

**IMPLEMENTASI ALGORITMA A-STAR DALAM MENENTUKAN RUTE
TERPENDEK DESTINASI WISATA KOTA MALANG**

SKRIPSI

**OLEH
RIYAN FAHMI SYIHABUDDIN
NIM.17610066**



**PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2021**

**IMPLEMENTASI ALGORITMA A-STAR DALAM MENENTUKAN RUTE
TERPENDEK DESTINASI WISATA KOTA MALANG**

SKRIPSI

**Diajukan Kepada
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Matematika (S.Mat)**

**OLEH
RIYAN FAHMI SYIHABUDDIN
NIM. 17610066**

**PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2021**

**IMPLEMENTASI ALGORITMA A-STAR UNTUK MENENTUKAN RUTE
TERPENDEK DESTINASI WISATA KOTA MALANG**

SKRIPSI

Oleh
Riyan Fahmi Syihabuddin
NIM. 17610066

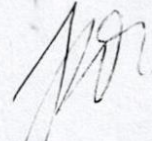
Telah Diperiksa dan Disetujui untuk Diuji
Tanggal 6 Desember 2021

Pembimbing I



Mohammad Nafie Jauhari, M.Si
NIDT. 19870218201608011056

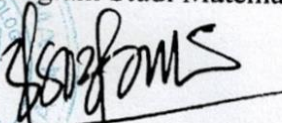
Pembimbing II



Muhammad Khudzaifah, M.Si
NIDT. 1990051120168011057



Mengetahui
Ketua Program Studi Matematika



Dr. Elly Susanti, M.Sc
NIP. 19741129 200012 2 005

**IMPLEMENTASI ALGORITMA A-STAR UNTUK MENENTUKAN RUTE
TERPENDEK DESTINASI WISATA KOTA MALANG**

SKRIPSI

Oleh
Riyan Fahmi Syihabuddin
NIM. 17610066

Telah Dipertahankan di Depan Penguji Skripsi
dan Dinyatakan Diterima sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Matematika (S.Mat)

Tanggal, 15 Desember 2021

Penguji Utama : Dr. Hairur Rahman, M.Si (.....)

Ketua Penguji : Hisyam Fahmi, M.Kom (.....)

Sekretaris Penguji : Mohammad Nafie Jauhari, M.Si (.....)

Anggota Penguji : Muhammad Khudzaifah, M.Si (.....)

Mengetahui
Ketua Program Studi Matematika


Dr. Elly Susanti, M.Sc
NIP. 19741129 200012 2 005

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Riyan Fahmi Syihabuddin

NIM : 17610066

Program Studi : Matematika

Fakultas : Sains dan Teknologi

Judul Skripsi : Implementasi Algoritma A-Star dalam Menentukan Rute
Terpendek Destinasi Wisata Kota Malang.

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

Malang, 25 November 2021
Yang membuat pernyataan



Riyan Fahmi Syihabuddin
NIM. 17610066

MOTTO

“Menjadi versi terbaik dari diri sendiri dengan memanfaatkan segala potensi yang telah dianugerahkan Allah swt”

PERSEMBAHAN

Skripsi ini penulis persembahkan untuk:

Ayahanda Abduk Kholiq yang selalu mendukung dan mendoakakan setiap langkah dan keputusan yang penulis ambil. Ibunda Sofiatun yang tidak henti-hentinya mendoakan semua anaknya di sepertiga malam terakhir. Teman-teman yang selalu membantu dalam menyelesaikan kesulitan penulis. Dosen-dosen yang telah membantu menyelesaikan skripsi penulis. Serta diri saya sendiri yang mau diajak untuk bersusah payah dalam menyelesaikan segala tantangan dan hambatan selama menempuh pendidikan strata satu.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Segala puji bagi Allah Swt yang telah melimpahkan rahmat, taufik dan hidayah-Nya, sehingga penulis mampu menyelesaikan skripsi ini yang diberi berjudul “Implementasi Algoritma A-Star dalam Menentukan Rute Terpendek Destinasi Wisata Kota Malang”, sebagai syarat untuk memperoleh gelar sarjana dalam bidang matematika di Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Dalam penyusunan skripsi ini tidak lepas bimbingan dan arahan baik secara langsung maupun tidak langsung dari berbagai pihak. Untuk itu ucapan terima kasih yang sebesar-besarnya dan penghargaan yang setinggi-tingginya penulis sampaikan terutama kepada:

1. Prof. Dr. H. M. Zainuddin, M.A, selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. Sri Harini, M.Si, selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Elly Susanti, M.Sc, selaku ketua Program Studi Matematika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Abdul Aziz, M.Si, selaku dosen wali yang telah memberikan arahan dan bimbingan sejak semester awal.
5. Mohammad Nafie Jauhari, M.Si, selaku dosen pembimbing I yang telah banyak memberikan arahan, nasihat, motivasi dan pengalaman berharga kepada penulis.
6. Muhammad Khudzaifah, M.Si, selaku dosen pembimbing II yang telah banyak memberikan arahan dan berbagi ilmunya kepada penulis.
7. Segenap sivitas akademika Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang terutama seluruh dosen, terima kasih atas segala ilmu yang diberikan dan atas segala bimbingannya.

8. Ayahanda Abdul Kholiq dan Ibunda Sofiatun serta kakak tercinta yang selalu mendoakan, memberikan semangat, dukungan dan motivasi yang luar biasa kepada penulis.
9. Seluruh teman-teman di Program Studi Matematika angkatan 2017 yang berjuang bersama-sama untuk mewujudkan mimpi dan menemani baik suka maupun duka selama pendidikan.
10. Semua pihak yang tidak dapat disebutkan satu-persatu yang telah memberikan bantuan kepada penulis dalam menyelesaikan skripsi ini baik moril maupun materiil.

Semoga Allah Swt melimpahkan rahmat dan karunia-Nya kepada kita semua. Selain itu, penulis juga berharap semoga skripsi ini dapat bermanfaat khususnya bagi penulis dan pembaca pada umumnya. *Aamiin*

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Malang, 25 November 2021

Penulis

DAFTAR ISI

HALAMAN JUDUL	
HALAMAN PENGAJUAN	
HALAMAN PENGESAHAN	
HALAMAN KEASLIAN TULISAN	
HALAMAN MOTTO	
HALAMAN PERSEMBAHAN	
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
ABSTRAK	xiv
ABSTRACT	xv
ملخص	xvi
BAB I PENDAHULUAN	
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	4
1.3 Tujuan.....	4
1.4 Manfaat.....	5
1.5 Batasan Masalah	6
BAB II TINJAUAN PUSTAKA	
2.1 Teori Graf.....	7
2.2 Algoritma <i>A-star</i>	8
2.3 Kota Malang.....	19
2.4 <i>Google Map</i>	21
2.5 <i>Geographic Information System (GIS)</i>	23
2.6 Python.....	25
2.6.1 Networkx	26
2.7 Kajian Keislaman Tentang Perjalanan (<i>Safar</i>).....	27

BAB III METODOLOGI PENELITIAN

3.1 Pendekatan Penelitian	30
3.2 Sumber Data.....	30
3.3 Identifikasi Variabel.....	31
3.4 Tahapan Penelitian	32

BAB IV PEMBAHASAN

4.1 Deskripsi Data.....	35
4.2 Pengujian Pencarian Rute Terpendek	39
4.3 Hasil Pengujian	57
4.4 Evaluasi Pengujian	60

BAB V PENUTUP

5.1 Kesimpulan	64
5.2 Saran.....	65

DAFTAR PUSTAKA	66
-----------------------------	-----------

LAMPIRAN

DAFTAR RIWAYAT HIDUP

DAFTAR GAMBAR

Gambar 2.1	Flowchart A-Star	13
Gambar 2.2	Contoh Algoritma A-Star Bekerja	15
Gambar 2.3	Graf Label	17
Gambar 2.4	Gambar Data Grafis Berupa (a) Titik, (b) Garis, dan (c) Area	24
Gambar 2.5	Contoh Struktur Data GIS, Bagian Atas Struktur Data Vektor, Bagian Bawah Raster	24
Gambar 2.6	Cara Penyimpanan Data Atribut Pada Struktur Data Vektor, (a) Data Grafis dan (b) Data Atribut	25
Gambar 3.1	Flowchart Alur Pengujian Rute.....	34
Gambar 4.1	Peta Jalan Kota Malang.....	37
Gambar 4.2	Visualisasi Graf Kota Malang.....	39
Gambar 4.3	Hasil Pengujian Rute Terpendek.....	57
Gambar 4.4	Visualisasi Rute Pada Maps	58

DAFTAR TABEL

Tabel 2.1	Closed <i>List</i> 1	17
Tabel 2.2	Open <i>list</i> 1	18
Tabel 2.3	Closed <i>List</i> 2	18
Tabel 2.4	Open <i>List</i> 2	18
Tabel 2.5	Closed <i>List</i> 3	19
Tabel 2.6	Open <i>List</i> 3	19
Tabel 2.7	Closed <i>List</i> 4	19
Tabel 3.1	Variabel Penelitian	31
Tabel 4.1	Kode Wisata Sejarah	35
Tabel 4.2	Kode Wisata Taman	35
Tabel 4.3	Kode Wisata Kampung Tematik	36
Tabel 4.4	Kode Wisata Religi	36
Tabel 4.5	Kode Wisata Hiburan	36
Tabel 4.6	Tabel Kode Kecamatan	38
Tabel 4.7	Tabel Format Penulisan Data	38
Tabel 4.8	Tabel Contoh Pengujian Rute	40
Tabel 4.9	Hasil Percobaan Pengujian Rute	59
Tabel 4.10	Hasil Persentase Keefektifan Rute	61

ABSTRAK

Syihabuddin, Riyan Fahmi. 2021. **Implementasi Algoritma A-Star dalam Menentukan Rute Terpendek Destinasi Wisata Kota Malang**. Skripsi. Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing (1) : Mohammad Nafie Jauhari, M.Si, Pembimbing (2) : Muhammad Khudzaifah, M.Si.

Kata Kunci : heuristik, *list* tutup, *list* buka, euclid, *a-star*.

Berwisata menjadi salah satu kebutuhan setiap orang untuk merelaksasikan pikiran dari kesibukan yang dijalani di setiap harinya. Salah satu kota di Jawa Timur yang menjadi tempat primadona untuk berwisata adalah Kota Malang, kota terbesar kedua di Jawa Timur setelah Surabaya. Kota ini memiliki kurang lebih 43 destinasi wisata. Biasanya wisatawan yang ingin berkunjung tidak hanya ke salah satu tempat, namun beberapa tempat. Akibatnya dibutuhkan sebuah bantuan untuk menentukan destinasi mana saja yang harus dikunjungi terlebih dahulu agar perjalanan berwisata mereka menjadi efektif. Proses pencarian rute terpendek pada penelitian ini menggunakan Algoritma A-Star, salah satu algoritma BFS yang dalam prosesnya sangat mempertimbangkan nilai heuristik. Nilai heuristik diperoleh dari jarak setiap titik menuju titik tujuan. Proses pengujian rute terpendek dilakukan dengan memilih titik awal, kemudian memilih beberapa lokasi wisata. Selanjutnya akan dicari rute terpendek menggunakan algoritma A-star pada setiap destinasi, kemudian dipilih destinasi mana yang akan dikunjungi pertama. Begitupun seterusnya sampai destinasi terakhir. Keefektifan rute melibatkan perbandingan dengan rute yang disajikan oleh *google maps*. Berdasarkan hasil 30 kali percobaan pada beberapa destinasi yang bersifat menyeluruh, didapatkan hasil bahwa rata-rata pencarian rute menggunakan algoritma A-Star adalah 44,17% lebih pendek dibandingkan yang tersaji pada *google maps*. Hal ini dikarenakan keunikan dari algoritma di mana terdapat nilai heuristik dan penyeleksian pada tiap destinasi sehingga menjadikan rute lebih efektif.

ABSTRACT

Syihabuddin, Riyan Fahmi. 2021. **An Implementation of A-Star Algorithm to Determine The Shortest Path of Malang Tourism Objects**. Thesis. Department of Mathematics, Faculty of Science and Technology, Maulana Malik Ibrahim State Islamic University of Malang. Advisor (1) : Mohammad Nafie Jauhari, M.Si, Advisor (2) : Muhammad Khudzaifah, M.Si.

Keywords: heuristic, closed list, open list, euclid, a-star.

