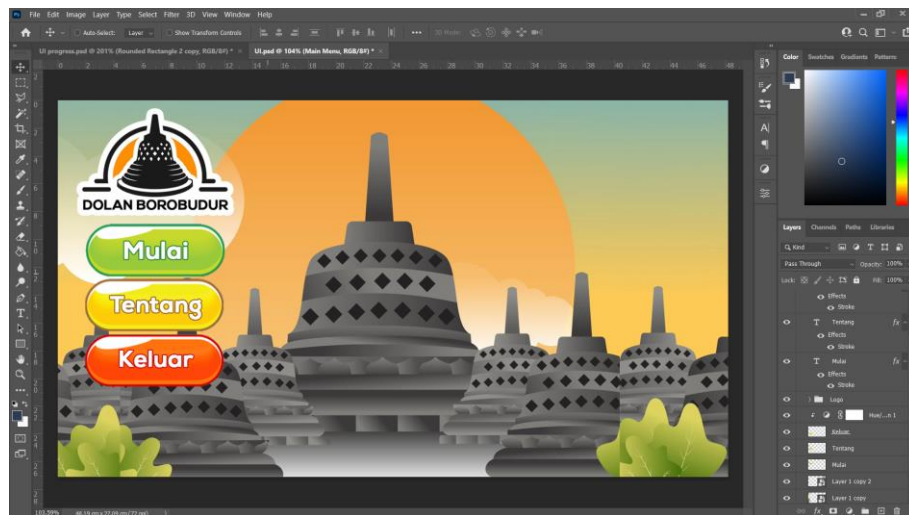
### 3.2 Pembuatan Desian Asset 3D dan UI

Data-data yang sudah diperoleh akan dibuat sebuah *game* simulasi dengan tujuan memvisualisasikan keadaan jalur menuju Candi Borobudur dari beberapa *starting point* dengan *minimap* yang telah disesuaikan pada *game*. Permainan ini bercerita tentang seseorang yang ingin berlibur ke Candi Borobudur tetapi belum mengetahui pemilihan jalur mana yang tepat dari *starting point* yang diinginkan. Tugas dari *player* yaitu memilih *starting point* yang diinginkan, kemudian *player* akan diberi petunjuk jalur.

Ketika *player* tiba di lokasi tujuan maka akan diberikan pilihan, melanjutkan memilih jalur lain dari *starting point* yang sama maupun yang berbeda atau menekan tombol *finish* untuk mengakhiri permainan, jika *player* memilih jalur baru *player* akan kembali ke awal *game* untuk memilih *starting point* mana yang diinginkan.

### 3.3 Desain *Interface*

Desain *interface*, berupa *user interface* yang ada dalam permainan ini. Dalam proses desain menggunakan *software* Photoshop CC 2019. Gambar di bawah ini adalah proses pembuatan UI.



Gambar 3. 2 Proses Desain UI

Gambar 3.2 menggambarkan proses desain UI dalam pembuatan sistem.



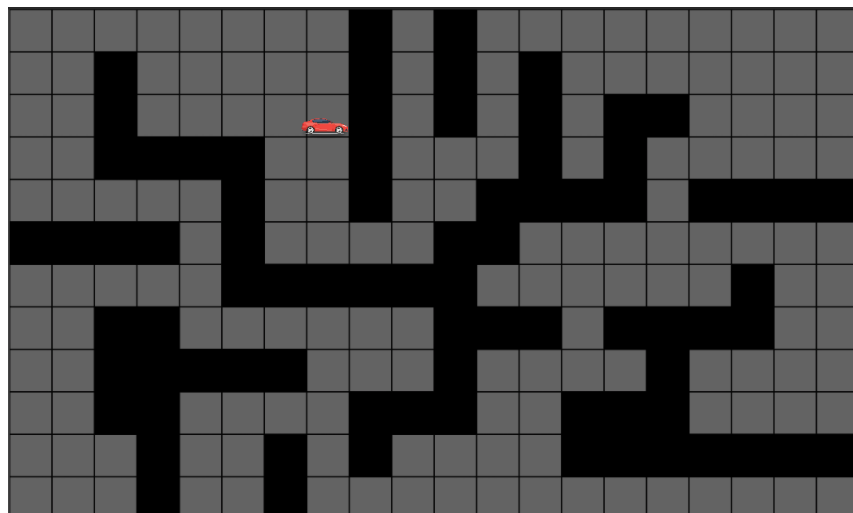
Gambar 3. 3 Menu Utama

Gambar 3.3 merupakan menu utama *game* ini.



Gambar 3. 4 Pemilihan Mode

Pada gambar 3.4 *player* bisa memilih menggunakan mode *game* atau mode penerapan A\*.



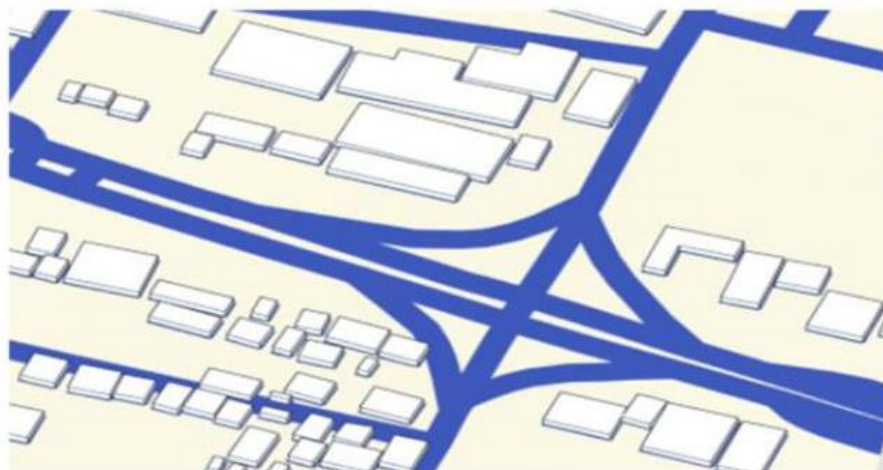
Gambar 3. 5 Mode Algoritma Penerapan A\*

Gambar 3.5 adalah mode penerapan A\* yang ada dalam *game*. Tampilan ini akan muncul jika *player* memilih mode penerapan A\* di awal permainan.



Gambar 3. 6 *Starting Point*

Gambar 3.6 adalah pemilihan *starting point* oleh *player* diharuskan memilih titik awal keberangkatan agar sistem bisa memberikan petunjuk jalur yang akan dilalui untuk mencapai tujuan.



Gambar 3. 7 *Minimap*

Gambar 3.7 adalah *minimap* yang terlihat setelah *player* memilih titik awal keberangkatan.



Gambar 3. 8 *Gameplay*

Gambar 3.8 *gameplay* setelah *player* menentukan titik awal keberangkatan. *Player* berjalan mengikuti petunjuk dari navigasi sampai menjumpai persimpangan.



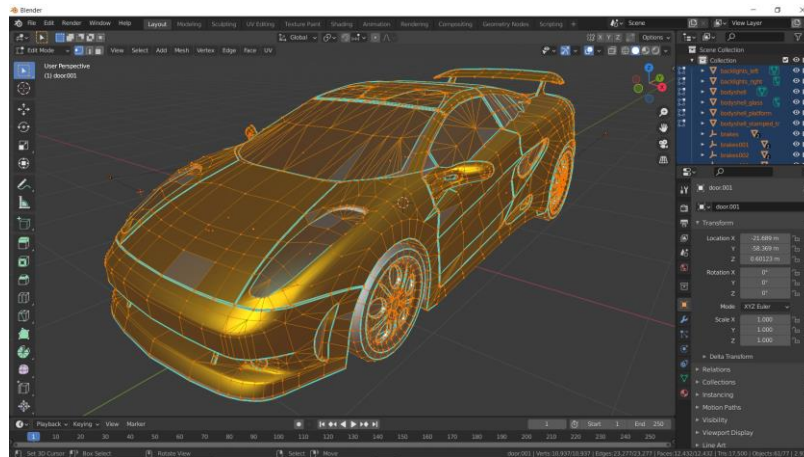
Gambar 3. 9 *Player Sampai Tujuan*

Pada gambar 3.9 setelah *player* mencapai tujuan akan terlihat hasil berupa *score* angka dan juga *score* bintang yang sudah di kalkulasi oleh algoritma A\* dan AHP.

### 3.3.1 Desain 3D

Desain 3D pada *game* ini digunakan sebagai visualisasi karakter dan kota tempat melakukan perjalanan. *Software* yang digunakan untuk mendesain 3D

adalah Blender3D versi 2.93.4 LTS. Berikut merupakan desain 3D yang digunakan pada penelitian ini.



Gambar 3. 10 Proses Pembuatan Karakter dengan *Software* Blender 3D

Gambar 3.10 adalah proses pembuat karakter yang akan digunakan dalam permainan ini.



Gambar 3. 11 Desain 3D Mobil

Pada gambar 3.11 terdapat *asset game* desain 3D mobil yang ada dalam permainan ini.



Gambar 3. 12 Desain 3D Motor

Pada gambar 3.12 terdapat *asset game* desain 3D motor yang ada dalam permainan ini.

Beberapa karakter 3D tersebut berfungsi sebagai *playable character* pada *game* penelitian ini. Nantinya pada saat *game* akan dimulai, *player* diharuskan untuk memilih kendaraan yang akan digunakan saat memainkan *game* ini.