Traveling is one of the necessary for everyone to relax the mind from the busy work daily activities. One of the most favorite cities in East Java for traveling is Malang City, the biggest city in East Java after Surabaya. This city has approximately 43 tourist destinations. The tourists usually want to visit not only just one place, but several places. As a result, an assistance is needed to determine which destinations must be visited first, so their travel trip becomes more effective. The process of finding the shortest path in this research uses the A-Star Algorithm, one of the BFS algorithms which in the process really considers the heuristic values. The heuristic values are obtained from the distance of each nodes to the destination node. The process of testing the shortest path is done by selecting a starting node, then selecting several tourist destinations. Next, the shortest path will be searched using the A-star algorithm at each destination, then which destination to visit first will be chosen. This process will be run until the final destination has been visited. The effectiveness of the path will involve the comparison between the final path of research and final path presented by google maps. Based on the results of comprehensive 30 trials on several destinations, the result show that the average of final path using the A-Star algorithm was 44.17% shorter than presented on google maps. This is due to the uniqueness of the algorithm in which there are a selection at each destination and heuristic values, so it make the final path more effective.

ملخص

شهاب الدين ، ريان فهمي . ٢٠٢١. فعالية خوارزمية (*A-Star*) في تحديد أقصر طريق إلى الوجهات السياحية لمدينة مالانج . البحث الجامعي . قسم الرياضيات، كلية العلوم والتكنولوجيا ، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف: (١) محمد نافع جوهرى الماجستير (٢) المشرف : محمد حذيفة الماجستير

الكلمات المفتاحية :الاستدلال، القائمة المغلقة (*List tutup*)، القائمة المفتوحة (*List buka*)، الإقليدية (*Euclid*)، (*A-star*)

الرحلة هي من احتياجات للناس جميعا لإرخاء الذهن من الانشغال اليومي . مدينة مالانج هي إحدى المدن في جاوي الشرقية وهي من المدن الجملة للرحلة. وهي المدينة الكبيرة الثانية في جاوي الشرقية بعد سورابايا. وفيها هناك ٤٣ وجهة سياحية. ومن العادة ، كان السياح لا يحبّ للزيارة في مكان واحد، ولكن عدة أماكن. ولذلك ، تحتاج إلى مساعدة لتحديد الوجهات التي يجب زيارتها أولاً حتى تكون رحلات الرحلة الخاصة بهم فعالة. وأما عملية إيجاد أقصر طريق في هذه الدراسة تستخدم خوارزمية (*A-Star*) . إحدى خوارزميات (*BFS*) والتي تعتبر في العملية القيم الاستدلالية حقًا. يتم الحصول على القيمة الإرشادية من مسافة كل نقطة إلى نقطة الوجهة. تتم عملية اختبار أقصر طريق عن طريق اختيار نقطة البداية ثم اختيار عدة مواقع سياحية. بعد ذلك ، سيتم البحث عن أقصر طريق باستخدام خوارزمية (*A-star*) في كل وجهة ، ثم الوجهة التي سيتم زيارتها أولاً. وهكذا حتى الوجهة النهائية. ستشمل فعالية المسار المقارنة مع المسار الذي قدمته خرائط (*Google*) . بناءً على نتائج ٣٠ تجربة على عدة وجهات شاملة ، وجد أن متوسط البحث عن المسار باستخدام خوارزمية (*A-Star*) كان أقصر بنسبة % ٤٤,١٧ من ذلك المعروف على خرائط (*Google*) . ويرجع ذلك إلى تفرد البحث حيث توجد قيمة إرشادية واختيار لكل وجهة لجعل المسار أكثر فعالية .

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kota Malang merupakan salah satu kota terbesar di Jawa Timur setelah Kota Surabaya. Kota ini memiliki semboyan Malang Kucecwara yang berarti menjunjung keadilan dan membasmi kebatilan. Tri Bina Cita Kota Malang merupakan tiga aspek yang sangat dijunjung Kota Malang yaitu Malang sebagai kota pendidikan, Malang sebagai kota industri, dan Malang sebagai kota pariwisata. Malang sebagai kota pendidikan ditunjukkan dengan berdirinya 66 universitas baik berstatus negeri maupun berstatus swasta. Contohnya adalah Universitas Brawijaya, UIN Maliki Malang, Universitas Negeri Malang, Universitas Muhammadiyah Malang, Politeknik Negeri Malang, Universitas Islam Malang, dan lain sebagainya. Malang sebagai kota industri ditunjukkan dengan berkembangnya berbagai industri, mulai dari industri keramik, industri keripik tempe, industri batik topangan malang, dan lain sebagainya. Meskipun industri hanya berskala menengah, namun tingkat daya saingnya tidak dapat diremehkan. Terakhir, Malang sebagai kota pariwisata. Hal ini ditunjukkan dengan berbagai destinasi wisata yang tersebar di Kota Malang (Badan Pusat Statistik Kota Malang, 2019).

Pertumbuhan destinasi wisata Kota Malang mengalami kenaikan tiap tahun. Hal ini ditunjukkan data kunjungan wisatawan baik domestik maupun mancanegara setiap tahunnya yang meningkat (Disporapar, 2019). Tidak adanya potensi wisata alam menjadikan Pemerintah Kota Malang berpikir bagaimana cara menciptakan pariwisata yang menjadi kebutuhan serta inspirasi bagi kota-kota besar yang ada di

Indonesia. Hal inilah yang menjadikan Kota Malang unik dari daerah lainnya. Salah satu upaya pemerintah adalah memperbanyak kampung tematik. Kampung tematik merupakan kampung wisata berbasis masyarakat di mana sangat menonjolkan sumber daya masyarakat di daerah tersebut. Namun, banyak destinasi wisata yang telah dibuat tidak sebanding dengan informasi bagaimana cara mengakses destinasi tersebut.

Ajaran Islam mengajarkan bahwa usaha untuk meningkatkan pemahaman informasi berwisata harus dikaitkan dengan tujuan yang mulia. Sebagaimana yang telah difirmankan Allah swt:

هُوَ الَّذِي جَعَلَ لَكُمُ الْأَرْضَ ذُلُولًا فَامْشُوا فِي مَنَاكِبِهَا وَكُلُوا مِن رِّزْقِهِ ^طوَإِلَيْهِ
النُّشُورُ ۚ

Dialah (Allah) yang menjadikan bumi itu mudah bagimu (untuk ditelusuri), maka berjalanlah (bepergianlah) ke segala penjurunya dan makanlah sebagian dari rezeki-Nya. Dan hanya kepada-Nya-lah kamu (kembali setelah) dibangkitkan.(Q.S Al-Mulk:15)

Ayat ini memberikan pemahaman kepada manusia tentang bagaimana berwisata dalam islam. Dalam islam berwisata erat kaitannya dengan safar, di mana hal tersebut digunakan untuk merenungi keindahan ciptaan Allah swt serta menikmati indahnya alam sebagai penggerak jiwa manusia untuk meningkatkan keimanan dan memotivasi diri agar lebih semangat menjalani hidup (Hidayati, 2017).

Permasalahan penentuan akses atau rute suatu perjalanan erat kaitannya dengan Teori Graf. Teori Graf merupakan himpunan dari titik-titik $V(G)$ dan himpunan sisi $E(G)$ yang menghubungkan beberapa titik atau yang disebut dengan

edge (Munir, 2016). Kajian ilmu ini sering diaplikasikan dalam berbagai bidang, contohnya adalah penentuan jadwal sebuah kegiatan, penentuan warna daerah peta, penentuan aliran listrik dan lain sebagainya. Hubungan kajian ilmu ini dengan permasalahan destinasi wisata Kota Malang adalah tentang peta wisata Kota Malang. Titik-titik destinasi wisata Kota Malang dapat dijadikan titik-titik dari sebuah graf yang kemudian dari titik-titik tersebut dapat dihubungkan. Dengan menghubungkan titik-titik tersebut akan dicari solusi bagaimana ketika diambil sembarang titik, akan ditemukan rute terpendek menuju destinasi wisata yang diinginkan. Terdapat banyak algoritma dalam menentukan rute terpendek, salah satunya adalah algoritma *A-star*.

Algoritma *A-star* merupakan perluasan algoritma Dijkstra, namun dalam algoritma ini sangat dipengaruhi oleh nilai heuristik. Nilai heuristik adalah nilai yang diperoleh dari menarik garis lurus dari suatu titik yang dilewati menuju titik yang menjadi tujuan. Pertimbangan nilai heuristik dalam penentuan jalur ini merupakan salah satu kelebihan yang dimiliki oleh algoritma ini. Hal ini menjadi alasan dipilihnya algoritma *A-star* dalam permasalahan penentuan rute terpendek destinasi wisata Kota Malang. Selain alasan tersebut pemilihan algoritma ini juga berdasarkan pada penelitian-penelitian sebelumnya yang telah menunjukkan pengimplementasian algoritma dalam permasalahan penentuan rute atau jalur terpendek.

Penelitian yang dilakukan oleh Dona, dkk (2020) menunjukkan bahwa penggunaan algoritma *A-star* sangat efisien dalam menyelesaikan permasalahan jalur termudah untuk menemukan restoran Padang di Palembang. Penelitian tersebut juga mengungkapkan penentuan nilai heuristik sangat membantu dalam

memilih jalur yang tepat secara cepat. Selain itu, penelitian yang dilakukan oleh Sandra, dkk (2019) juga mengungkapkan penggunaan algoritma *A-star* sangat sesuai untuk menentukan rute perjalanan wisata di Kota Bogor. Rute perjalanan dapat dibuat seminimal mungkin sehingga menghemat total biaya akomodasi dalam berkeliling wisata di Bogor. Penelitian-penelitian tersebut menunjukkan bahwa hasil pengimplementasian algoritma *A-star* cukup baik.

Inovasi terhadap permasalahan peta wisata Kota Malang dapat diselesaikan dengan mengimplementasikan algoritma *A-star* untuk menentukan rute yang dapat dilalui oleh wisatawan ketika berkunjung ke Malang. Selain itu wisatawan juga dapat memperkirakan waktu dan biaya agar lebih efisien ketika berkunjung ke Kota Malang. Ulasan-ulasan yang telah dipaparkan menjadikan pondasi yang kuat dilakukannya penelitian ini. Sehingga fokus penelitian ini adalah pengimplementasian algoritma *A-star* untuk menentukan rute terpendek destinasi wisata Kota Malang.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam penelitian ini adalah bagaimana efektivitas penerapan algoritma *A-star* dalam menentukan rute terpendek destinasi wisata Kota Malang?

1.3 Tujuan

Tujuan dari dilakukannya penelitian ini adalah untuk melihat efektivitas algoritma *A-star* dalam menentukan rute terpendek destinasi wisata di Kota Malang.

1.4 Manfaat

Manfaat dari penelitian ini dapat dibedakan menjadi tiga aspek. Ketiga aspek tersebut adalah:

1. Kepariwisataaan

Manfaat yang didapatkan dalam aspek kepariwisataan adalah:

- a. *Branding* pariwisata, maksud dari hal ini adalah peningkatan integritas serta level dari pariwisata khususnya pariwisata Kota Malang.
- b. Inovasi dalam bidang pariwisata Kota Malang. Inovasi yang terbaru serta memanfaatkan kecanggihan teknologi dapat menjadi senjata ampuh dalam hal promosi.

2. Pemerintah

Manfaat yang dapat berdampak kepada pemerintah adalah:

- a. Perwujudan *SDGs* Pemerintah Indonesia 2030. *SDGs* merupakan program pembangunan berkelanjutan, salah satunya dalam segi teknologi.
- b. Kontribusi perwujudan *smart city of* Malang. Satu langkah kontribusi yang kita lakukan dapat memberikan dampak untuk terwujudnya cita-cita tersebut.

3. Wisatawan

Manfaat yang diperoleh oleh wisatawan yang berkunjung ke Kota Malang adalah:

- a. Membantu wisatawan menuju destinasi wisata. Sehingga wisatawan dapat menghemat waktu dan meminimalkan biaya yang dikeluarkan.
- b. Menarik wisatawan untuk berkunjung kembali. Memberikan pelayanan yang baik untuk wisatawan akan memberi kesan untuk kembali berkunjung.

1.5 Batasan Masalah

Batasan masalah dalam penelitian ini ada tiga yaitu:

1. Objek yang menjadi kajian penelitian adalah daerah wisata Kota Malang, bukan Malang Raya.
2. Penentuan rute hanya didasarkan pada kondisi jalan utama menuju destinasi wisata Kota Malang.
3. Penelitian ini hanya menggunakan bobot jarak, untuk waktu diabaikan karena kondisi lalu lintas di Kota Malang tidak menentu.