Gambar 3. 13 Desain 3D Kota

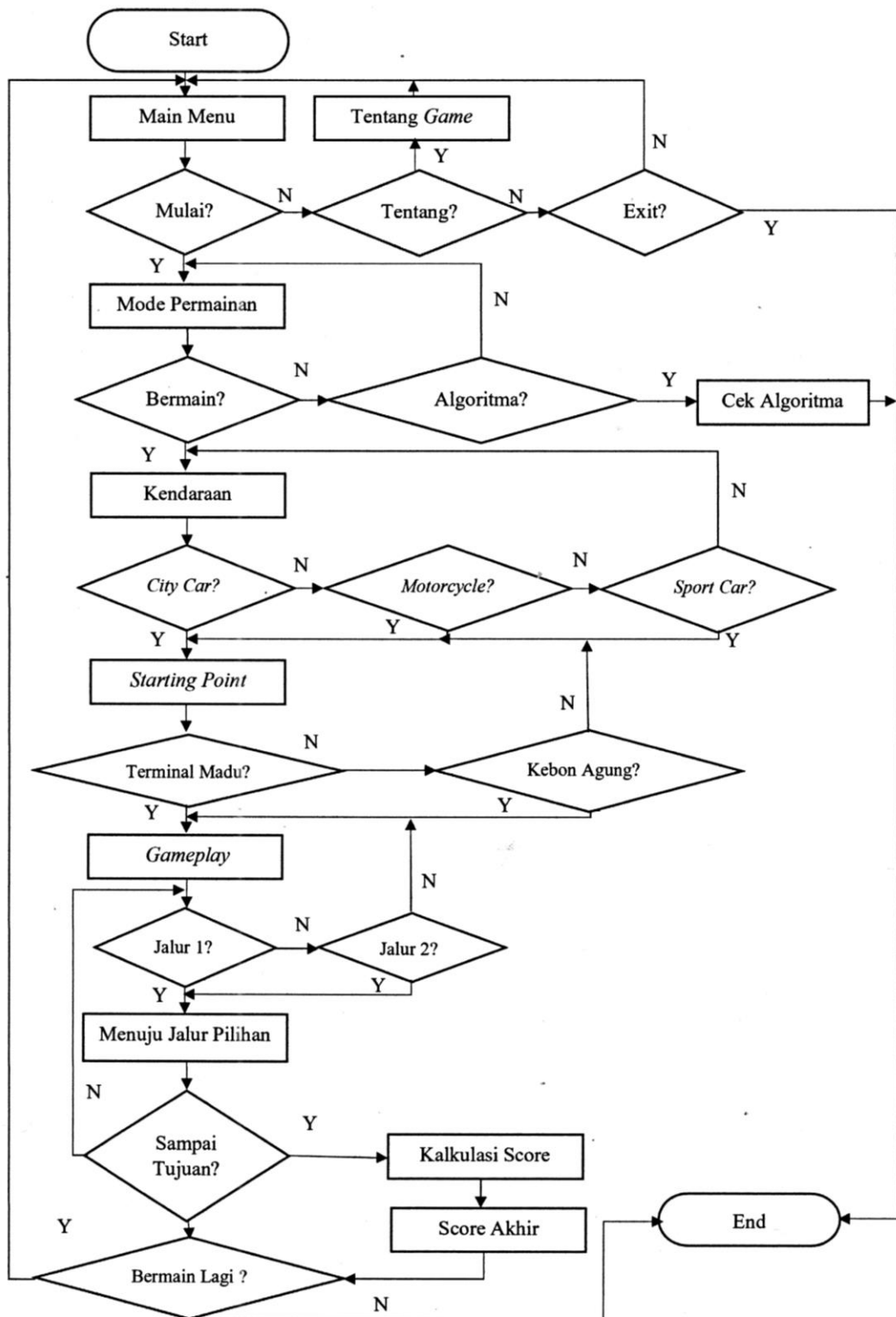
Pada gambar 3.13 terdapat desain kota yang berfungsi sebagai lintasan yang akan di lewati *player* saat memainkan *game* ini.

### 3.3.2 FSM (*Finish State Machine*)

Menurut Setiawan (2006) *Finite state machine* (FSM) merupakan sebuah metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan tiga hal berikut: *State* (keadaan), *event* (kejadian) dan *action* (aksi). Pada satu saat dalam periode waktu yang cukup signifikan, sistem akan berada pada salah satu state yang aktif. Sistem dapat beralih atau bertransisi menuju state lain jika mendapatkan masukan atau event tertentu, baik yang berasal dari perangkat luar atau komponen dalam sistemnya itu sendiri. Transisi keadaan ini umumnya juga disertai oleh aksi yang dilakukan oleh sistem ketika menanggapi masukan yang terjadi. Aksi yang dilakukan tersebut dapat berupa aksi yang sederhana atau melibatkan rangkaian proses yang relatif kompleks.

### 3.3 Rencana Skenario *Game*

Rencana skenario *game* merupakan susunan *scene* yang telah disusun sebagai urutan bagaimana *player* akan memainkan *game* ini. Skenario *game* pada penelitian ini dapat di lihat pada gambar 3.14 berikut.

Gambar 3. 14 *Flow Chart Skenario Game*

Gambar 3.14 merupakan rancangan skenario *game* yang akan di implementasikan pada penelitian ini. Untuk penjelasannya sebagai berikut.

1. Main menu, merupakan tampilan awal saat memulai *game* ini. Pada tampilan main menu terdapat tiga pilihan opsi, terdapat mulai, tentang, dan keluar, dengan fungsi sebagai berikut :
  - Mulai, untuk navigasi *player* agar dapat memulai permainan.
  - Tentang, akan muncul *pop up* yang menjelaskan tentang penjelasan dari *game* ini.
  - Keluar, sebagai navigasi *player* jika ingin keluar dari *game* ini.
2. Memilih mode permainan, dalam *game* ini terdapat dua mode permainan sebagai berikut.
  - Mode *game*, yaitu mode yang terdapat tantangan untuk *player* agar mendapatkan *score* sebaik mungkin dengan memilih kendaraan yang tepat berdasarkan kondisi yang sedang terjadi, dan juga harus memilih jalur terpendek agar bisa paling cepat menuju tujuan.
  - Mode penerapan algoritma, adalah mode penerapan algoritma A\* tidak terdapat tantangan pada mode ini, *player* bisa melakukan simulasi algoritma A\* dengan memilih jalur yang ingin dituju oleh karakter. Pada mode ini, *player* juga dapat menambahkan *obstacle* berupa jalur yang tidak dapat di lewati oleh karakter.
3. Memilih kendaraan, pada sesi ini terdapat beberapa pilihan kendaraan dan terdapat kondisi yang akan mempengaruhi hasil akhir dari permainan. Berikut

adalah beberapa pilihan yang kendaraan dan beberapa kondisi yang di *random* tiap permainan di mulai:

- Kondisi *random* meliputi kepadatan lalu lintas, cuaca, *budget* perjalanan.
  - *Selectable character* antara lain motor, mobil, mobil *sport*.
4. Memilih *starting point*, *player* dapat memilih tempat memulai permainan yaitu Terminal Madu dan Kebon Agung.
  5. Memilih jalur, setelah memilih *starting point* maka *player* akan masuk ke dalam *gameplay* utama. Pada *gameplay* tersebut *player* di haruskan untuk memilih jalur terpendek dibantu dengan *minimap*. Setelah itu *player* akan melakukan perjalanan pada jalur yang sudah dipilih.
  6. Saat *player* sudah mencapai tujuan yang di pilih *player*, terdapat dua kondisi, yaitu *player* sudah *finish* (menuju tujuan akhir), atau belum *finish* (masih belum menuju tujuan akhir). Jika *player* belum *finish* maka diharuskan untuk memilih jalur selanjutnya yang akan di tempuh. Namun jika *player* sudah *finish*, akan dilakukan proses kalkulasi hasil.
  7. Kalkulasi hasil *score*, berfungsi untuk menghitung *score* dari *player* menggunakan algoritma A\* dan AHP, untuk penjelasan perhitungan hasil *score* akan di jelakan pada subab selanjutnya.
  8. Setelah *score* selesai di kalkulasi akan muncul *pop up* tampilan *score* yang di dapatkan oleh *player*, berupa nilai angka dan tampilan bintang sebagai *rating score*. Setelah ini *player* dapat memainkan *game* ini lagi, dan juga dapat keluar dari *game* ini.

### 3.4 Perhitungan A\*

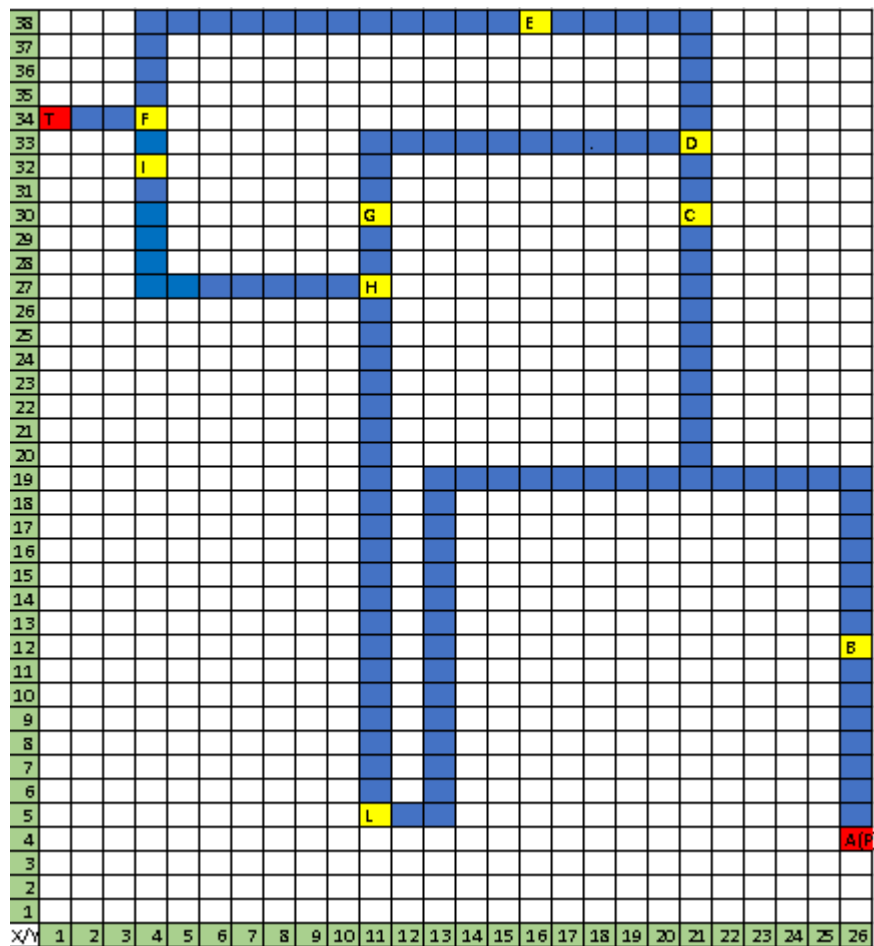
Langkah pertama dalam perhitungan algoritma A\* adalah mencari nilai heuristik dari n dengan menggunakan persamaan *euclidean distance* dibawah ini.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Setelah nilai heuristik dari n telah diketahui langkah kedua adalah menghitung nilai estimasi biaya terendah menggunakan persamaan dibawah ini.