BAB II

TINJAUAN PUSTAKA

2.1 Teori Graf

Teori graf sebagai salah satu cabang matematika sudah terdapat lebih dari 200 tahun silam yang pertama kali diperkenalkan pada tahun 1736 oleh Leonhard Euler untuk menyelesaikan permasalahan jembatan *Konigsberg* (Cipta, 2019). Dari segi matematika, pada periode awalnya tidak terlalu diperhatikan karena pengimplementasiannya hanya sebagai pemecahan solusi *puzzle* (teka-teki). Selang satu abad kemudian pada tahun 1874 seorang matematikawan bernama G.E Kirchof berhasil mengembangkan teori pohon (*Theory of Trees*) yang digunakan di dalam persoalan jaringan listrik. Setelah penemuan tersebut perkembangan teori graf mulai mendapat perhatian, sehingga muncullah pengaplikasian teori graf dalam bidang keilmuan lainnya seperti pada bidang ilmu komputer, teknik, sains, dan lain-lain (Budayasa, 2016).

Graf G didefinisikan sebagai pasangan $(V(G), E(G))$ dengan $V(G)$ adalah himpunan tidak kosong dan berhingga dari objek-objek yang disebut titik. Sedangkan $E(G)$ adalah himpunan (mungkin kosong) pasangan tak berurutan dari titik-titik berbeda di $V(G)$ yang disebut sisi. Banyaknya unsur di $V(G)$ disebut order dari G dan dilambangkan dengan $p(G)$, dan banyaknya unsur di $E(G)$ disebut ukuran dari G dan dilambangkan dengan $q(G)$. Jika graf yang dibicarakan hanya graf G , maka order dan ukuran dari G masing-masing cukup ditulis p dan q . Graf dengan order p dan ukuran q dapat disebut graf $-(p, q)$ (Abdussakir, 2009).

Sisi $e = (u, v)$ disebut menghubungkan antara titik u dan v jika $e = (u, v)$ adalah sisi di graf G , maka u dan v disebut terhubung langsung (*adjacent*). Disisi lain, v dan e serta u dan e dapat dikatakan terkait langsung (*incident*). Titik u dan v disebut ujung dari e . Graf yang tidak mempunyai titik (tidak mempunyai garis) disebut graf kosong (*null graph*).

Berdasarkan jenis garisnya, graf dapat diklasifikasikan menjadi dua jenis. Jenis pertama adalah graf yang berarah (*directed graph*) dan graf tak berarah (*undirected graph*). Jenis graf yang semua garisnya memiliki arah yang menunjukkan titik asal dan tujuan garis yang bersangkutan merupakan graf yang berarah. Sebaliknya, jika semua garis pada graf tersebut tidak memiliki arah, maka graf tersebut disebut graf tak berarah. Jika penyebutannya hanya graf saja, maka yang dimaksud adalah graf tak berarah. Suatu graf biasanya dipresentasikan secara grafis, dengan setiap *vertex* disimbolkan sebagai titik atau lingkaran kecil, dan setiap sisi (*edge*) dapat disimbolkan dengan $e = uv$ dengan penggambarannya sebagai sebuah garis atau kurva yang menghubungkan titik-titik yang bersesuaian dengan u dan v (Budayasa, 2016).

2.2 Algoritma A-star

A-star merupakan salah satu jenis algoritma pencarian jalur terpendek terbaik pada graf yang mampu menemukan jalur dengan biaya pengeluaran minimum dari titik awal sampai ke titik yang menjadi tujuan. Algoritma ini menerapkan fungsi *distance-plus-cost* (biasanya di notasikan dengan $f(n)$) untuk menentukan urutan kunjungan pencarian *node* di dalam Graf. Gabungan jarak dan biaya merupakan penjumlahan dari dua fungsi. Fungsi yang pertama yaitu fungsi

path-cost yang dinotasikan dengan $h(n)$. Fungsi ini adalah jumlah biaya atau beban jarak tempuh yang harus dikeluarkan dari *node* awal menuju *node* yang akan dituju. Fungsi kedua adalah sebuah fungsi heuristik yaitu nilai jarak yang dihitung dari titik ke- $n + 1$ ke titik tujuan, biasanya dinotasikan dengan $g(n)$ (Hart, 1995).

Nilai heuristik digunakan dengan tujuan untuk mempersempit ruang pencarian. Metode pencarian *A-star* menghasilkan jalur optimal mulai dari tempat awal menuju tempat yang menjadi tujuan. Metode ini berdasarkan formula:

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

Keterangan:

$h(n)$ = biaya estimasi dari node n ke tujuan.

$g(n)$ = biaya path / perjalanan

$f(n)$ = solusi biaya estimasi termurah *node* n untuk mencapai tujuan.

$h(n)$ ditentukan dengan melakukan perhitungan dengan rumus *Heuristic Euclidean Distance*. Berikut adalah rumus dari *HED* di mana a dan b merupakan kedudukan dua titik yang akan dihitung:

$$d(a, b) = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2} \quad (2)$$

Keterangan :

$d(a, b)$ = jarak antara titik a dan b

x = nilai absis suatu titik

y = nilai ordinat suatu titik

Semakin akurat nilai heuristik yang digunakan, semakin sedikit perulangan yang akan dilakukan oleh algoritma *A-star* sehingga semakin cepat dijalankan. Jika nilai heuristik yang ditemukan tepat (selalu mengembalikan jarak rute minimum antara dua *node* sangat tepat), maka *A-star* akan menunjukkan pada hasil pencarian yang tepat sehingga algoritma akan menjadi $O(p)$, di mana p adalah jumlah langkah dalam *path* (rute). Penentuan nilai heuristik tepat atau tidaknya tergantung pada apakah heuristik tersebut terlalu rendah atau terlalu tinggi (Fernando, 2020).

a. Heuristik yang Terlalu Rendah

Jika heuristik terlalu rendah, maka *A-star* memperkirakan terlalu rendah panjang rute asli, sehingga algoritma *A-star* membutuhkan waktu lebih lama untuk dijalankan. Perkiraan jarak tempuh akan lebih spesifik menjadi kurang efisien karena nilai heuristik terlalu kecil dari kenyataan aslinya. Jika nilai heuristik terlalu rendah, maka hasil yang dihasilkan akan menjadi *path* terbaik yang memungkinkan, dan akan menjadi *path* yang benar-benar sama dengan yang dihasilkan oleh Algoritma *Dijkstra*.

b. Heuristik yang terlalu Tinggi

Jika nilai heuristik terlalu tinggi, maka dapat diprediksi panjang dari rute asli terlalu panjang. Kemungkinan *A-star* tidak dapat menghasilkan jalur terbaik. Perkiraan nilai total jarak akan membias kepada fungsi heuristik. Algoritma *A-star* lebih tidak memperhatikan secara baik jarak sejauh ini dan akan lebih efektif jika node memiliki sisa jarak yang lebih kecil. Hal ini akan

merusak fokus pencarian ke tujuan lebih cepat. Hal tersebut menandakan bahwa panjang total dari jalur diprediksi lebih besar dari panjang total dari jalur terbaik. Namun, tidak bermakna bahwa akan langsung didapatkan jalur yang buruk.

Pengimplementasian algoritma *A-star* dengan fungsi heuristik yang tepat dapat memberikan hasil yang optimal. Hakikatnya, *Depth-first search (DFS)* dan *Breadth First-Search (BFS)* adalah contoh dari dua kasus khusus algoritma *A-star*. Algoritma Dijkstra yang merupakan perluasan dari salah satu BFS, adalah kasus khusus dari *A-star* di mana nilai $h(n) = 0$ untuk setiap nilai x . DFS membuat suatu *counter global C* yang diinisialisasi dengan nilai yang cukup besar. Pada setiap langkah-langkahnya akan diperiksa pada setiap titik, kemudian diberikan nilai C ke semua titik yang bertetangga dengan titik tersebut. Setelah tiap-tiap pemberian nilai, kurangi *counter C* dengan 1. Jadi, semakin awal sebuah titik diproses, akibatnya semakin tinggi nilai $h(n)$ yang dimilikinya (Maulana, 2017).

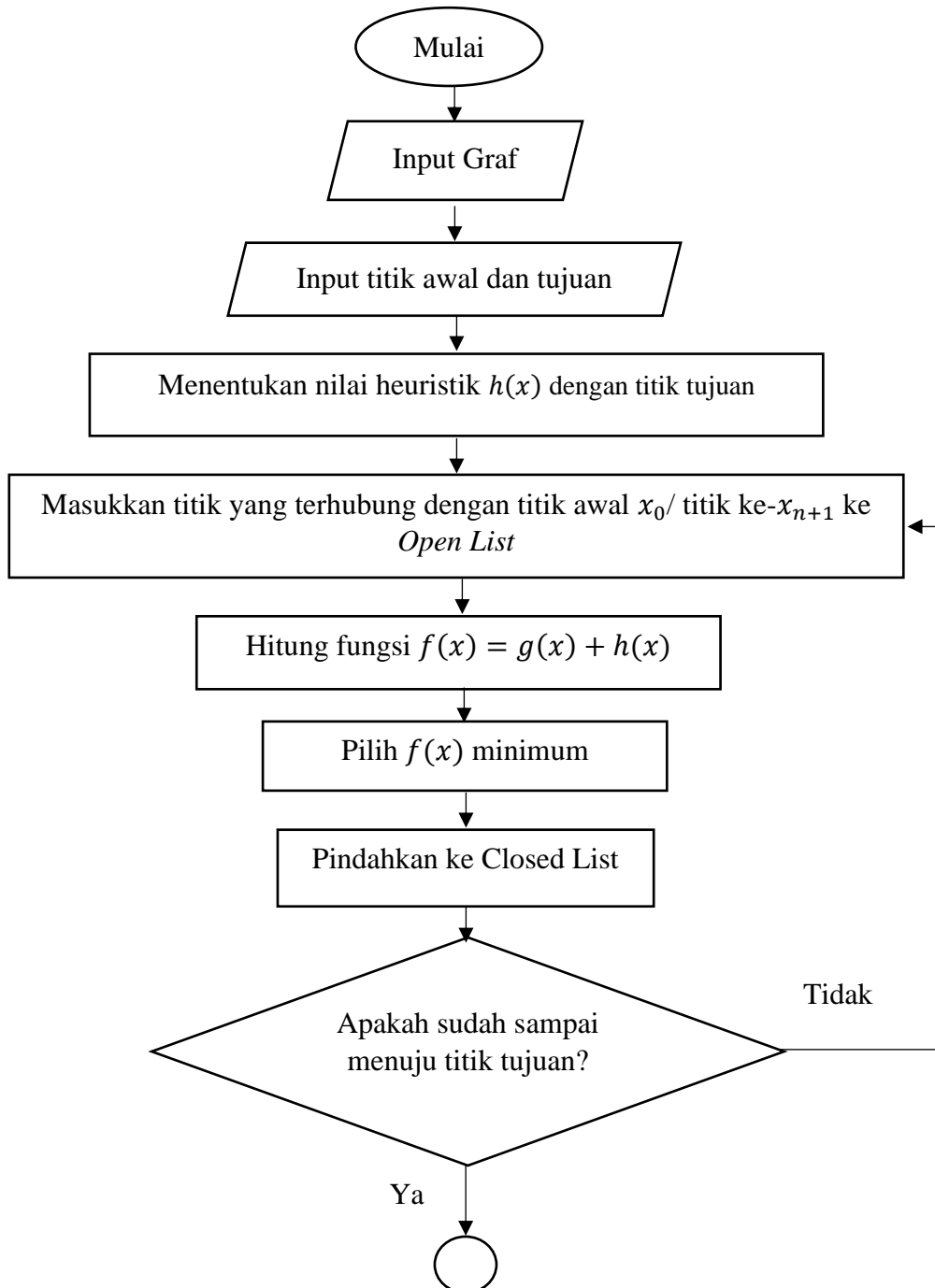
Beberapa terminologi dasar yang terdapat pada algoritma *A-star* adalah:

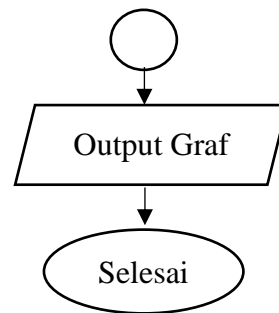
1. *Starting point* adalah sebuah terminologi untuk posisi awal sebuah titik.
2. *A* adalah simpul yang sedang dijalankan dalam algoritma pencarian rute terpendek.
3. Simpul adalah petak-petak kecil sebagai simbol dari daerah *pathfinding*.
4. *Open list* adalah tempat penyimpanan data simpul yang dapat diakses dari titik awal maupun simpul yang sedang dijalankan.
5. *Closed list* adalah tempat penyimpanan data simpul sebelum *A* yang merupakan bagian dari jalur terpendek.
6. Harga (F) adalah nilai yang diperoleh dari proses penjumlahan, nilai g

merupakan jumlah nilai tiap simpul dalam jalur terpendek dari titik awal ke A , dan H adalah jumlah perkiraan nilai dari sebuah simpul ke simpul tujuan.