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

Dibawah ini akan dihitung jarak terpendek antara candi Borobudur dari posisi *player*. Terdapat tiga jalur yang telah ditentukan untuk menuju candi Borobudur dari *starting point* yang dipilih oleh *player*, penentuan jalur diambil dari *google maps*.



Gambar 3. 15 *Indeks Poin Terminal Madu*

Pada gambar 3.15 setiap poin mewakili jarak 300m. *Player* berada di koordinat (26,4) dan tujuan dari *player* adalah koordinat (1,34). Berikut data jalur yang dipilih dari google *maps*:

Jalur 1:           A (Terminal Madu) – B (Krapyak) – C (Salam) – D (Lojirejo) – E  
 (Pertigaan Palbapang) – F (Pertigaan Balaputra) – T (Pintu Masuk  
 Candi Borobudur)

Jalur 2: A (Terminal Madu) – B(Krapyak) – C (Salam) – L (Dekso) – H (Nanggulan) – I (Wonorejo) – F (Pertigaan Balaputra) – T (Pintu Masuk Candi Borobudur)

Jalur 3: A (Terminal Madu) – B (Krapyak) – C (Salam) – D (Lojirejo) – G (Blongkeng) – H (Nanggulan) – I (Wonorejo) – F (Pertigaan Balaputra) – T (Pintu Masuk Candi Borobudur)

Berikut perhitungan jalur terpendek menggunakan algoritma A\* pada *starting point* terminal madu:

Tabel 3. 1 Perhitungan Algoritma A\* Jalur 1 Terminal Madu

	X1	X2	Y1	Y2	H(N)	G(N)	F(N)
A KE B	26	26	4	12	8	8	16
B KE C	26	21	12	30	19	23	41,682
C KE D	21	21	30	33	3	3	6
D KE E	21	16	33	38	7,1	10	17,071
E KE F	16	4	38	34	13	16	28,649
F KE T	4	1	34	34	3	3	6
TOTAL F(N)							115,4

Tabel 3. 2 Perhitungan Algoritma A\* Jalur 2 Terminal Madu

	X1	X2	Y1	Y2	H(N)	G(N)	F(N)
A KE B	26	26	4	12	8	8	16
B KE C	26	21	12	30	19	23	41,682
C KE L	21	11	30	5	27	35	61,926
L KE H	11	11	5	27	22	22	44
H KE I	11	4	27	32	8,6	12	20,602
I KE F	4	4	32	34	2	2	4
F KE T	4	1	34	34	3	3	6
TOTAL F(N)							194,21

Tabel 3. 3 Perhitungan Algoritma A\* Jalur 3 Terminal Madu

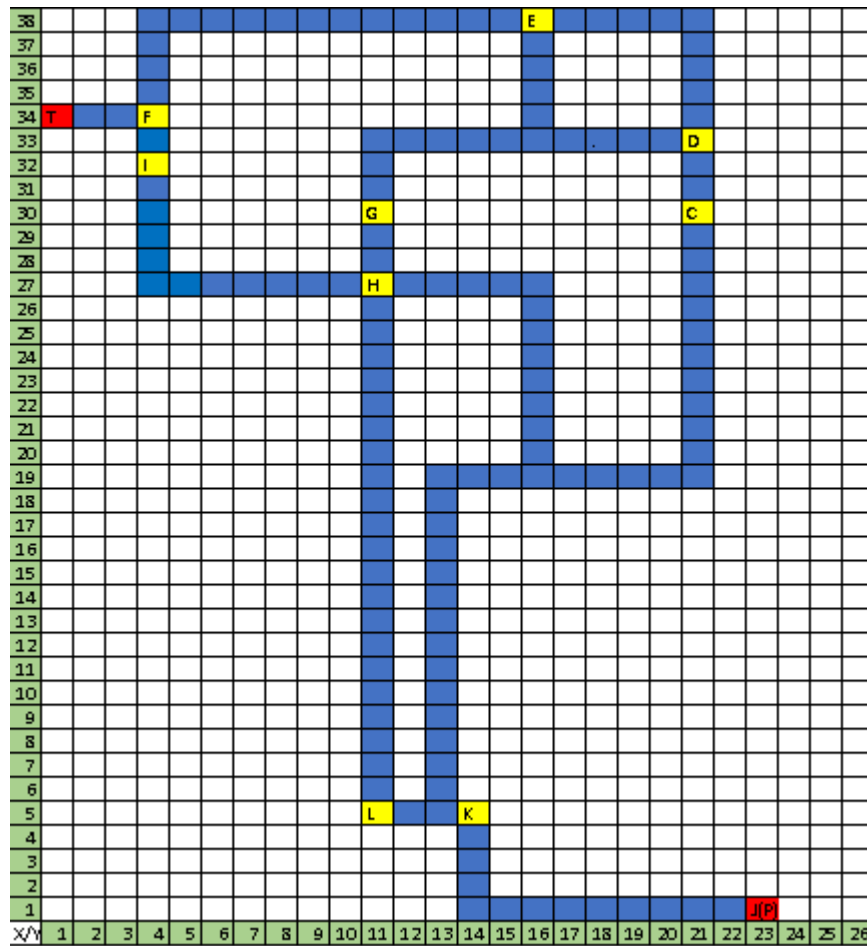
	X1	X2	Y1	Y2	H(N)	G(N)	F(N)
A KE B	26	26	4	12	8	8	16
B KE C	26	21	12	30	19	23	41,682
C KE D	21	21	30	33	3	3	6
D KE G	21	11	33	30	10	10	20,44
G KE H	11	11	30	27	3	16	19
H KE I	11	4	27	32	8,6	12	20,602
I KE F	4	4	32	34	2	2	4
F KE T	4	1	34	34	3	3	6
TOTAL F(N)							117,72

Setelah diketahui nilai F(N) pada setiap jalur , langkah berikutnya adalah menghitung jarak sebenarnya dengan melakukan perkalian antara F(N) dengan skala 300m, karena setiap titik pada indeks poin di atas mewakili 300m jarak sebenarnya.

$$\text{Jalur 1: } 115.4 \times 300\text{m} = 34.620\text{km}$$

$$\text{Jalur 2: } 194.21 \times 300\text{m} = 58,262\text{km}$$

$$\text{Jalur 3: } 117.72 \times 300\text{m} = 35.317\text{km}$$



Gambar 3. 16 *Indeks Poin Kebon Agung*

Pada gambar 3.16 setiap poin mewakili jarak 300m. *Player* berada di koordinat (23,1) dan tujuan dari *player* adalah koordinat (1,34). Berikut data jalur yang dipilih dari google *maps*:

Jalur 1:            J (Kebon Agung) – K (Plumbangan) – L (Dekso) – H (Nanggulan)  
                           – I (Wonorejo) – F (Pertigaan Balaputra) – T (Pintu Masuk Candi  
                           Borobudur)

Jalur 2: J (Kebon Agung) – K (Plumbangan) – L (Dekso) – C (Salam) – D (Lojirejo) – E (Pertigaan Palbapang) – F (Pertigaan Balaputra) – T (Pintu Masuk Candi Borobudur)

Jalur 3: J (Kebon Agung) – K (Plumbangan) – L (Dekso) – C (Salam) – D (Lojirejo) – G (Blongkeng) – H (Nanggulan) – I (Wonorejo) – F (Pertigaan Balaputra) – T (Pintu Masuk Candi Borobudur)

Berikut perhitungan jalur terpendek menggunakan algoritma A\* pada *starting point* kebon agung:

Tabel 3. 4 Perhitungan Algoritma A\* Jalur 1 Kebon Agung

	X1	X2	Y1	Y2	H(N)	G(N)	F(N)
J KE K	23	14	1	5	9,85	13	22,849
K KE L	14	11	5	5	3	3	6
L KE H	11	11	5	27	22	22	44
H KE I	11	4	27	32	8,6	12	20,602
I KE F	4	4	32	34	2	2	4
F KE T	4	1	34	34	3	3	6
TOTAL F(N)							103,45

Tabel 3. 5 Perhitungan Algoritma A\* Jalur 2 Kebon Agung

	X1	X2	Y1	Y2	H(N)	G(N)	F(N)
J KE K	23	14	1	5	9,85	13	22,849
K KE L	14	11	5	5	3	3	6
L KE C	11	21	5	30	26,9	22	48,926
C KE D	21	21	30	33	3	3	6
D KE E	21	16	33	38	7,07	10	17,071
E KE F	16	4	38	34	12,6	16	28,649
F KE T	4	1	34	34	3	3	6
TOTAL F(N)							112,65

Tabel 3. 6 Perhitungan Algoritma A\* Jalur 3 Kebon Agung

	X1	X2	Y1	Y2	H(N)	G(N)	F(N)
J KE K	23	14	1	5	9,85	13	22,849
K KE L	14	11	5	5	3	3	6
L KE C	11	21	5	30	26,9	22	48,926
C KE D	21	21	30	33	3	3	6
D KE G	21	11	33	30	10,4	10	20,44
G KE H	11	11	30	27	3	16	19
H KE I	11	4	27	32	8,6	12	20,602
I KE F	4	4	32	34	2	2	4
F KE T	4	1	34	34	3	3	6
TOTAL F(N)							124,97

Setelah diketahui nilai  $F(N)$  pada setiap jalur , langkah berikutnya adalah menghitung jarak sebenarnya dengan melakukan perkalian antara  $F(N)$  dengan skala 300m, karena setiap titik pada indeks poin di atas mewakili 300m jarak sebenarnya.