7. Simpul tujuan yaitu simpul yang menjadi tempat akhir perjalanan.

Berikut merupakan Flowchart Algoritma A-Star:





Gambar 2.1 Flowchart A-Star

Berikut akan dijelaskan langkah-langkah algoritma A-star:

a. Memproses *Current Node*

Saat proses iterasi, A-star memperhatikan setiap koneksi yang keluar dari *current node*. Pada tiap koneksi, A-star menemukan *end node* dan menyimpan *total cost* dari *path* dan koneksi sampai ke titik lain. Terlebih lagi, A-star menyimpan satu nilai lagi yaitu nilai perkiraan dari *cost* total untuk sebuah *path* dari node awal menuju *current node* dan menuju tujuan (*total cost*). Perkiraan ini adalah jumlah dari dua nilai yaitu nilai *cost* sejauh yang ditentukan dan nilai seberapa jauh dari node tersebut hingga node tujuan. Perkiraan-perkiraan ini dinamakan “nilai heuristik” dari sebuah node dan nilainya tidak dapat negatif. Penentuan dari nilai heuristik adalah perhatian utama dalam pengimplementasian algoritma A-star.

b. Node List

Sama halnya dengan sebelumnya, algoritma ini menyimpan daftar dari *open node* yang telah dilewati namun belum diproses sedangkan *closed node* yang telah diproses. Node kemudian dipindahkan ke *open list* saat ketika didapati pada ujung koneksi. Sedangkan node dipindahkan ke *closed list* setelah diproses di dalam iterasi masing-masing. Tidak sama dengan sebelumnya, node

dari *open list* dengan perkiraan *total cost* minimum dipilih pada setiap iterasi, sehingga selalu berbeda dengan *cost* terkecil terbaru. Perubahan ini menjadikan algoritma tersebut memeriksa node yang lebih tepat terlebih dahulu. Jika node memiliki perkiraan *total cost* yang kecil, maka node tersebut pasti memiliki nilai *cost* yang relatif pula. Jika perkiraannya sangat tepat, maka node-node yang lebih dekat dengan tujuan akan dipertimbangkan terlebih dahulu. Hal ini mempersempit pencarian ke area yang paling menguntungkan.

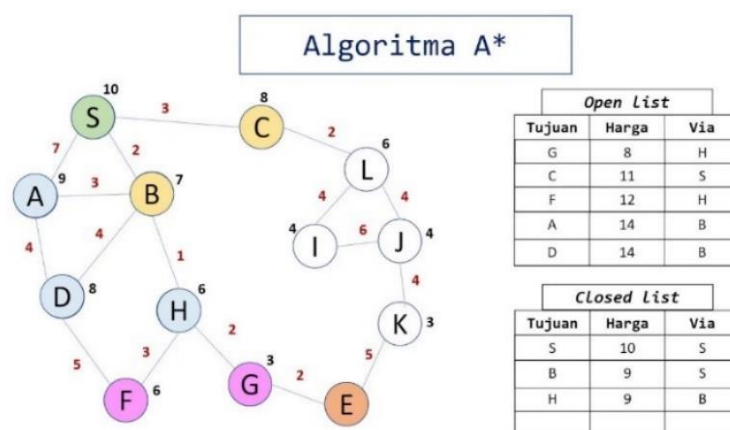
c. Menghitung Jarak untuk *Open* dan *Closed Nodes*

Sama halnya dengan sebelumnya, setelah sampai pada *open* dan *closed node* proses iterasi, selanjutnya nilai yang terekam harus diperbarui. Nilai *cost* terbaru ini terhitung normal dan jika nilai baru lebih rendah dari nilai untuk node yang sudah ada, maka nilainya perlu diperbarui. Perbandingan ini secara ketat didasarkan pada nilai *cost* terbaru di mana hanya nilai ini yang dapat diandalkan, karena tidak mengandung nilai perkiraan apapun, bukan perkiraan *total cost*. Tidak sama dengan algoritma Dijkstra, algoritma *A-star* dapat menemukan rute yang lebih baik menuju node yang sudah terdapat pada *closed list*. Jika nilai sebelumnya sangat diperhitungkan pemilihannya, maka node telah diproses menandakan bahwa itu adalah pilihan terbaik padahal kenyataannya bukan. Hal ini mengakibatkan *knock-on problem*. Jika node yang samar-samar telah diproses dan dimasukkan dalam *closed list*, hal ini menandakan semua koneksinya telah diperhitungkan. Bisa jadi seluruh himpunan node mempunyai nilai *cost* terbaru berdasarkan node terbaru dari node yang samar. Perbaruan nilai dari node yang samar tidaklah cukup. Seluruh koneksinya juga harus diperiksa dan diproses kembali untuk mendistribusikan nilai baru tersebut. Dalam kasus

perbaruan node dalam *open list*, proses ini tidak diperlukan kembali karena telah diketahui bahwa koneksi dari sebuah node dalam *open list* belum diproses. Namun terdapat salah satu cara sederhana untuk memaksa algoritma tersebut menghitung kembali dan menyebarkan nilai baru tersebut. Node dari *closed list* dapat dipindahkan kembali ke *open list*, kemudian akan menunggu sampai tertutup dan koneksinya yang telah dipertimbangkan kembali. Node yang bergantung pada nilainya selanjutnya akan diproses kembali sebanyak satu kali. Jadi, *closed node* nilainya telah direvisi dikeluarkan dari *closed list* dan ditempatkan ke *open list*. *Open nodes* yang memiliki nilai yang terbaru tetap pada *open list*, seperti hal sebelumnya.

d. Mengakhiri Algoritma

A-star akan berakhir saat node tujuan adalah node terkecil pada *open list*. Akan tetapi node yang memiliki nilai perkiraan *total cost* terkecil (akan diproses pada iterasi selanjutnya dan dimasukkan dalam *closed list*) bisa saja nilainya harus diperbarui. Tidak ada lagi jaminan bahwa hanya karena node tersebut paling kecil di *open list*, maka *shortest path* akan didapatkan. Oleh karena itu saat mengakhiri A-star saat node tujuan merupakan nilai terkecil di *open list* dan tidak akan menjamin bahwa rute terpendek telah ditemukan.



Gambar 2.2 Contoh Algoritma A-Star Bekerja

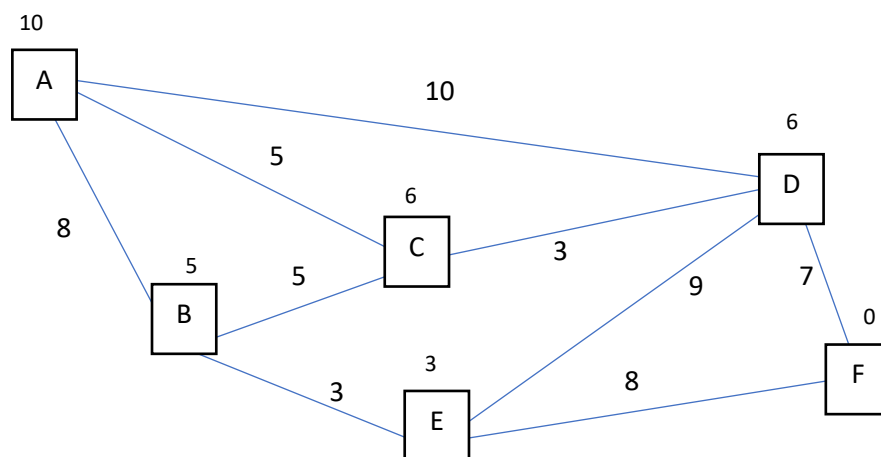
Berdasarkan hal tersebut, sangat wajar jika ada yang bertanya apakah algoritma *A-star* dapat dijalankan sedikit lebih lama untuk menghasilkan hasil yang terjamin optimal. Hal ini dapat dilakukan dengan menjadikan algoritma tersebut untuk berakhir hanya jika node di *open list* dengan *cost* sejauh ini terkecil memiliki nilai *cost* terbaru lebih besar dari *cost path* yang ditemui menuju node tujuan. Dengan hal tersebut dapat dijamin bahwa tidak ada *path* selanjutnya yang akan ditemukan yang membentuk *shortcut*. Ini merupakan kondisi pengakhiran yang paling efektif yang dapat dilihat pada Dijkstra, dan dapat dilihat bahwa penerapan kondisi tersebut akan menghasilkan jumlah *fill* yang sama halnya menggunakan algoritma *pathfinding* Dijkstra. Node bisa dicari dengan urutan yang berbeda dan bisa saja terdapat perbedaan dalam himpunan node-node di *open list*, namun tingkat *fill* rata-rata sama. Dengan kata lain, Dijkstra mengambil seluruh kelebihan cara kerja dari algoritma *A-star* dan membuatnya tidak bernilai. Pengimplementasi seluruh *A-star* bergantung pada kenyataan bahwa *A-star* dapat menghasilkan hasil yang tidak optimal secara teoritis. Namun, hal ini dapat dipatahkan dengan fungsi heuristik. Pemilihan fungsi heuristik sangat diperhitungkan, hasil yang maksimal akan terpenuhi atau hasil sub-maksimal yang memberikan eksekusi yang lebih cepat dapat dan akurat. Karena *A-star* sering berhubungan dengan hasil maksimal, pengimplementasian *A-star* dalam jumlah besar malah berakhir saat node tujuan dikunjungi pertama tanpa menunggu untuk menjadi terkecil di *open list*. Kelebihan cara kerja *A-star* tidak sebesar melakukan sama halnya dengan Dijkstra, namun banyak pengembang berpendapat bahwa setiap bit berharga terutama saat algoritma tidak dibutuhkan untuk menjadi maksimal.

e. Mengambil *Path*

Path terakhir didapatkan dengan cara yang sama seperti halnya sebelumnya yaitu dengan memulai dari *node* tujuan dan mengakumulasikan koneksi saat bergerak mundur ke *start node* (*node* awal). Koneksi-koneksi tersebut kemudian dibalikkan kembali untuk membentuk rute yang benar dan memiliki hasil yang optimal (Sandra, 2019).

Misalkan akan ditentukan rute terpendek dari titik A ke titik F. Berikut langkah-langkahnya:

1. Kita tentukan terlebih dahulu nilai heuristik dari semua titik terhadap titik F dengan menggunakan perhitungan pada persamaan (2). Kita dapatkan hasil sebagai berikut:



Gambar 2.3 Graf Label

2. Titik pertama kita masukkan dalam *closed list* sebagaimana pada tabel 2.1.

Tabel 2.1 *Closed List* 1

No	Nama titik-titik jalur	Total Jarak
1	A	10

3. Titik A berhubungan dengan titik B, C, dan D. hitung nilai fungsinya masing-masing terhadap titik A, kemudian dimasukkan ke dalam tabel 2.2 *Open list*.

Tabel 2.2 Open list 1

No	Nama Jalur	$f(n) = g(n) + h(n)$
1	AB	$8 + 5 = 13$
2	AC	$5 + 6 = 11$
3	AD	$10 + 6 = 16$

Dari tabel *open list* di atas kita pilih jalur yang mempunyai nilai fungsi $f(n)$ minimal. Pilih AC dan masukkan ke tabel 2.3 *closed list*.

Tabel 2.3 Closed List 2

Nama titik-titik jalur	Total Jarak
A – C	11

4. Titik C berhubungan dengan titik B dan D. Hitung nilai fungsinya masing-masing terhadap titik C, kemudian dimasukkan ke dalam tabel 2.4 *open list*.

Tabel 2.4 Open List 2

No	Nama Jalur	$f(n) = g(n) + h(n)$
1	CB	$10 + 5 = 15$
2	CD	$8 + 6 = 14$

Dari tabel *open list* di atas kita pilih jalur yang mempunyai nilai fungsi $f(n)$ minimal. Kemudian pilih sisi CD dan masukkan ke *closed list* seperti pada tabel 2.5.

Tabel 2.5 Closed List 3

Nama titik-titik jalur	Total Jarak
A – C – D	14

5. Titik D berhubungan dengan titik E dan F. Hitung nilai fungsinya masing-masing terhadap titik D, kemudian dimasukkan ke dalam *open list* sebagaimana pada tabel 2.6.