Jalur 1:  $115.4 \times 300\text{m} = 31.035\text{km}$

Jalur 2:  $194.21 \times 300\text{m} = 33,793\text{km}$

Jalur 3:  $117.72 \times 300\text{m} = 37.49\text{km}$

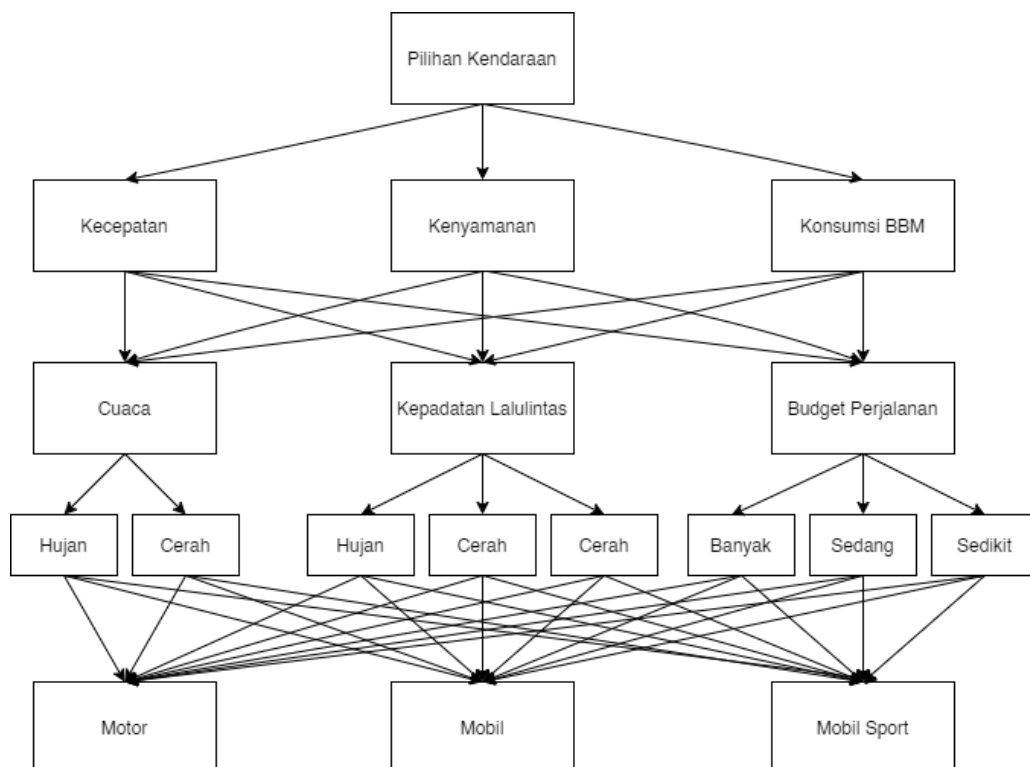
### 3.5 Penerapan AHP (Analytical Hierarchy Process)

Penerapan AHP pada penelitian ini berfungsi untuk melakukan spesifikasi pada hasil *score* yang akan keluar pada *game* ini. Terdapat beberapa input yang akan menentukan standarisasi *score* pada *game* ini antara lain:

1. Kenyamanan
2. Kecepatan
3. Kondisi cuaca

4. Kepadatan lalu lintas
5. *Budget perjalanan player*
6. Jenis kendaraan yang dipilih

Beberapa input tersebut akan menjadi variabel tambahan untuk menambah tantangan *player* dalam memainkan *game* ini. Dengan adanya metode AHP ini, *score* tertinggi tidak hanya bisa di dapatkan dengan menempuh jarak terdekat dalam *game*, namun *player* juga diharuskan memilih kendaraan yang tepat berdasarkan cuaca, kepadatan lalu lintas, dan *budget player* yang akan di berikan secara *random* pada saat *player* memulai permainan. Berikut adalah diagram blok proses AHP pada penelitian ini.



Gambar 3. 17 Diagram Blok Proses AHP

Gambar 3.17 menjelaskan alur sistem yang ada di AHP. Penelitian ini menggunakan dua kali proses AHP dan akan di golongkan menjadi variabel tetap dan variabel yang dapat berubah kondisi. Pemrosesan AHP dapat dilihat pada bab berikut.

### 3.5.1 AHP Variabel Tetap

Beberapa variabel yang digunakan pada AHP variabel tetap diantaranya kenyamanan, kecepatan, konsumsi BBM. Variabel tersebut digolongkan variabel tetap dikarenakan telah *build-in* dengan konfigurasi kendaraan. Oleh karena itu variabel ini tidak dapat berubah, dan hanya bisa berubah jika *player* mengganti jenis kendaraanya.

Langkah pertama dalam perancangan AHP adalah penyusunan matrik perbandingan berpasangan dan di dapatkan matrik berikut.

Tabel 3. 7 Matrik Perbandingan Berpasangan

	Kecepatan	Kenyamanan	Konsumsi BBM
Kecepatan	1,00	0,33	5,00
Kenyamanan	3,00	1,00	0,33
Konsumsi BBM	0,20	3,00	1,00

Langkah berikutnya adalah normalisasi matrik perbandingan, dengan menjumlahkan nilai tiap kolom matriks dan hasil akhirnya harus bernilai 1. Normalisasi matriks ditunjukkan dalam tabel 3.8 di bawah ini.

Tabel 3. 8 Normalisasi Matriks Keputusan

	Kecepatan	Kenyamanan	Kapasitas
Kecepatan	0,24	0,08	0,79
Kenyamanan	0,71	0,23	0,05
Konsumsi BBM	0,05	0,69	0,16
Jumlah kolom	1,00	1,00	1,00

Selanjutnya adalah melakukan analisis konsistensi, hal pertama yang harus dilakukan adalah memperoleh bobot dari setiap alternatif, diantaranya motor, mobil, mobil *sport*. Hasil dari analisis konsistensi setiap alternatif adalah sebagai berikut.

Berikut merupakan matrik perbandingan alternatif terhadap kriteria kecepatan.

Tabel 3.9 Matrik Perbandingan Alternatif Terhadap Kecepatan.

	Motor	Mobil	Mobil <i>Sport</i>
Motor	1,00	0,20	0,14
Mobil	3,00	1,00	0,33
Mobil <i>Sport</i>	5,00	3,00	1,00

Tabel 3.10 Normalisasi Matrik Perbandingan Alternatif Terhadap Kecepatan.

	Motor	Mobil	Mobil <i>Sport</i>
Motor	1,00	0,20	0,14
Mobil	3,00	1,00	0,33
Mobil <i>Sport</i>	5,00	3,00	1,00
Jumlah kolom	1,00	1,00	1,00

Berikut merupakan matrik perbandingan alternatif terhadap kriteria kenyamanan.

Tabel 3.11 Matrik Perbandingan Alternatif Terhadap Kenyamanan.

	Motor	Mobil	Mobil <i>Sport</i>
Motor	1,00	0,20	0,33
Mobil	5,00	1,00	3,00
Mobil <i>Sport</i>	3,00	0,33	1,00

Tabel 3. 12 Normalisasi Matrik Perbandingan Alternatif Terhadap Kenyamanan.

	Motor	Mobil	Mobil <i>Sport</i>
Motor	0,11	0,13	0,08
Mobil	0,56	0,65	0,69
Mobil <i>Sport</i>	0,33	0,22	0,23
Jumlah kolom	1,00	1,00	1,00

Berikut merupakan matrik perbandingan alternatif terhadap kriteria konsumsi BBM.

Tabel 3. 13 Matrik Perbandingan Alternatif Terhadap Konsumsi BBM.

	Motor	Mobil	Mobil <i>Sport</i>
Motor	1,00	3,00	9,00
Mobil	0,33	1,00	3,00
Mobil <i>Sport</i>	0,11	0,33	1,00

Tabel 3. 14 Normalisasi Matrik Perbandingan Alternatif Terhadap Konsumsi BBM

	Motor	Mobil	Mobil <i>Sport</i>
Motor	0,69	0,69	0,69
Mobil	0,23	0,23	0,23
Mobil <i>Sport</i>	0,08	0,08	0,08
Jumlah kolom	1,00	1,00	1,00

Selanjutnya mencari vektor *eigen* dari matrik dengan menghitung rata-rata normalisasi.

$$C = \begin{bmatrix} 0.09 & 0.11 & 0.69 \\ 0.27 & 0.63 & 0.23 \\ 0.65 & 0.26 & 0.08 \end{bmatrix}$$

Langkah selanjutnya, mengontrol nilai bobot konsistensi.

$$A \times C = \begin{bmatrix} 0.09 & 0.11 & 0.69 \\ 0.27 & 0.63 & 0.23 \\ 0.65 & 0.26 & 0.08 \end{bmatrix} \times \begin{bmatrix} 0.37 \\ 0.33 \\ 0.30 \end{bmatrix}$$

$$A \times C = \begin{bmatrix} 0.275 \\ 0.38 \\ 0.35 \end{bmatrix}$$

Pada perhitungan tersebut didapatkan hasil bobot akhir berupa motor dengan bobot 0,275, mobil dengan bobot 0,38, dan mobil *sport* dengan bobot 0,35.

### 3.5.2 AHP Variabel Tidak Tetap

Selanjutnya adalah perhitungan AHP variabel tidak tetap, perhitungan ini dilakukan dengan tujuan agar standarisasi *scoring* tidak hanya terbatas dengan satu sampai tiga standard saja, namun terdapat banyak standard berbeda bergantung pada kondisi awal *game* dimainkan. Beberapa variabel tidak tetap yang digunakan adalah sebagai berikut.

1. Cuaca, terbagi menjadi cuaca cerah dan hujan.
2. Kepadatan lalu lintas, terbagi menjadi ramai, sedang, dan sepi.
3. *Budget* perjalanan, terbagi menjadi banyak, sedang, dan sedikit.

Dengan beberapa opsi perbedaan kondisi tersebut, maka dapat dimungkinkan terjadi 18 skenario permainan yang dapat dimainkan *player*. Beberapa skenario yang dapat dimainkan *player* antara lain sebagai berikut.

Tabel 3. 15 List Kondisi Awal Permainan

No	Kondisi	Cuaca	Kepadatan Lalu lintas	Budget Perjalanan
1	Kondisi 1	Hujan	Ramai	Banyak
2	Kondisi 2	Hujan	Ramai	Sedang
3	Kondisi 3	Hujan	Ramai	Sedikit
4	Kondisi 4	Hujan	Sedang	Banyak
5	Kondisi 5	Hujan	Sedang	Sedang
6	Kondisi 6	Hujan	Sedang	Sedikit
7	Kondisi 7	Hujan	Sepi	Banyak
8	Kondisi 8	Hujan	Sepi	Sedang
9	Kondisi 9	Hujan	Sepi	Sedikit
10	Kondisi 10	Cerah	Ramai	Banyak
11	Kondisi 11	Cerah	Ramai	Sedang
12	Kondisi 12	Cerah	Ramai	Sedikit
13	Kondisi 13	Cerah	Sedang	Banyak
14	Kondisi 14	Cerah	Sedang	Sedang
15	Kondisi 15	Cerah	Sedang	Sedikit
16	Kondisi 16	Cerah	Sepi	Banyak

17	Kondisi 17	Cerah	Sepi	Sedang
18	Kondisi 18	Cerah	Sepi	Sedikit

Setiap kondisi di atas, akan di hitung nilai AHP dengan rumus dan langkah yang sama seperti perhitungan AHP dengan variabel tetap. Hasil tadi perhitungan AHP beberapa kondisi di atas mendapatkan beberapa bobot akhir sebagai berikut,

Tabel 3. 16 Tabel Bobot Akhir AHP

No	Kondisi	Motor	Mobil	Mobil Sport
1	Kondisi 1	0,76297186	1,10204082	1,13498733
2	Kondisi 2	0,90007087	1,04996456	1,04996456
3	Kondisi 3	1,20508289	0,93410755	0,86080956
4	Kondisi 4	0,76297186	1,10204082	1,13498733
5	Kondisi 5	0,90007087	1,04996456	1,04996456
6	Kondisi 6	1,20508289	0,93410755	0,86080956
7	Kondisi 7	0,61667415	1,18290675	1,2004191
8	Kondisi 8	0,64676617	1,17091016	1,18232368
9	Kondisi 9	0,80994489	1,10585672	1,08419839
10	Kondisi 10	0,73801743	1,11151961	1,15046296
11	Kondisi 11	1,13798791	0,87978743	0,98222467
12	Kondisi 12	1,37385621	0,8130719	0,8130719
13	Kondisi 13	1,12628938	0,88309637	0,99061426
14	Kondisi 14	0,90007087	1,04996456	1,04996456
15	Kondisi 15	1,20508289	0,93410755	0,86080956
16	Kondisi 16	0,55215686	1,22086275	1,22698039
17	Kondisi 17	0,76297186	1,13498733	1,10204082
18	Kondisi 18	1,13798791	0,98222467	0,87978743

### 3.6 Proses Scoring

Setelah perhitungan A\* dan AHP dilakukan, proses selanjutnya adalah proses perhitungan *score*. Setiap *player* menyelesaikan permainan, maka *score* dari permainan akan dihitung menggunakan kombinasi A\* dan AHP, *score player* nantinya akan dibandingkan dengan kemungkinan *score* maksimal dan akan didapatkan lima rating penilaian, diantaranya adalah sebagai berikut.

1. Bintang 1 untuk *score* sebesar 0 – 20% dari *score* maksimal

2. Bintang 2 untuk *score* sebesar 21 – 40% dari *score* maksimal
3. Bintang 3 untuk *score* sebesar 41 – 60% dari *score* maksimal
4. Bintang 4 untuk *score* sebesar 61 – 80% dari *score* maksimal
5. Bintang 5 untuk *score* sebesar 81 – 100% dari *score* maksimal

Berdasarkan lima rating di atas, maka *output* dari *game* ini ada lima jenis *output* berdasarkan *score* yang di dapatkan *player* saat memainkan *game* ini. *Score* *player* tersebut di dapatkan dari beberapa variabel, diantaranya sebagai berikut.

1. Jarak yang di tempuh *player* selama permainan.
2. Jarak terdekat berdasarkan pencarian algoritma A\*.
3. Bobot akhir AHP variabel tetap terbaik.
4. Bobot akhir AHP variabel tidak tetap terbaik.
5. Bobot akhir AHP variabel tetap yang dipilih oleh *player*.
6. Bobot akhir AHP variabel tidak tetap yang dipilih oleh *player*.

Setelah data-data di atas didapatkan, dilanjutkan dengan proses penghitungan *score* *player* dengan rumus sebagai berikut.

$$\frac{\left(200 - \left(\frac{Ja}{Jb} \times 100\right)\right) + \left(\frac{(Ka1 + Ka2)}{Kb1 + Kb2} \times 100\right)}{2}$$

*Ja* : Jarak yang di tempuh *player* selama permainan.

*Jb* : Jarak terdekat berdasarkan pencarian logika A\*.

*Kb1* : Bobot akhir AHP variabel tetap terbaik.

*Kb2* : Bobot akhir AHP variabel tidak tetap terbaik.

$Ka1$  : Bobot akhir AHP variabel tetap yang dipilih oleh *player*.

$Ka2$  : Bobot akhir AHP variabel tidak tetap yang dipilih oleh *player*.

Sebagai contoh perhitungan akan disimulasikan *player* bermain pada kondisi 1, menempuh jarak perjalanan 80Km, dengan menggunakan kendaraan motor. Maka perhitungan *score player* adalah sebagai berikut.

$$Score = \frac{\left(200 - \left(\frac{80}{66} \times 100\right)\right) + \left(\frac{(0.27 + 0.77)}{0.37 + 1.12} \times 100\right)}{2}$$

$$Score = \frac{(200 - 121) + \left(\frac{1.04}{1.12} \times 100\right)}{2}$$

$$Score = \frac{79 + 71}{2}$$

$$Score = 75\%$$

Dengan hasil perhitungan di atas, hasil yang di dapatkan adalah score 75%, maka *player* akan mendapatkan *score* bintang 4.

### 3.6 Rencana Pengujian

Dalam menentukan pengujian yang akan dilakukan memiliki hasil yang baik, beberapa faktor terkait akan digunakan untuk melihat efektifnya metode yang diterapkan pada penelitian ini. Dalam pengujian ini menggunakan efektifitas jarak terpendek dalam perjalanan menuju objek wisata. Proses pengujian ini dilakukan untuk mengetahui estimasi jarak yang dibutuhkan untuk perjalanan menuju Candi Borobudur dari *starting point* yang dipilih. Dalam hal ini, jarak yang ditempuh akan

diketahui ketika pengunjung berhasil sampai di tempat tujuan dengan pemilihan jalur yang telah diberikan. Semakin pendek jarak yang dilalui dalam perjalanan menuju objek wisata, maka semakin berhasil pengujian tersebut.

Untuk mengetahui akurasi yang di hasilkan berdasarkan error yang didapatkan pada *game* simulasi ini menggunakan rumus galat *error* sebagai berikut.

$$error = abs \left( \frac{Exact Value - Approximate Value}{Exact Value} \right) \times 100$$

*Exact value* adalah waktu sebenarnya yang di dapatkan dari google *maps*. Sedangkan *approximate value* adalah nilai dugaan atau prediksi yang di hasilkan dari *game* simulasi ini.

## BAB IV

### IMPLEMENTASI DAN PENGUJIAN

Bab ini menjelaskan mengenai implementasi dan pengujian terhadap sistem yang telah dibuat. Tujuan dari implementasi dan pengujian ini untuk membuktikan keakurasian metode yang digunakan peneliti dalam *game* simulasi.

#### 4.1 Implementasi

Tujuan dari dilakukannya implementasi adalah untuk menguji penerapan langkah-langkah yang ada pada bab sebelumnya.

##### 4.1.1 Perangkat Keras yang Digunakan

Perangkat keras yang digunakan dalam implementasi dan pengujian pada penelitian ini adalah

Tipe	Laptop
Merek	Asus X441U
CPU	Intel Core i3-6100U
GPU	Nvidia MX930
RAM	4GB DDR4

##### 4.1.2 Perangkat Lunak yang Digunakan

Dalam penelitian *game* simulasi pemilihan jalur terpendek pada *game* candi Borobudur menggunakan algoritma A\* digunakan *hardware* sebagai berikut.

<i>Operating System</i>	Windows 10 Home
<i>Game Engine</i>	Unity3D 2019 v4.23f1
<i>3D Modeller</i>	Blender 3D v2.93.4
<i>UI Editor</i>	Photoshop 2021
<i>Script Editor</i>	Visual Studio 2019

### 4.1.3 Implementasi Skenario *Game*

Implementasi skenario *game* adalah visualisasi dari aplikasi yang telah dibangun menggunakan metode yang telah disebutkan di atas. Berikut urutan skenario bagaimana *game* simulasi pemilihan jalur terpendek pada *game* candi Borobudur menggunakan algoritma A\*.

#### 1. *Main Menu*



Gambar 4. 1 *Main Menu*

*Main menu*, adalah tampilan awal saat memulai *game* ini. Pada tampilan main menu terdapat tiga pilihan opsi, terdapat mulai, tentang, dan keluar, dengan fungsi sebagai berikut:

- Mulai, untuk navigasi *player* agar dapat memulai permainan.
- Tentang, akan muncul *pop up* yang menjelaskan tentang penjelasan dari *game* ini.
- Keluar, sebagai navigasi *player* jika ingin keluar dari *game* ini.