Tabel 2.6 Open List 3

No	Nama Jalur	$f(n) = g(n) + h(n)$
1	DE	$17 + 3 = 20$
2	DF	$15 + 0 = 15$

Dari tabel *open list* di atas kita pilih jalur yang mempunyai nilai fungsi $f(n)$ minimal. Pilih DF dan masukkan ke *closed list* seperti pada tabel 2.7.

Tabel 2.7 Closed List 4

Nama titik-titik jalur	Total Jarak
A – C – D – F	15

6. Jadi rute terpendek dari titik A ke titik F adalah A – C – D – F dengan total jarak 15 km.

2.3 Kota Malang

Kota Malang berdiri sejak tanggal 1 April 1914. Luas wilayah Kota Malang adalah seluas $\pm 110,06 \text{ km}^2$ dengan ketinggian 440-667m di atas permukaan laut. Kota Malang dialiri oleh 3 sungai besar, yaitu Sungai Brantas, Sungai Amprong

dan Sungai Bango. Kota Malang juga dikelilingi oleh 4 gunung, di utara terdapat Gunung Arjuno, di timur terdapat Gunung Semeru, di selatan terdapat Gunung Kelud dan di barat terdapat Gunung Kawi dan Gunung Panderman.

Kota Malang memiliki lambang yaitu Malangucecwara yang artinya “Tuhan Menghancurkan yang Bathil dan Menegakkan yang Benar”. Lambang ini diusulkan oleh Prof. Dr. R. No. Poerbatjaraka dan erat hubungannya dengan sejarah Kota Malang pada zaman Ken Arok kurang lebih tujuh abad lalu di sekitar candi bernama Malang(kucecwara). Lambang tersebut dipakai sejak tahun berdirinya Kotapraja Malang 1964. Sebelum itu lambang yang digunakan adalah “Malang Namaku, Maju Tujuanku” terjemahan dari Malang *nominor, sursum moveor* yang ditetapkan atau disahkan “bji Gouvernement besluit dd. 25 April 1983, N 027”. Kota Malang juga memiliki simbol yaitu ASOY yang artinya “Arek Malang Selalu Okey”.

Di samping lambang dan slogan Kota Malang juga memiliki cita-cita yang disebut sebagai Tri Bina Cita Kota Malang yaitu Malang sebagai Kota Pendidikan, Malang sebagai Kota Industri, dan Malang sebagai Kota Pariwisata.

1. Kota Pelajar/ Kota Pendidikan

Kota Malang dikenal sebagai Kota Pendidikan, karena memiliki banyak Perguruan Tinggi ternama yang berjumlah 66 perguruan tinggi baik berstatus negeri maupun swasta serta SMK dan SMA bertaraf Nasional dan Internasional.

2. Kota Industri

Kota Malang memiliki pola pertumbuhan industri yang unik, di mana sebagian besar industrinya disokong oleh sektor industri menengah, kecil dan mikro (UMKM). Hal ini menyebabkan banyaknya industri yang tumbuh

berbasis industri rumah dan membuka banyak lapangan kerja. Di samping UMKM, Kota Malang pun menjadi rumah bagi beberapa industri manufaktur besar. Umumnya industri berskala besar yang terdapat di Kota Malang sebagian bergerak di bidang industri manufaktur padat karya.

3. Kota Pariwisata

Kota Malang memiliki potensi wisata yang luar biasa dengan potensi alam yang dimiliki oleh Kota Malang, yaitu pemandangan alam yang elok serta hawa yang sejuk, teduh dan asri serta bangunan-bangunan kuno peninggalan Belanda. Di samping itu tentunya banyak sekali destinasi wisata buatan hasil kreasi masyarakat bersama pemerintah menjadikan Kota Malang layak menjadi tujuan wisata bagi wisatawan dalam maupun luar negeri. *Brand image* “Beautiful Malang” merupakan brand image pariwisata Kota Malang yang dirilis pada tanggal 23 Agustus 2015. Brand ini bertujuan untuk memperkuat *city branding* akan keindahan dan kecantikan Kota Malang. Lebih lanjut, *Beautiful Malang* diharapkan dapat memperkuat *brand image* pariwisata Kota Malang khususnya pada bidang pemasaran jasa dan produk lainnya. Tujuan utamanya adalah untuk meningkatkan ketertarikan wisatawan asing maupun domestik untuk berkunjung ke Kota Malang sebagai Kota Pariwisata (Disporapar, 2019).

2.4 Google Map

Google Maps merupakan layanan dari *google* yang membantu penggunanya untuk melakukan kemampuan pencarian rute atau lokasi yang diinginkan. Sedangkan *Google Maps API* merupakan pengembangan dalam menggabungkan *Google Maps* pada situs *web*. Penggunaan *Google Maps API* memungkinkan untuk

menjadikan situs *Google Maps* ke dalam situs *eksternal* di mana situs tersebut memuat data tertentu yang dapat dilakukan *overlay*.

Walaupun pada awalnya hanya *JavaScript API*, *API Maps*, sejak penggunaannya diperluas untuk menyertakan sebuah API untuk *Adobe Flash* aplikasi, layanan pengambilan gambar peta statis, dan layanan *web* untuk melakukan *geocoding*, petunjuk arah mengemudi, dan memperoleh data profil elevasi.

Kelas kunci dalam perpustakaan *Maps* adalah *Map view*, sebuah *subclass* dari *View group* dalam standar perpustakaan *Android*. Sebuah *Map view* menampilkan peta dengan data yang diperoleh dari layanan *Google Maps*. Apabila *Map view* memiliki fokus, maka dapat dipastikan pengguna dapat menangkap tombol yang ditekan dan gerakan sentuh untuk panel dan *zoom* peta secara otomatis. Hal ini dapat menyediakan semua elemen UI yang diperlukan bagi pengguna untuk mengendalikan peta tersebut. Aplikasi tersebut juga dapat menggunakan metode *Map view* kelas untuk mengontrol *Map view* secara terstruktur, terprogram, dan menarik sejumlah jenis Tampilan di atas peta (TIK, 2017).

Secara sederhana, kelas *Map view* menyediakan pembungkus di *Google Maps API* yang memungkinkan aplikasi tersebut memanipulasi data *Google Maps* dengan metode kelas dan hal itu memungkinkan untuk dikerjakan dengan data *Maps*. Perpustakaan *maps eksternal* bukan bagian dari perpustakaan *Android* standar sehingga tidak mungkin ada pada sejumlah perangkat *Android* biasa. Sama halnya, perpustakaan *Maps* eksternal tidak termasuk dalam perpustakaan *Android standar* yang disediakan dalam sistem SDK. *Google API* hanya menyediakan perpustakaan *Maps* untuk dapat mengembangkan, membangun, dan menjalankan

aplikasi berbasis peta di SDK *Android*, dengan akses penuh ke data *Google Maps*.

2.5 Geographic Information System (GIS)

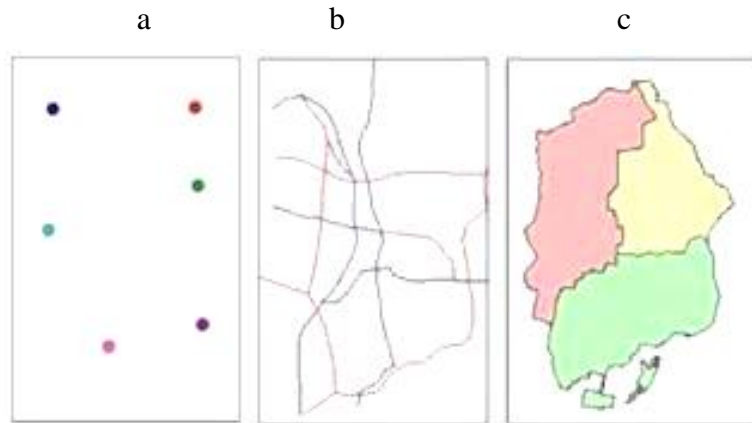
Geographic information system atau disingkat dengan GIS merupakan sebuah sistem berbasis komputer di mana mempunyai kemampuan pemasukan, pengambilan, analisis dan tampilan data geografis yang sangat diutamakan bagi pengambilan sebuah keputusan. SIG merupakan sistem komputer yang terdiri dari perangkat keras, perangkat lunak dan personal (manusia) yang dirancang secara efisien guna memasukkan, menyimpan, memperbaharui, manipulasi, menganalisis dan menyajikan untuk semua informasi yang orientasinya berupa geografis. (Agarwal, 2016)

Data GIS digolongkan menjadi dua macam. Data pertama yaitu data spasial dan data kedua adalah data atribut atau tabular. Secara umum data spasial adalah data yang mengimplementasikan bentuk atau penampakan objek pada permukaan bumi. Gambaran sederhana dari data tabular adalah data deskriptif yang menyajikan nilai dari data grafis tersebut. Berikut penjelasan rinci dua jenis data tersebut:

1. Data spasial

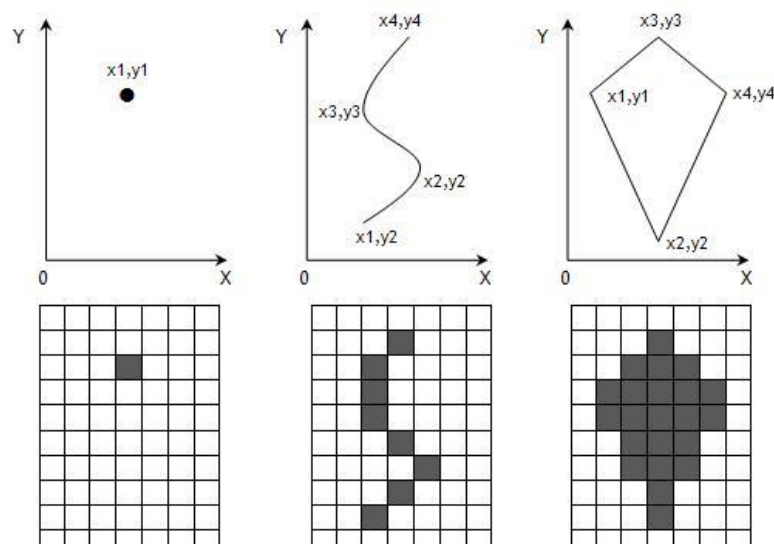
Secara umum data spasial dibagi menjadi dua yaitu data Vektor dan data Raster. Data vektor dibagi lagi menjadi tiga macam data yaitu data titik (*point*), data garis (*line/polyline*), dan data area (*region/polygon*). Data grafis titik umumnya digunakan untuk menggambarkan objek kota, stasiun, curah hujan, alamat customer, dan lain sebagainya. Data Garis dapat diimplementasikan untuk menggambarkan jalan, sungai, jaringan listrik, dan lain sebagainya. Data yang terakhir yaitu data area digunakan untuk menggambarkan suatu batas wilayah

seperti batas wilayah administrasi, batas wilayah penggunaan lahan, batas wilayah kemiringan lereng, dan batas-batas wilayah lainnya. Gambar-gambar di bawah ini memperjelas penggambaran ketiga macam data grafis tersebut.



Gambar 2.4 Gambar Data Grafis Berupa (a) Titik, (b) Garis, dan (c) Area

Pada struktur sebuah data vektor, posisi dari objek ditulis pada sistem koordinat (koordinat kartesius). Sedangkan, objek pada struktur data raster digambarkan pada *grid* dua dimensi yang biasa kita kenal dengan baris dan kolom. Agar memperjelas pemahaman tentang struktur data GIS. Mari kita perhatikan dua di bawah ini yang mengilustrasikan kedua data tersebut.



Gambar 2.5 Contoh Struktur Data GIS, Bagian Atas Struktur Data Vektor, Bagian Bawah Raster

2. Data Atribut

Data atribut atau bisa disebut dengan data tabular merupakan data yang menyimpan semua informasi tentang nilai atau besaran dari sebuah data grafis. Pada struktur data vektor, data atribut tersimpan secara terpisah dalam bentuk tabel. Sedangkan pada struktur data raster nilai data grafisnya tersimpan langsung pada nilai grid atau piksel tersebut. Cara penyimpanan dari data atribut dan koneksi di mana antara data grafis dan atribut di dalam struktur data vektor dan raster disajikan seperti gambar 2.6 .