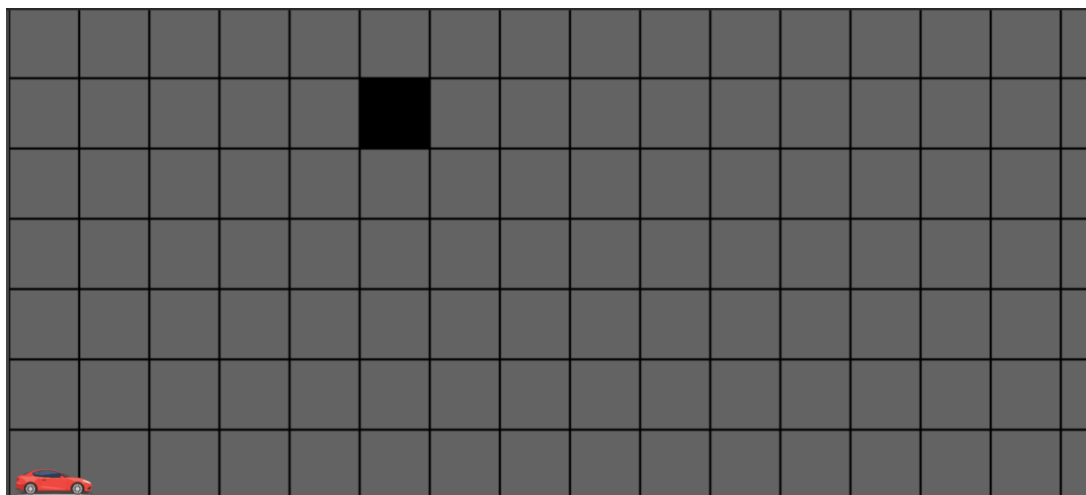
## 2. Pemilihan Mode



Gambar 4. 2 Pemilihan Mode

Memilih mode permainan, dalam *game* ini terdapat dua mode permainan sebagai berikut.

- Mode *game*, yaitu mode yang terdapat tantangan untuk *player* agar mendapatkan *score* sebaik mungkin dengan memilih kendaraan yang tepat berdasarkan kondisi yang sedang terjadi, dan juga harus memilih jalur terpendek agar bisa paling cepat menuju tujuan.



Gambar 4. 3 Mode Penerapan Algoritma A\*

- Mode penerapan algoritma A\*, adalah mode penerapan algoritma A\* tidak terdapat tantangan pada mode ini, *player* bisa melakukan simulasi algoritma A\* dengan memilih jalur yang ingin dituju oleh karakter. Pada mode ini, *player* juga dapat menambahkan *obstacle* berupa jalur yang tidak dapat di lewati oleh karakter.

### 3. Pemilihan Kendaraan



Gambar 4. 4 Pemilihan Kendaraan

Memilih kendaraan, pada sesi ini terdapat beberapa pilihan kendaraan dan terdapat kondisi yang akan mempengaruhi hasil akhir dari permainan. Berikut adalah beberapa pilihan yang kendaraan dan beberapa kondisi yang *random* tiap permainan di mulai:

- Kondisi random meliputi kepadatan lalu lintas, cuaca, *budget* perjalanan.
- *Selectable character* antara lain motor, mobil, mobil *sport*.

#### 4. Pemilihan *Starting Point*



Gambar 4. 5 Pemilihan *Starting Point*

Memilih *starting point*, *player* dapat memilih tempat memulai permainan, pilihan tersebut diantaranya adalah Terminal Madu dan Kebon Agung.

#### 5. Doa Sebelum Perjalanan



Gambar 4. 6 Doa Sebelum Perjalanan

Tampilan ini akan muncul setelah *player* menentukan kendaraan yang dipilih. Beberapa doa akan muncul saat permainan.

## 6. Pemilihan Rute



Gambar 4. 7 Pemilihan Rute

Memilih jalur, setelah memilih *starting point* maka *player* akan masuk ke dalam *gameplay* utama. Pada *gameplay* tersebut *player* di haruskan untuk memilih jalur terpendek dibantu dengan *minimap*. Setelah itu *player* akan melakukan perjalanan pada jalur yang sudah dipilih.



Gambar 4. 8 Animasi *Traveling*

Saat sudah mencapai tujuan yang di pilih *player*, terdapat dua kondisi, yaitu *player* sudah *finish* (menuju tujuan akhir), atau belum *finish* (masih belum menuju tujuan akhir).



Gambar 4. 9 *Player* Mencapai Tujuan Akhir

Jika *player* belum *finish* maka diharuskan untuk memilih jalur selanjutnya yang akan di tempuh. Namun jika *player* sudah *finish*, akan dilakukan proses kalkulasi hasil. Kalkulasi hasil *score*, berfungsi untuk menghitung *score* dari *player* menggunakan algoritma A\* dan AHP. Jarak ditempuh adalah jarak yang ditempuh *player* selama proses permainan berlangsung. Jarak terbaik adalah jarak terpendek berdasarkan algoritma A\*. Setelah *score* selesai di kalkulasi akan muncul *pop up* tampilan *score* yang di dapatkan oleh *player*, berupa nilai angka dan tampilan bintang sebagai *rating score*. Setelah ini *player* dapat memainkan *game* ini lagi, dan juga dapat keluar dari *game* ini.

#### 4.2 Pengujian A\* Pada *Game*

Pengujian ini bertujuan untuk mengetahui apakah algoritma A\* pada penelitian ini dapat mencapai target yang diinginkan. Pengujian ini dilakukan dengan melakukan simulasi perjalanan dari beberapa *starting point* menuju tujuan akhir yaitu candi Borobudur. Untuk mengetahui jumlah akurasi yang di hasilkan

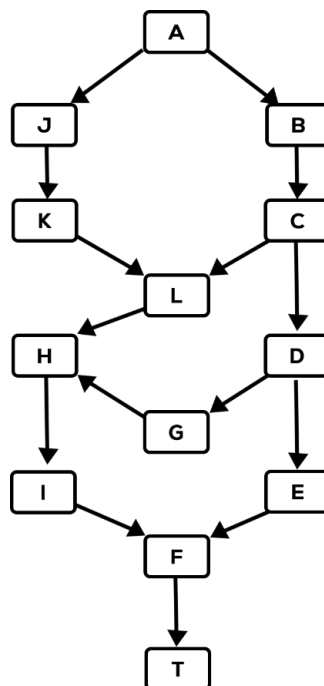
berdasarkan *error* yang terjadi pada pengujian, peneliti menggunakan rumus galat *error* sebagai berikut.

$$\%error = abs \left( \frac{Exact\ Value - Approximate\ Value}{Exact\ Value} \right) \times 100$$

*Exact value* merupakan nilai *real* dimana data jarak yang didapatkan dari *google maps* secara langsung, sedangkan *approximate value* merupakan nilai prediksi atau dugaan dimana hasil jarak yang di tempuh berdasarkan pengujian simulasi pada *game* ini.

#### 1. *Starting Point* Terminal Madu

Pada *starting point* terminal Madu, terdapat beberapa jalur yang dapat di lalui *player*. Berikut adalah visualisasi jalur yang dapat di lalui *player*.



Gambar 4. 10 Visualisasi Jalur Rute dari *Starting Point* Terminal Madu

A : Terminal Madu

H : Nanggulan

B : Krapyak

I : Wonorejo

C : Salam

J : Kebon Agung

D : Lojirejo

K : Plumbangan

E : Pertigaan Palbapang

L : Dekso

F : Pertigaan Balaputra

T : Candi Borobudur

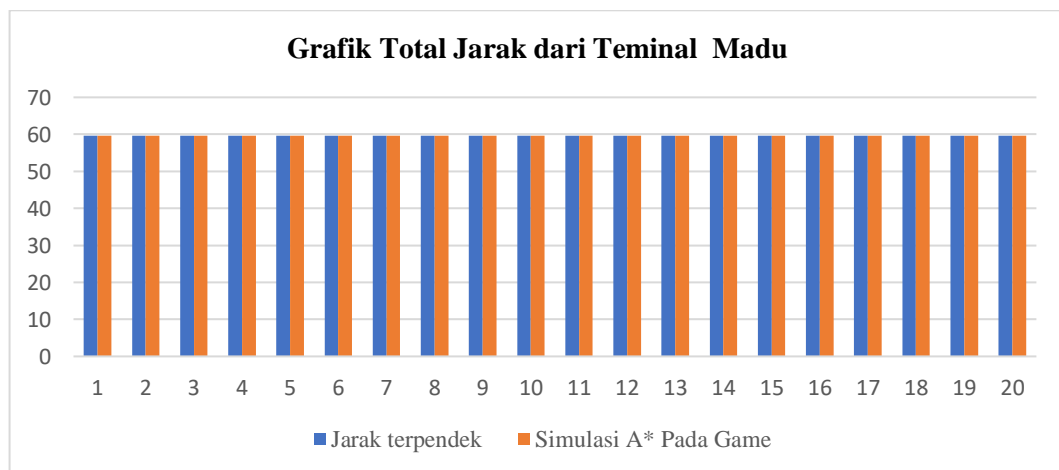
G : Blongkeng

Berikut adalah tabel pengujian pencarian jalur terpendek menggunakan algoritma A\* pada *game* simulasi wisata candi Borobudur melalui *starting point* terminal Madu.

Tabel 4. 1 Tabel Uji Pencarian Jalur dari Terminal Madu

Uji ke	<i>Starting Point</i>	Rute	Jarak Total	Error
1	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
2	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
3	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
4	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
5	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
6	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
7	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
8	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
9	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
10	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
11	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.

12	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
13	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
14	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
15	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
16	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
17	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
18	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
19	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
20	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.

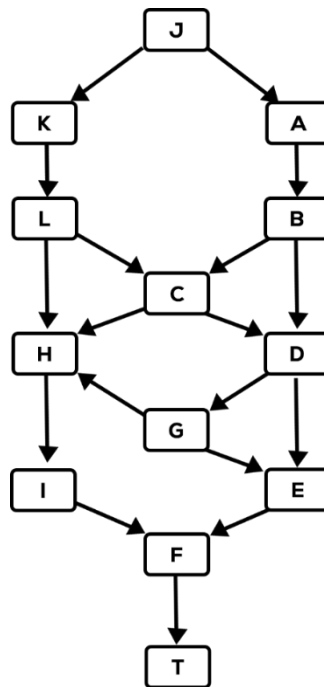


Gambar 4. 11 Grafik Total Jarak dari Teminal Madu

Berdasarkan pengujian yang telah dilaksanakan sebanyak 20 kali, dihasilkan rata-rata persentase *error* sebesar 0%, dan nilai akurasi yang di dapatkan dari rata-rata *error* adalah 100%. Hal ini menunjukkan bahwa algoritma A\* berjalan dengan sangat baik di *game* simulasi ini.

## 2. *Starting Point* Kebon Agung

Pada *starting point* Terminal Madu, terdapat beberapa jalur yang dapat di lalui *player*. Berikut adalah visualisasi jalur yang dapat di lalui *player*.



Gambar 4. 12 Visualisasi Jalur Rute dari *Starting Point* Kebon Agung

A : Terminal Madu

H : Nanggulan

B : Krpyak

I : Wonorejo

C : Salam

J : Kebon Agung

D : Lojirejo

K : Plumbangan

E : Pertigaan Palbapang

L : Dekso

F : Pertigaan Balaputra

T : Candi Borobudur

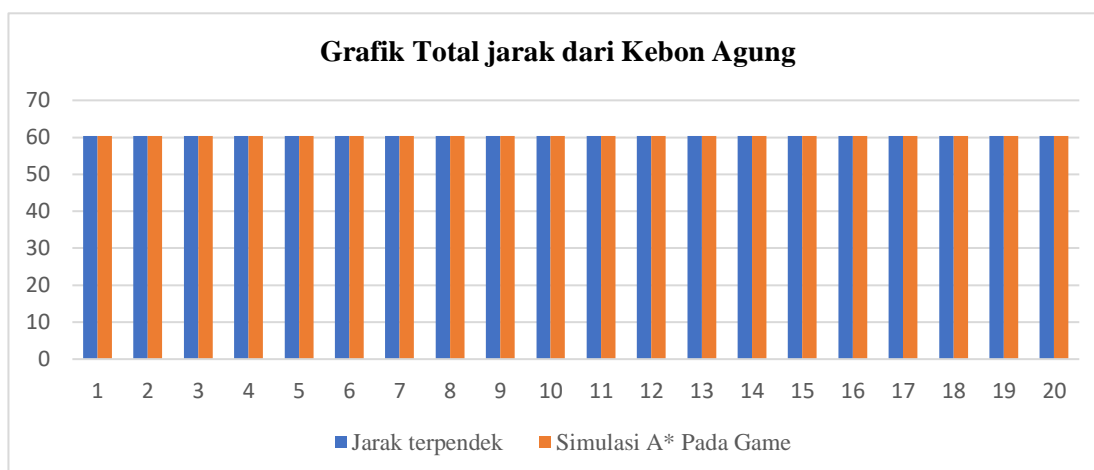
G : Blongkeng

Berikut adalah tabel pengujian pencarian jalur terpendek menggunakan algoritma A\* pada *game* simulasi wisata candi Borobudur melalui *starting point* Kebon Agung.

Tabel 4. 2 Tabel Uji Pencarian Jalur dari Kebon Agung

Uji ke	<i>Starting Point</i>	Rute	Jarak Total	Error
1	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
2	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
3	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
4	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
5	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
6	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
7	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
8	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
9	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
10	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
11	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
12	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
13	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
14	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
15	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
16	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
17	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
18	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.

19	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.
20	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	Tidak, karena <i>player</i> dapat menemukan tujuan akhir.



Gambar 4. 13 Grafik Total jarak dari Kebon Agung

Berdasarkan pengujian yang telah dilakukan sebanyak 20 kali, memiliki hasil yang sama dengan *starting point* sebelumnya, yaitu untuk rata-rata persentase *error* sebesar 0%, dan nilai akurasi yang di dapatkan dari rata-rata *error* adalah 100%. Hal ini menunjukkan bahwa algoritma A\* berjalan dengan sangat baik di *game* simulasi ini.

### 4.3 Pengujian AHP Pada *Game*

Pengujian ini dilakukan dengan tujuan untuk mengetahui apakah metode AHP pada *game* simulasi pemilihan jalur terpendek pada *game* candi Borobudur menggunakan algoritma A\* dapat berjalan sesuai fungsinya. Pengujian AHP dilakukan dengan melakukan percobaan perjalanan beberapa kali dengan skenario yang sudah ada, pengujian dinyatakan sukses jika *value score* mendekati dengan skenario. Untuk mengetahui jumlah akurasi *score*, perhitungan dilakukan dengan

menghitung *error* yang terjadi pada pengujian, peneliti menggunakan rumus galat *error* sebagai berikut.

$$\%error = abs \left( \frac{Exact Value - Approximate Value}{Exact Value} \right) \times 100$$

*Exact value* merupakan nilai yang berada diskenario yang sebelumnya sudah di hitung menggunakan *Microsoft Excel*, sedangkan *approximate value* merupakan nilai *score* berdasarkan pengujian simulasi pada *game* ini.

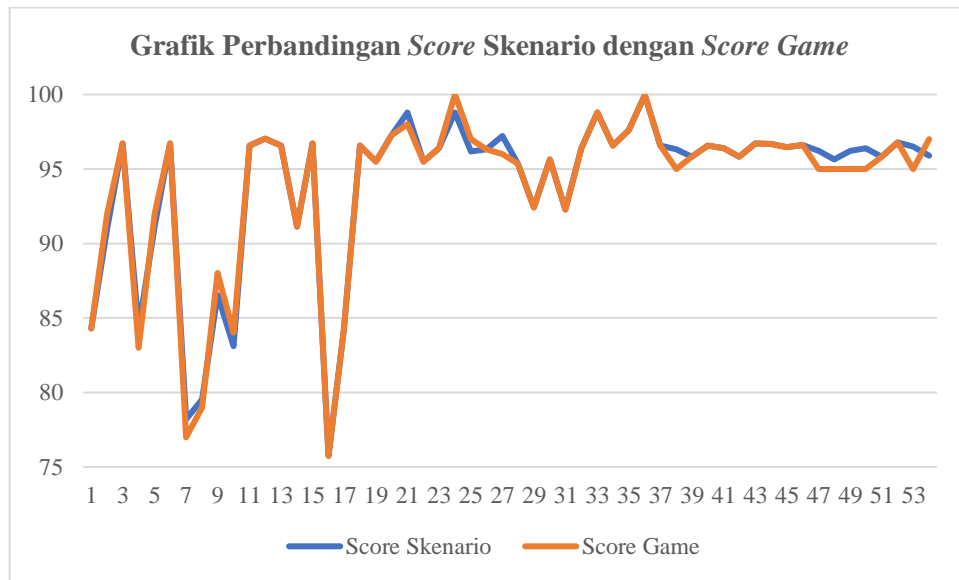
Tabel 4. 3 Tabel Uji Metode AHP

Uji ke	Kondisi	Cuaca	Kepadatan Lalu lintas	<i>Budget</i> Perjalanan	Kendaraan	<i>Starting Point</i>	Jalur	Jarak	<i>Score</i> Skenario (%)	<i>Score</i> Game (%)
1	Kondisi 1	Hujan	Ramai	Banyak	Motor	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	84	84
2	Kondisi 2	Hujan	Ramai	Sedang	Motor	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	91	92
3	Kondisi 3	Hujan	Ramai	Sedikit	Motor	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	97	97
4	Kondisi 4	Hujan	Sedang	Banyak	Motor	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	84	83
5	Kondisi 5	Hujan	Sedang	Sedang	Motor	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	91	92
6	Kondisi 6	Hujan	Sedang	Sedikit	Motor	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	97	97
7	Kondisi 7	Hujan	Sepi	Banyak	Motor	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	78	77
8	Kondisi 8	Hujan	Sepi	Sedang	Motor	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	80	79
9	Kondisi 9	Hujan	Sepi	Sedikit	Motor	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	87	88
10	Kondisi 10	Cerah	Ramai	Banyak	Motor	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	83	84
11	Kondisi 11	Cerah	Ramai	Sedang	Motor	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	97	97
12	Kondisi 12	Cerah	Ramai	Sedikit	Motor	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	97	97
13	Kondisi 13	Cerah	Sedang	Banyak	Motor	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	97	97
14	Kondisi 14	Cerah	Sedang	Sedang	Motor	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	91	91
15	Kondisi 15	Cerah	Sedang	Sedikit	Motor	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	97	97
16	Kondisi 16	Cerah	Sepi	Banyak	Motor	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	76	76

17	Kondisi 17	Cerah	Sepi	Sedang	Motor	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	84	84
18	Kondisi 18	Cerah	Sepi	Sedikit	Motor	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	97	97
19	Kondisi 1	Hujan	Ramai	Banyak	Mobil	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	95	95
20	Kondisi 2	Hujan	Ramai	Sedang	Mobil	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	97	97
21	Kondisi 3	Hujan	Ramai	Sedikit	Mobil	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	99	98
22	Kondisi 4	Hujan	Sedang	Banyak	Mobil	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	95	95
23	Kondisi 5	Hujan	Sedang	Sedang	Mobil	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	96	96
24	Kondisi 6	Hujan	Sedang	Sedikit	Mobil	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	99	100
25	Kondisi 7	Hujan	Sepi	Banyak	Mobil	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	96	97
26	Kondisi 8	Hujan	Sepi	Sedang	Mobil	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	96	96
27	Kondisi 9	Hujan	Sepi	Sedikit	Mobil	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	97	96
28	Kondisi 10	Cerah	Ramai	Banyak	Mobil	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	95	95
29	Kondisi 11	Cerah	Ramai	Sedang	Mobil	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	92	92
30	Kondisi 12	Cerah	Ramai	Sedikit	Mobil	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	96	96
31	Kondisi 13	Cerah	Sedang	Banyak	Mobil	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	92	92
32	Kondisi 14	Cerah	Sedang	Sedang	Mobil	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	96	96
33	Kondisi 15	Cerah	Sedang	Sedikit	Mobil	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	99	99
34	Kondisi 16	Cerah	Sepi	Banyak	Mobil	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	97	97