ID	Negara	Luas_km2	Jml_Penduduk
1	Bulgaria	110.801,500	8.943.258
2	Rumania	236.654,000	23.540.550
3	Serbia	88.201,758	9.979.116

b

Gambar 2. 6 Cara Penyimpanan Data Atribut Pada Struktur Data Vektor, (a) Data Grafis dan (b) Data Atribut

2.6 Python

Python merupakan bahasa pemrograman interpretatif multiguna. Tidak seperti bahasa program lain yang sulit untuk dibaca dan dipahami, python lebih menekankan pada keterbacaan kode agar lebih mudah untuk memahami suatu sintaks. Hal ini menjadikan Python sangat mudah dipelajari baik untuk pemula maupun untuk yang sudah menguasai bahasa pemrograman lain. Bahasa ini muncul

pertama kali pada tahun 1991, dirancang oleh seorang bernama Guido van Rossum. Sampai saat ini Python masih dikembangkan oleh Python Software Foundation. Bahasa Python mendukung hampir semua sistem operasi, bahkan untuk sistem operasi Linux, hampir semua distronya sudah menyertakan Python di dalamnya. Dengan kode yang simpel dan mudah diimplementasikan, seorang programmer dapat lebih mengutamakan pengembangan aplikasi yang dibuat, bukan malah sibuk mencari syntax error.

```
print("Python sangat simpel")
```

Hanya dengan menuliskan kode *print* seperti yang di atas, sudah bisa mencetak apapun yang diinginkan di dalam tanda kurung (). Di bagian akhir kode pun, tidak harus diakhiri dengan tanda *semicolon* (TIK, 2017).

2.6.1 Networkx

Networkx merupakan salah satu library yang dimiliki oleh Python. Library ini memuat beberapa fungsi untuk memvisualisasikan sebuah graf. Adanya library ini membuat seseorang sangat mudah melakukan sebuah pengujian-pengujian permasalahan yang berhubungan dengan graf. Mulai dari permasalahan pewarnaan graf, analisis graf berarah dan tak berarah. Penentuan rute terpendek, pembagian wilayah jaringan, dan lain sebagainya. Dalam pengoperasiannya pun library ini sangat mudah terutama dalam penginstalannya di PC. Library ini juga menyediakan fitur membaca file csv maupun file txt. Fitur ini sangat membantu pengguna ketika memiliki data yang memiliki ekstensi file seperti yang telah disebutkan.

2.7 Kajian Keislaman Tentang Perjalanan (*Safar*)

Safar merupakan kegiatan perjalanan yang dilakukan seseorang dengan tujuan tertentu. Sebagian besar atau bahkan seluruh umat manusia di dunia pernah melakukan hal tersebut. Jarak tempuh yang dilakukan pun beragam, ada yang jauh dan ada pula yang dekat, tentunya dengan intensitas yang berbeda. Kebutuhan seseorang bervariasi, mulai dari kebutuhan yang bersifat duniawi (berwisata) atau bersifat *ukhrawi* (perjalanan religi). Dalam islam hukum asal bepergian ke segala penjuru bumi adalah mubah (diperbolehkan) selama tidak melanggar syariat. Sebagaimana Allah swt berfirman:

هُوَ الَّذِي جَعَلَ لَكُمُ الْأَرْضَ ذَلُولًا فَامْشُوا فِي مَنَاكِبِهَا وَكُلُوا مِنْ رِزْقِهِ ۗ وَإِلَيْهِ

النُّشُورُ

Dialah (Allah) yang menjadikan bumi itu mudah bagimu (untuk ditelusuri), maka berjalanlah (bepergianlah) ke segala penjurunya dan makanlah sebagian dari rezeki-Nya. Dan hanya kepada-Nya-lah kamu (kembali setelah) dibangkitkan. (Q.S Al-Mulk:15)

Imam Ibnu Katsir menjelaskan bahwa kandungan dari ayat ini adalah “Bepergianlah kamu ke segala penjuru bumi sesuai dengan keinginanmu, kemudian telusurilah semua tempat dan pelosoknya untuk berbagai macam usaha dan perniagaanmu. Ketahuilah usaha yang kamu lakukan tidak bermanfaat sedikitpun bagimu, kecuali jika Allah swt memudahkan hal (jalan) bagimu”. Allah swt berfirman bahwa bepergian adalah perintah yang bersifat mubah (dibolehkan). Bentuk dari perintah ini bertujuan untuk memperlihatkan keagungan anugerah-Nya kepada para hamba-Nya. Namun, perjalanan yang hukum asalnya mubah dapat berubah menjadi pundi-pundi pahala atau bisa jadi menyebabkan dosa. Hal tersebut bergantung pada tujuan masing-masing individu dalam melakukan perjalanan.

Sebagaimana Rasulullah saw pernah bersabda:

إِنَّمَا الْأَعْمَالُ بِالنِّيَّاتِ

Sesungguhnya (balasan) amalan-amalan itu tergantung niatnya (HR. Al-Bukhari dan Muslim).

Para Ulama membagi *safar* (perjalanan) menjadi beberapa macam di mana masing-masing memiliki hukum dan konsekuensi. Berdasarkan sifatnya, *safar* dibagi atas tiga jenis:

1. Pertama adalah *safar* taat di mana *safar* tersebut berpahala jika dilakukan dengan niat ikhlas karena Allah swt.
2. Kedua adalah *safar* mubah di mana jika diniatkan untuk memperkokoh ketaatan dan keimanan.
3. Ketiga adalah *safar* maksiat di mana perjalanan tersebut menimbulkan dosa jika dilakukan. Contohnya *safar* bertujuan melakukan pencurian, perampokan atau berbagai perbuatan maksiat lainnya.

Berdasarkan tujuannya *safar* digolongkan menjadi *safar harab* dan *safar thalab*. *Safar harab* adalah *safar* yang memiliki tujuan meninggalkan berbagai keburukan menuju kebaikan), misalnya *safar* dari negeri zalim ke negeri yang damai. Sedangkan pengertian *safar thalab* adalah *safar* dengan tujuan mencari atau menuntut sesuatu, misalnya *safar* ibadah haji atau umrah, *safar* menuntut ilmu, dan lain sebagainya (Hidayati, 2017).

Berdasarkan penjelasan makna perjalanan (*safar*), seyogyanya kita senantiasa berpikir dan bermusyawarah sebelum melakukan perjalanan agar perjalanan kita lebih bermanfaat dan menghasilkan pahala. Hal ini juga selaras dengan kegiatan berwisata. Kaum muslim saat berwisata tidak hanya untuk

kesenangan pribadi, namun kaum muslim yang berwisata harus diniatkan ibadah untuk melihat tanda-tanda kebesaran Allah swt. Niat tersebut akan mengantarkan kepada pintu keberkahan dalam kehidupan. Agar perjalanan yang disertai dapat maksimal, kaum muslim harus mempertimbangkan rute mana yang harus dipilih.

BAB III

METODOLOGI PENELITIAN

3.1. Pendekatan Penelitian

Penelitian ini dilakukan dengan pendekatan studi lapangan dan studi penelitian evaluasi secara kuantitatif. Studi lapangan dilakukan dengan mencari sumber informasi dengan mengamati langsung obyek yang akan diteliti. Studi penelitian evaluasi secara kuantitatif merupakan studi penelitian pada suatu sistem yang dijalankan dengan tujuan melihat sebuah keefektifan sistem tersebut saat digunakan. Kemudian didapatkan sebuah simpulan secara kuantitatif atas pengujian yang telah dilakukan.

3.2. Sumber Data

Data yang digunakan dalam penelitian ini merupakan data sekunder yang terdiri antara dua kategori. Berikut merupakan rincian kategori tersebut:

1. Data Titik Wisata

Data titik wisata ini bersumber dari laman website Disporapar Kota Malang dan buku panduan Pekandayu Kota Malang. Data tersebut berjumlah 43 titik dengan rincian sebagai berikut:

- a. 11 wisata *heritage* (Seni dan Budaya)
- b. 14 wisata taman
- c. 9 Kampung Tematik
- d. 7 wisata religi
- e. 2 wisata air/ hiburan

2. Data Titik Persimpangan Dan Jalan

Data ini bersumber dari laman Dinas Tata Ruang Kota Malang. Dari laman tersebut diperoleh data berupa 236 titik persimpangan pada 624 jalan utama beserta data bobotnya. Jenis jalan utama yang dijadikan data ini adalah jalan kolektor dan jalan arteri (BPPD Kota Malang, 2021).

3.3. Identifikasi Variabel

Variabel yang digunakan dalam penelitian ini adalah variabel bebas dan variabel terikat. Berikut penjelasan variabel tersebut yang disajikan dalam bentuk tabel:

Tabel 3.1 Variabel Penelitian

No	Variabel	Keterangan Variabel	Definisi
1	$V(G)$	Himpunan titik-node $\{x_1, x_2, \dots, x_n\}$	Titik titik yang menyimbolkan tiap destinasi
2	$W_{E(G)}$	Himpunan bobot jarak antar node $\{d(x_1, x_2), \dots, d(x_m, x_n)\}$	Nilai/ jarak sebuah rute dari satu titik dengan titik yang menghubungkannya
3	F	Himpunan pengujian rute $\{f_1(n), \dots, f_k(n)\}$	Nilai pengujian terbaik dari pengujian pada <i>Open List</i> (Himpunan Buka)
4	H	Himpunan nilai heuristik $\{h_1(n), \dots, h_k(n)\}$	Nilai/ Jarak yang sebenarnya dengan node yang menjadi tujuan.
5	G	Himpunan nilai rute sebenarnya $\{g_1(n), \dots, g_k(n)\}$	Nilai/ jarak rute antar node yang terhubung secara langsung

3.4. Tahapan Penelitian

Tahapan penelitian ini terdiri dari enam langkah. Langkah-langkah tersebut adalah sebagai berikut:

a) Peninjauan Langsung

Langkah pertama adalah mengamati langsung obyek yang akan menjadi bahan penelitian. Pada penelitian ini ada beberapa aspek yang akan ditinjau, mulai dari titik lokasi yang akan dijadikan sebuah node graf, rute yang akan dijadikan sebagai edge pada graf, dan juga mengamati apakah *edge* tersebut satu arah atau dua arah. Hal itu akan menjadi pertimbangan dalam mengumpulkan data yang dibutuhkan.

b) Pengumpulan Data

Tahap ini merupakan tahap setelah peninjauan langsung objek penelitian. Pengumpulan data tersebut berupa kumpulan nama destinasi wisata di Kota Malang dan kumpulan rute yang akan dijadikan *edge* dari sebuah graf di mana *edge* ini juga sudah diseleksi baik yang satu arah maupun dua arah. Pengumpulan data berupa rute ini hanya akan diambil dari rute jalan utama yang meliputi jalan kolektor dan jalan arteri yang bersumber dari website peta Dispetarung (Dinas Penataan Tata Ruang) Kota Malang. Hal ini dilakukan karena jalan utama sangat memungkinkan untuk dilewati segala jenis kendaraan. Proses pengumpulan data terakhir adalah mengumpulkan data jarak rute antar node yang akan dijadikan bobot *edge* pada graf.

c) Penentuan Titik Koordinat Destinasi Wisata

Penentuan titik dari setiap node adalah penentuan titik lokasi koordinat destinasi wisata dan titik persimpangan jalan yang dilakukan dengan mengambil

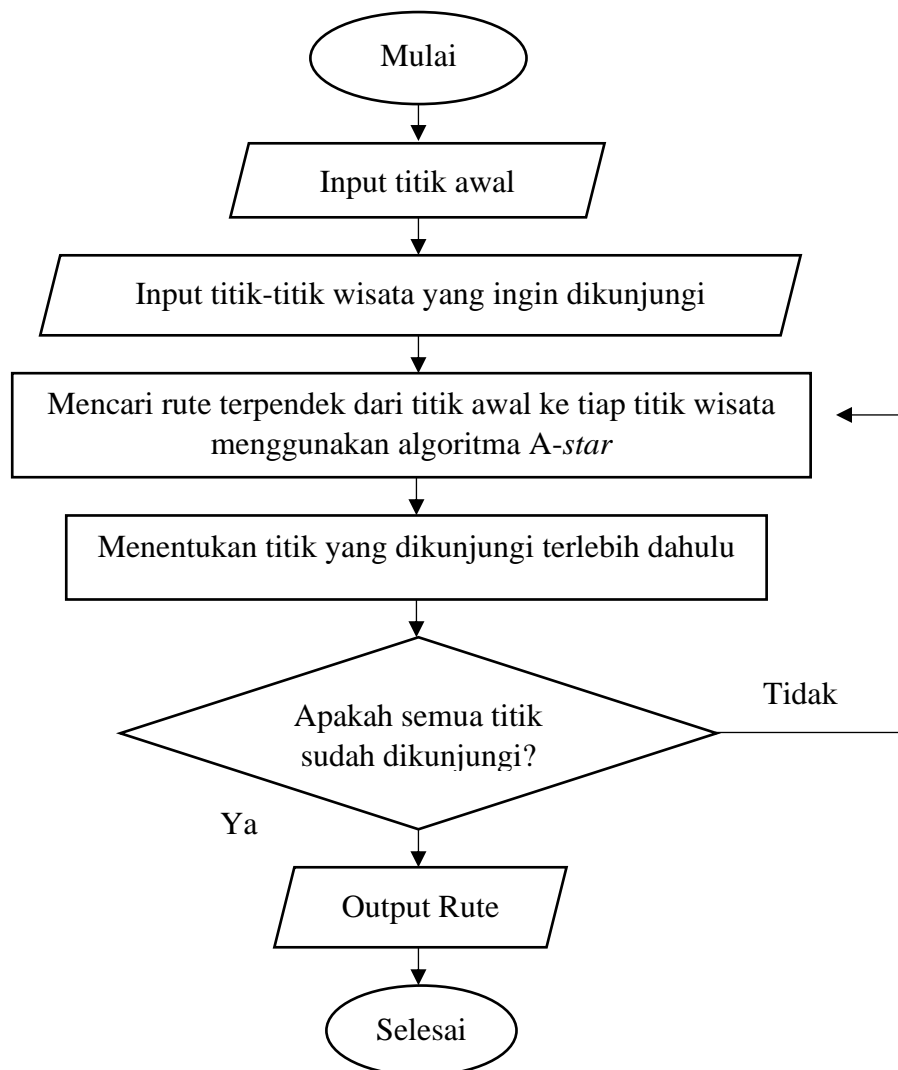
titik longitude dan latitude pada *Google Maps*. Hasil dari penentuan tersebut akan dijadikan sebagai acuan menghitung nilai heuristik yang dibutuhkan dalam proses pencarian rute terpendek menggunakan algoritma *A-star*.

d) Membuat Graf

Proses pembuatan graf dilakukan dengan menggambar seluruh data yang diperoleh, dimulai dari semua titik dan semua sisi yang menghubungkan antar titik satu dengan titik lainnya. Sisi yang sudah tergambar akan diberi label bobot jarak. Sistem label tersebut juga menggunakan alat bantu google maps. Satuan jarak yang digunakan adalah satuan km (kilometer). Pemilihan satuan tersebut karena dinilai orang lebih familiar ketika akan membahas jarak rute untuk mengunjungi beberapa tempat.

e) Melakukan Eksperimen Untuk Menentukan Rute Terpendek

Pangkalan data yang sudah diubah menjadi graf melalui program python, maka langkah selanjutnya adalah melakukan pengujian rute terpendek. Dalam satu pengujian akan dipilih beberapa lokasi titik destinasi wisata yang akan dikunjungi, kemudian ditentukan destinasi mana yang akan menjadi rute awal sampai akhir. Proses pengujian dilakukan beberapa kali sehingga bisa menjangkau pengujian seluruh destinasi wisata di Kota Malang. Penentuan rute dalam pengujian ini adalah algoritma *A-star* dengan penambahan seleksi destinasi sebagai pengembangan dari penelitian. Berikut merupakan gambaran *flowchart* pengujian rute dalam penelitian rute terpendek. Hal ini akan disajikan dalam gambar 3.1 tentang *flowchart* alur penelitian:



Gambar 3.1 Flowchart Alur Pengujian Route

f) Evaluasi Hasil

Tahap terakhir dari penelitian ini adalah evaluasi hasil dari rute terpendek yang dihasilkan. Evaluasi akan mengkaji tentang seberapa efektifkah hasil rute yang dihasilkan. Faktor yang dijadikan sebagai penentu keefektifan adalah bergantung bagaimana penentuan nilai heuristik yang diambil. Selain itu yang akan dijadikan keefektifan adalah jarak yang akan ditempuh untuk berkunjung ke beberapa tempat destinasi wisata.

BAB IV

PEMBAHASAN

4.1 Deskripsi Data

Data pada penelitian ini terbagi menjadi dua kategori yaitu data titik destinasi wisata dan titik persimpangan serta jalan. Berikut merupakan deskripsi dari masing-masing kategori:

1. Data Titik Destinasi Wisata

Data titik lokasi wisata terdiri dari beberapa jenis destinasi, berikut tabel penjabaran data tersebut:

a. Destinasi Wisata Sejarah

Tabel 4.1 Kode Wisata Sejarah

No	Nama Destinasi	Simbol
1	Museum Mpu Purwa	W4
2	Indonesian Old Cinema Museum	W6
3	Monumen Pahlawan Trip	W8
4	Museum Pendidikan Indonesia	W19
5	Wisma Tumapel	W23
6	Patung Chairil Anwar	W24
7	Toko Es Krim & Roti Oen	W24
8	Pasar Besar	W28
9	Museum Musik Indonesia	W32
10	Patung Hamid Rusdi	W35
11	Museum Brawijaya	W37

b. Destinasi Hiburan

Tabel 4.2 Kode Wisata Hiburan

No	Nama Destinasi	Simbol
1	Tlogomas Park	W1
2	Hawaii Water Park	W11

4. Data Titik Koordinat Destinasi Taman Kota

Tabel 4.3 Kode Wisata Taman Kota

No	Nama Destinasi	Simbol
1	Taman Singha Merjosari	W3
2	Taman Krida	W5
3	Taman Kunang-kunang	W7
4	Taman kenededes	W10
5	Taman Dieng	W16
6	Taman Edukasi Rolak	W18
7	Taman Bentoel Trunojoyo	W22
8	Alun-Alun Merdeka	W26
9	Taman Cerme	W30
10	Taman Slamet	W31
11	Taman Merbabu	W35
12	Taman Dempo	W36
13	Taman Ijen	W37
14	Alu-Alun Tugu	W38

5. Data Titik Koordinat Destinasi Religi

Tabel 4.4 Kode Wisata Religi

No	Nama Destinasi	Simbol
1	Gereja Cathedral Ijen	W8
2	Vihara Vajra Bumi Kertanegara	W16
3	Masjid Ki Ageng Gribig	W17
4	Kelenteng Eng An Kiong	W20
5	Gereja Hati Kudus Yesus Kayutangan	W24
6	Gereja GPIB Immanuel	W25
7	Masjid Agung Jami' Malang	W26

6. Destinasi Kampung Tematik

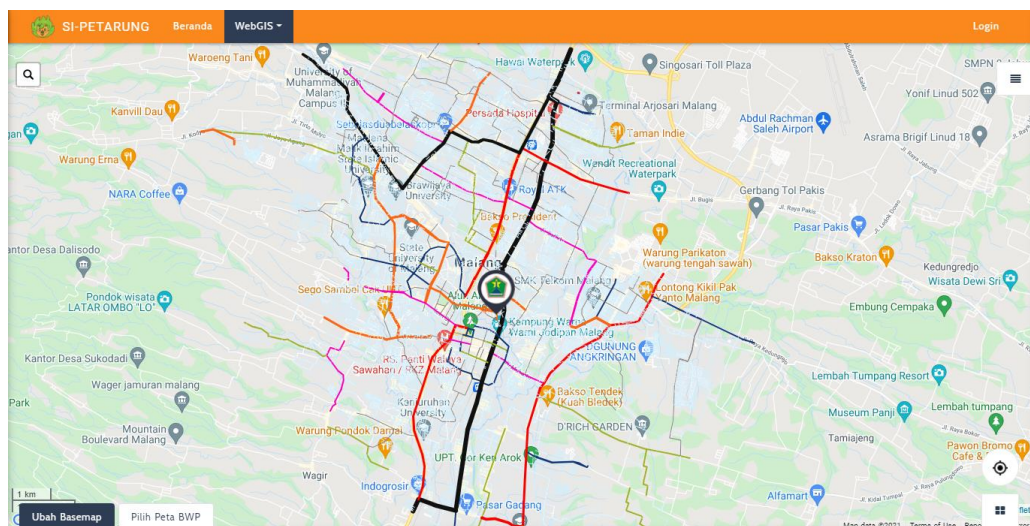
Tabel 4.5 Kode Wisata Kampung Tematik

No	Nama Destinasi	Simbol
1	Kampung Wisata Keramik Dinoyo	W2
2	Kampung Budaya Polowijen	W9
3	Kampung Glintung Go Green (3G)	W12
4	Kampung Keripik Tempe Sanan	W13
5	Kampung Warna Warni Jodipan	W15
6	Kampung Tiga Dimensi (3D) Tridi	W15
7	Kampung Biru Arema	W21
8	Kampung Heritage Kayutangan	W27
9	Kampung Putih	W29

Simbol huruf **W** menyimbolkan bahwa titik tersebut adalah titik wisata. Hal ini dilakukan untuk pembeda dengan titik-titik persimpangan yang akan digunakan. Simbol tersebut akan digunakan agar pemilihan lokasi wisata yang diinginkan lebih mudah.

2. Data Titik Persimpangan Jalan

Data titik persimpangan jalan diambil dari jenis jalan kolektor dan jalan arteri. Pemilihan dua jenis jalan tersebut karena memungkinkan untuk berbagai jenis kendaraan. Selain itu pemilihan dua jenis jalan itu juga berlandaskan fokus penelitian ini pada jalan utama. Gambaran jalan yang dijadikan acuan pengambilan data tersaji dalam gambar 4.1.



Gambar 4.1 Peta Jalan Kota Malang

Berdasarkan gambar tersebut, ada berbagai warna yang mendeskripsikan dua jenis jalan tersebut. Jalan berwarna hitam merupakan jalan kolektor dan arteri primer. Jalan berwarna merah merupakan jalan arteri sekunder I, jalan berwarna *orange* menyimbolkan jalan arteri sekunder II, serta jalan berwarna ungu menggambarkan jalan arteri sekunder III. Jalan kolektor sekunder I digambarkan dengan warna biru gelap, sedangkan Jalan sekunder II digambarkan dengan warna hijau. Jalan yang diberi warna biru laut dan abu-abu

merupakan jalan lingkungan dan lokal, di mana jalan tersebut tidak digunakan karena merupakan jalan sempit dan kurang efektif jika dilalui berbagai jenis macam kendaraan.

Setelah dilakukan seleksi data dari laman Dinas Tata Ruang Kota Malang didapatkan hasil berupa 236 titik. Setiap titik disimbolkan dengan kode kecamatan dan nomor identitas titik. Tabel 4.6 akan menampilkan rincian data titik persimpangan per kecamatan:

Tabel 4.6 Tabel Kode Kecamatan

No	Nama Kecamatan	Kode	Jumlah
1	Lowokwaru	L	57
2	Sukun	S	15
3	Klojen	K	73
4	Blimbing	B	37
5	Kedungkandang	D	23

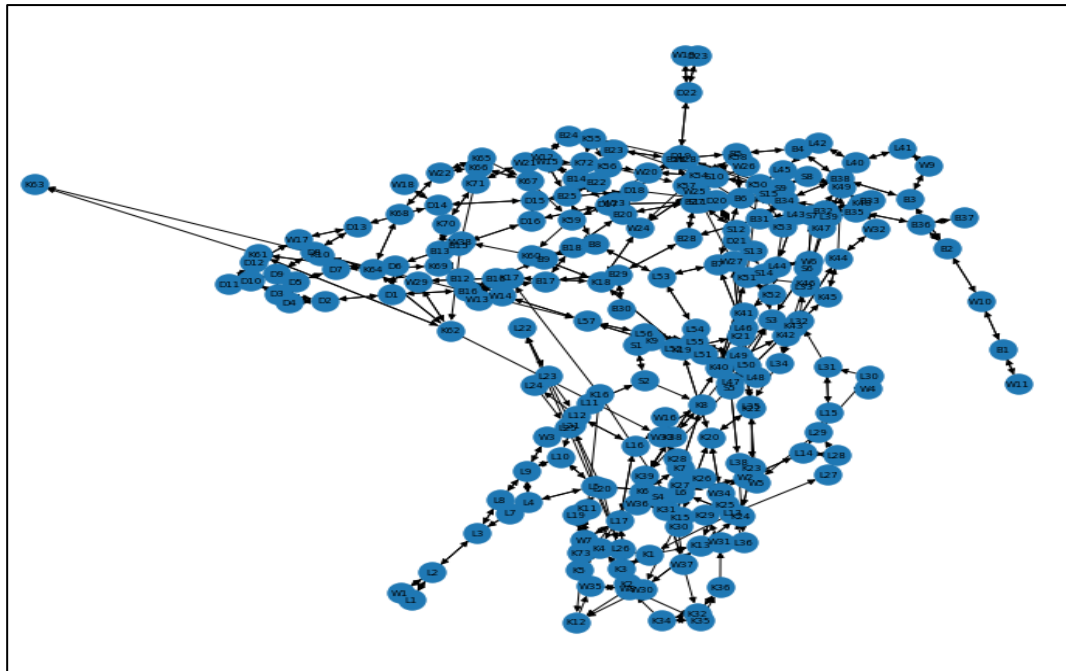
Titik-titik tersebut kemudian dirangkai menjadi sebuah graf berarah yang di sesuaikan dengan peta jalan Kota Malang. Pembentukan data graf berarah tersebut dibuat pada ekstensi *file* berformat *txt* dan menghasilkan 620 jalan. Tabel pembentukan data graf berarah tersaji dalam tabel 4.7.