35	Kondisi 17	Cerah	Sepi	Sedang	Mobil	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	98	98
36	Kondisi 18	Cerah	Sepi	Sedikit	Mobil	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	100	100
37	Kondisi 1	Hujan	Ramai	Banyak	Mobil <i>sport</i>	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	97	97
38	Kondisi 2	Hujan	Ramai	Sedang	Mobil <i>sport</i>	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	96	95
39	Kondisi 3	Hujan	Ramai	Sedikit	Mobil <i>sport</i>	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	96	96
40	Kondisi 4	Hujan	Sedang	Banyak	Mobil <i>sport</i>	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	97	97
41	Kondisi 5	Hujan	Sedang	Sedang	Mobil <i>sport</i>	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	96	96
42	Kondisi 6	Hujan	Sedang	Sedikit	Mobil <i>sport</i>	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	96	96
43	Kondisi 7	Hujan	Sepi	Banyak	Mobil <i>sport</i>	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	97	97
44	Kondisi 8	Hujan	Sepi	Sedang	Mobil <i>sport</i>	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	97	97
45	Kondisi 9	Hujan	Sepi	Sedikit	Mobil <i>sport</i>	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	96	96
46	Kondisi 10	Cerah	Ramai	Banyak	Mobil <i>sport</i>	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	97	97
47	Kondisi 11	Cerah	Ramai	Sedang	Mobil <i>sport</i>	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	96	95
48	Kondisi 12	Cerah	Ramai	Sedikit	Mobil <i>sport</i>	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	96	95
49	Kondisi 13	Cerah	Sedang	Banyak	Mobil <i>sport</i>	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	96	95
50	Kondisi 14	Cerah	Sedang	Sedang	Mobil <i>sport</i>	Terminal Madu	A,B,C,D,E,F,T	34,620 Km	96	95
51	Kondisi 15	Cerah	Sedang	Sedikit	Mobil <i>sport</i>	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	96	96
52	Kondisi 16	Cerah	Sepi	Banyak	Mobil <i>sport</i>	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	97	97

53	Kondisi 17	Cerah	Sepi	Sedang	Mobil <i>sport</i>	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	97	95
54	Kondisi 18	Cerah	Sepi	Sedikit	Mobil <i>sport</i>	Kebon Agung	J,K,L,H,I,F,T	31.035 Km	96	97



Gambar 4. 14 Grafik Perbandingan *Score Skenario* dengan *Score Game*

Berdasarkan pengujian yang dilakukan sebanyak 54 kali diperoleh nilai *error* sebesar 0,095% maka dihasilkan nilai akurasi rata-rata *error* sebesar 99,905%.

#### 4.4 Integrasi Islam

Pencaraian rute terpendek menuju objek wisata Borobudur diharapkan dapat memangkas waktu di perjalanan sehingga waktu untuk berpariwisata menjadi lebih lama dengan ini wisatawan bisa bersyukur dan mengambil banyak hikmah serta pelajaran yang berharga dari berbagai kejadian di perjalanan mauapun di suatu objek wisata. Melakukan perjalanan wisata juga salah satu anjuran dalam agama Islam. Salah satu tujuan wisata dalam agama Islam adalah untuk mengambil pelajaran dan peringatan. Berwisata juga bisa digunakan sebagai sarana dakwah seperti yang dilakukan oleh para pendahulu.

Berwisata dapat meningkatkan keimanan terhadap Allah SWT dengan mengingat akan kebesaran-Nya yang tertera pada ayat suci Al-Qur'an dalam surat Luqman ayat 31 yang berbunyi:

أَمْ تَرَأَنَّ الْفُلُوكَ تَجْرِي فِي الْبَحْرِ بِنِعْمَتِ اللَّهِ لِيُرِيَكُمْ مِنْ آيَاتِهِ إِنَّ فِي ذَلِكَ لَآيَاتٍ لِكُلِّ صَبَّارٍ شَكُورٍ ۝ ٣١

*“Tidakkah kamu memperhatikan bahwa sesungguhnya kapal itu berlayar di laut dengan nikmat Allah, supaya diperlihatkan-Nya kepadamu sebahagian dari tanda-tanda (kekuasaan)-Nya. Sesungguhnya pada yang demikian itu benar-benar terdapat tanda-tanda bagi semua orang yang sangat sabar lagi banyak bersyukur”* (QS. Luqman:31).

Tafsir Jalalain:

*“(Tidakkah kamu memperhatikan bahwa sesungguhnya bahtera itu) kapal itu (berlayar di laut dengan nikmat Allah, supaya diperlihatkan-Nya kepada Kamu sekalian) hai orang-orang yang diajak bicara dalam hal ini (sebagian dari tanda-tanda kekuasaan-Nya. Sesungguhnya pada yang demikian itu benar-benar terdapat tanda-tanda) yaitu pelajaran-pelajaran (bagi semua orang yang sangat bersabar) di dalam menahan diri dari perbuatan-perbuatan maksiat yang dilarang oleh Allah (lagi banyak bersyukur) atas nikmat-nikmat-Nya”*.

Menurut Imam Syafi'i jalan-jalan memiliki manfaat antara lain:

1. Semua pertanyaan akan hilang, karena dengan berwisata dapat mengamati keadaan kaum muslimin serta dapat mengambil pengalaman yang berada di sekitar tempat wisata.
2. Akan muncul ide-ide karena sering berinteraksi dengan wisatawan lain selama di perjalanan wisata.

Seruan untuk berwisata dalam agama Islam juga terdapat dalam Hadist berikut:

Dari Abu Umamah RA bahwa ada seorang pria berkata, “Wahai Rasulullah, izinkan aku untuk berwisata!” Maka Rasulullah bersabda, “Sesungguhnya wisata umatku adalah berjihad fi sabilillah (HR. Abu Daud).

## **BAB V**

### **PENUTUP**

Bab ini berisi kesimpulan dalam penelitian ini serta saran yang dapat dijadikan acuan untuk penelitian berikutnya.

#### **5.1 Kesimpulan**

Penelitian ini menggunakan implementasi dari algoritma A\* untuk pencarian jalur terpendek dan juga AHP sebagai rekomendasi konfigurasi berdasarkan beberapa kondisi. Pengujian A\* dilakukan dengan dilakukan simulasi perjalanan sebanyak 20 kali di setiap *starting point* dan mendapatkan akurasi yang sangat baik sebesar 100% pada *starting point* Terminal Madu, dan juga memiliki akurasi sebesar 100% pada *starting point* Kebon Agung. Oleh karena itu peneliti menarik kesimpulan bahwa algoritma A\* dapat bekerja dengan baik untuk pencarian rute terpendek, kemudian untuk pengujian metode AHP dilakukan dengan melakukan simulasi perjalanan dengan kondisi berbeda sebanyak 54 kali. Pada pengujian tersebut didapatkan hasil berupa nilai error sebesar 0,095% dan nilai akurasi yang didapatkan dari rata rata error adalah sebesar 99,905%. Hal ini juga menyimpulkan bahwa AHP dapat dengan baik mencari rekomendasi kendaraan berdasarkan kondisi tertentu.

#### **5.2 Saran**

Pembuatan permainan simulasi pada penelitian ini belum bisa dikatakan sempurna, masih ditemukan adanya kekurangan yang harus diperbaiki maupun fitur

yang dapat ditambah sebagai pengembangan untuk penelitian selanjutnya. Peneliti menyarankan beberapa hal untuk pengembangan selanjutnya antara lain:

1. Mengembangkan permainan simulasi ini dengan lebih menarik lagi, dengan menambahkan beberapa animasi karakter, UI, *visual effect*, dan beberapa *game feel* agar permainan lebih *immersive*.
2. Menambahkan beberapa *asset* 3D seperti peta sesuai dengan lokasi yang di lewati, dan karakter yang dapat dimainkan.
3. Mengembangkan *game* agar mendukung *cross platform*.
4. Menambahkan beberapa kondisi pada AHP, seperti keadaan jalan baik atau banyak lubang, serta jumlah penumpang yang sedang melakukan perjalanan bersama, dan sebagainya.

## DAFTAR PUSTAKA

- Alfarisi, W. (2016). *Pencarian Jalur Terpendek Pengiriman Barang Menggunakan Algoritma A\* Studi Kasus Kantor Pos Besar Medan*. Jurnal Riset Komputer (JURIKOM), Vol. 3, No.1, Februari 2016.
- Ansah et al. (2015). *Analytic Hierarchy Process Decision Making Algorithm*. *Global Journal of Pure and Applied Mathematics* ISSN 0973-1768, Vol. 11, No.4 2015.
- Ardhianto et al. (2012). *Augmented Reality Objek 3 Dimensis dengan Perangkat Artoolkit dan Blender*. Jurnal Teknologi Informasi DINAMIK, Vol. 17, No.2, Juli 2012.
- Arif, Yunifa Miftachul et al. (2019). *Selection of Tourism Destinations Priority using 6AsTD Framework and TOPSIS*. International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), Yogyakarta: 5-6 Desember 2019. Hal. 346-351.
- Emshoff & A. Simon. (1970). *Rancangan Ulang dan Simulasi*. Social Work Jurnal ISSN, 2339-0042.
- Harrel et al. (2004). *Simulation using ProModel*. New York: McGraw-Hill, 2004.
- Masri & Tri. (2014). *Pencarian Jalur Terpendek pada Snake Game menggunakan Algoritma A\**: Jurnal Ilmiah SISFOTENIKA Vol. 4, No.1, Januari 2014.
- Pramudhita, A., & Muljono. (2018). *Aplikasi Sistem Pencarian Halte BRT Terdekat Kota Semarang menggunakan Metode A\* Berbasis Android*. Jurnal Resti Vol.2, No.1, 2018.
- Purnama et al. (2018). *Penerapan Algoritma A Star untuk Penentuan Jarak Terdekat Wisata Kuliner di Kota Bandar Lampung*. Jurnal TEKNOINFO Vol. 12, No. 1, 2018.
- Russel & Norvig. (2003). *Artificial Intelligence, A Modern Approach*. Second Edition. University of Michigan Press.
- Saaty, T.L., (1995). *The Anallic Hierarchy Process*. New York : McGraw-Hill.
- Shanon, Robert E. (1975). *System Simulation: The Art and science*. Eaglewood Cliffs, NJ : PrenticeHall.
- Setiawan, I. 2006. *Perancangan Software Embedded System Berbasis FSM*. Semarang: Universitas Diponegoro.
- Sudarmilah et al. (2013). *Tech review: Game Platform for Upgrading Counting Ability on Preschool Children*. Prosidingon The 5th International Conference on Information Technology and Electrical Engineering (ICITEE 2013).

Suyanto. (2014). *Artificial Intelligence: Searching, Reasoning, Planning, Learning*  
2nd ed. Bandung: Informatika.