Tabel 4.7 Tabel Format Penulisan Data

Titik 1	Titik 2	Bobot (km)
L1	L2	0.5

Konsep pendeskripsian data tersebut memiliki arti titik L1 terhubung dengan L2 dan memiliki arah $L1 \rightarrow L2$ dengan bobot 0.5 km

Tahap selanjutnya adalah memvisualisasikan seluruh data melalui program Python dengan *library networkx*. Hasil dari visualisasi tersebut tersaji dalam gambar 4.2 .



Gambar 4.2 Visualisasi Graf Kota Malang

4.2 Pengujian Pencarian Rute Terpendek

Pengujian rute terpendek akan dilakukan sesuai dengan algoritma tahapan penelitian yang telah dijelaskan pada bab 3. Berikut merupakan alur penelitian pengujian rute:

a. Memilih Titik Awal dan Titik Tujuan

Tahap awal pada penelitian ini adalah menentukan titik awal perjalanan. Titik awal ini dipilih bebas sesuai dengan yang diinginkan. Namun, pemilihan titik awal harus memerhatikan kode tiap kecamatan dan juga batas titik pada setiap

kecamatan. Batas pemilihan titik dapat dilihat dari jumlah titik pada deskripsi data yang telah dipaparkan sebelumnya.

Pemilihan lokasi destinasi wisata dapat dilakukan untuk beberapa lokasi wisata, asalkan tidak memilih lebih dari 43 destinasi wisata. Hal ini didasarkan data bahwa lokasi destinasi wisata yang tersedia hanya berjumlah 43 destinasi. Format untuk pengisian beberapa titik wisata adalah dipisahkan dengan tanda koma (,). Bentuk pemilihan titik awal dan beberapa lokasi tujuan tersaji dalam tabel 4.8.

Tabel 4.8 Tabel Contoh Pengujian Rute

Titik Awal	L1
Titik-titik Tujuan Destinasi	W3,W1,W2

b. Penentuan Rute

Penentuan rute akan menyeleksi titik awal terhadap setiap lokasi titik tujuan menggunakan algoritma *A-star*. Berikut merupakan penjelasan seleksi penentuan rute tersebut berdasarkan contoh pada poin **a**.

- **Penentuan Rute L1 → W3**

L1 merupakan titik awal sehingga L1 kita masukkan ke dalam *List Tutup*

$$List\ Tutup = [[L1], 0]$$

L1 hanya terhubung dengan L2 sehingga L2 langsung kita masukkan ke *list tutup* menjadi:

$$List\ Tutup = [[L1, L2], 1.5]$$

Selanjutnya akan dicari titik mana saja yang terhubung dengan L2 dan berapa nilai masing-masing heuristik titik tersebut. Nilai heuristik diambil

melalui perhitungan jarak *Euclid* pada persamaan (2). Perhitungan ini diambil dari titik koordinat yang ada pada gambar visualisasi graf. Kemudian didapatkan hasil berupa panjang rute dan nilai heuristiknya sebagai berikut:

$$\text{Terpilih} = \{ 'W1': (0.15, 1.7), 'L3': (0.6, 0.7) \}$$

Kemudian kita seleksi melalui sebuah perhitungan dan dipilih nilai yang paling minimum.

$$\begin{aligned} f(W1) &= g(W1) + h(W1) \\ &= (1.5 + 0.15) + 1.7 \\ &= 3.35 \end{aligned}$$

$$\begin{aligned} f(L3) &= g(L3) + h(L3) \\ &= (1.5 + 0.6) + 0.7 \\ &= 2.8 \end{aligned}$$

Hasil yang diperoleh menunjukkan bahwa L3 memiliki perhitungan yang minimum, sehingga kita masukkan ke dalam *list* tutup menjadi:

$$\text{List Tutup} = [['L1', 'L2', 'L3'], 2.1]$$

Sama halnya dengan proses sebelumnya, akan dicari titik yang terhubung dengan L3, kemudian didapatkan hasil:

$$\text{Terpilih} = \{ 'L4': (0.5, 0.5), 'L8': (0.9, 0.4) \}$$

$$\begin{aligned} f(L4) &= g(L4) + h(L4) \\ &= (2.1 + 0.5) + 0.5 \\ &= 3.1 \end{aligned}$$

$$f(L8) = g(L8) + h(L8)$$

$$= (2.1 + 0.9) + 0.4$$

$$= 3.4$$

Melalui proses yang sama didapatkan hasil bahwa titik yang memiliki perhitungan minimum adalah titik L4. L4 kita masukkan dalam *list* tutup.

$$\text{List Tutup} = [[\text{'L1'}, \text{'L2'}, \text{'L3'}, \text{'L4'}], 2.6]$$

L4 terhubung dengan L9 dan L5. Panjang rute dan nilai heuristiknya adalah:

$$\text{Terpilih} = \{\text{'L9': (0.6, 0.1)}, \text{'L5': (0.5, 0.4)}\}$$

$$f(L9) = g(L9) + h(L9)$$

$$= (2.6 + 0.6) + 0.1$$

$$= 3.3$$

$$f(L5) = g(L5) + h(L5)$$

$$= (2.6 + 0.5) + 0.4$$

$$= 3.5$$

Melalui penyeleksian yang sama pula didapatkan hasil bahwa titik yang memiliki perhitungan minimum adalah titik L9. L9 kita masukkan dalam *list* tutup.

$$\text{List Tutup} = [[\text{'L1'}, \text{'L2'}, \text{'L3'}, \text{'L4'}, \text{'L9'}], 3.2]$$

L9 terhubung dengan L8, W3, dan L11. Kemudian kita ketahui informasi di dalamnya tentang bobot dan nilai heuristiknya adalah:

$$\text{Terpilih} = \{\text{'L8': (0.4, 0.3)}, \text{'L11': (0.7, 0.5)}, \text{'W3': (0.1, 0)}\}$$

$$f(L8) = g(L8) + h(L8)$$

$$= (3.2 + 0.4) + 0.3$$

$$= 3.9$$

$$f(L11) = g(L11) + h(L11)$$

$$= (3.2 + 0.7) + 0.5$$

$$= 4.4$$

$$f(W3) = g(W3) + h(W3)$$

$$= (3.2 + 0.1) + 0$$

$$= 3.3$$

W3 terpilih karena memiliki nilai perhitungan yang paling minum dan juga memiliki nilai heuristik 0 yang menandakan telah sampai pada lokasi tujuan. Sehingga didapatkan hasil:

$$List\ Tutup = [['L1', 'L2', 'L3', 'L4', 'L9', 'W2'], 3.3]$$

- **Penentuan Rute L1 → W1**

Proses mekanisme pencarian pada kali ini sama halnya dengan sebelumnya. Titik L1 hanya terhubung dengan L2. Sehingga didapatkan:

$$List\ Tutup = [['L1', 'L2'], 1.5]$$

L2 terhubung dengan dua titik yaitu L3 dan W2. Kedua titik tersebut mempunyai informasi bobot dan nilai heuristiknya terhadap W2.

$$Terpilih = \{ 'L3': (0.6, 0.4), 'W1': (0.15, 0) \}$$

$$f(L3) = g(L3) + h(L3)$$

$$= (1.5 + 0.6) + 0.4$$

$$= 2.5$$

$$f(W1) = g(W1) + h(W1)$$

$$= (1.5 + 0.15) + 0$$

$$= 1.65$$

W1 dipilih karena memiliki nilai perhitungan minimum. Selain itu W1 sudah memiliki nilai heuristik sebesar 0 di mana hal tersebut menandakan sudah sampai pada lokasi titik tujuan. Sehingga didapatkan hasil:

$$List\ Tutup = [['L1', 'L2', 'W1'], 1.65]$$

- **Penentuan Rute L1 → W2**

Sebagai titik awal, L1 kita masukkan ke dalam *list* tutup, sehingga dapat dituliskan sebagai

$$List\ Tutup = [['L1'], 0]$$

L1 terhubung dengan titik L2 sehingga L2 langsung kita masukkan ke *list* tutup. Selanjutnya kita tuliskan:

$$List\ Tutup = [['L1', 'L2'], 1.5]$$

Kemudian akan dicari titik mana saja yang terhubung dengan L2 dan berapa nilai masing-masing heuristik titik tersebut. Kemudian didapatkan hasil berupa panjang rute dan nilai heuristiknya sebagai berikut:

$$Terpilih = \{ 'W1': (0.15, 1.1), 'L3': (0.6, 0.5) \}$$

Kemudian kita seleksi melalui sebuah perhitungan dan dipilih nilai yang paling minimum.

$$f(W1) = g(W1) + h(W1)$$

$$= (1.5 + 0.15) + 1.1$$

$$= 2.75$$

$$\begin{aligned}
 f(L3) &= g(L3) + h(L3) \\
 &= (1.5 + 0.6) + 0.5 \\
 &= 2.6
 \end{aligned}$$

Hasil yang diperoleh menunjukkan bahwa L3 memiliki perhitungan yang minimum, sehingga kita masukkan ke dalam *list* tutup menjadi:

$$List\ Tutup = [[L1', L2', L3'], 2.1]$$

Sama halnya dengan proses sebelumnya, akan dicari titik yang terhubung dengan L3, kemudian didapatkan hasil:

$$Terpilih = \{L4': (0.5, 0.3), L8': (0.9, 1.7)\}$$

$$\begin{aligned}
 f(L4) &= g(L4) + h(L4) \\
 &= (2.1 + 0.5) + 0.3 \\
 &= 2.9
 \end{aligned}$$

$$\begin{aligned}
 f(L8) &= g(L8) + h(L8) \\
 &= (2.1 + 0.9) + 1.7 \\
 &= 4.7
 \end{aligned}$$

Melalui proses yang sama didapatkan hasil bahwa titik yang memiliki perhitungan minimum adalah titik L4. L4 kita masukkan dalam *list* tutup.

$$List\ Tutup = [[L1', L2', L3', L4'], 2.6]$$

L4 terhubung dengan titik L5 dan L9. Kedua titik tersebut mengandung informasi sebagai berikut:

$$Terpilih = \{L5': (0.4, 0.3), L9': (0.6, 1.5)\}$$

$$f(L5) = g(L5) + h(L5)$$

$$= (2.6 + 0.4) + 0.3$$

$$= 3.3$$

$$f(L8) = g(L8) + h(L8)$$

$$= (2.6 + 0.6) + 1.7$$

$$= 4.9$$

Berdasarkan proses perhitungan di atas, L5 memiliki hasil perhitungan minimum, sehingga L5 dapat dimasukkan dalam *list* tutup.

$$List\ Tutup = [[L1', L2', L3', L4', L5'], 3.0]$$

Titik L5 terhubung dengan titik L6 dan L10. Kedua titik tersebut memiliki informasi sebagai berikut

$$Terpilih = \{L6': (0.02, 0.3), L10': (0.6, 0.7)\}$$

$$f(L6) = g(L6) + h(L6)$$

$$= (3.0 + 0.02) + 0.3$$

$$= 3.32$$

$$f(L10) = g(L10) + h(L10)$$

$$= (3.0 + 0.6) + 0.7$$

$$= 4.3$$

L6 memiliki proses perhitungan paling minimum, sehingga L6 dapat dimasukkan dalam *list* tutup.

$$List\ Tutup = [[L1', L2', L3', L4', L5', L6'], 3.02]$$

L6 terhubung dengan titik L13 dan W2. Kedua titik tersebut memuat informasi sebagai berikut:

$$Terpilih = \{L13': (0.8, 1.3), W2': (0.25, 0)\}$$

$$\begin{aligned}
 f(L13) &= g(L13) + h(L13) \\
 &= (3.02 + 0.8) + 1.3 \\
 &= 5.12
 \end{aligned}$$

$$\begin{aligned}
 f(W2) &= g(W2) + h(W2) \\
 &= (3.02 + 0.25) + 0 \\
 &= 3.27
 \end{aligned}$$

W2 memiliki nilai heuristik 0 dan memiliki perhitungan paling minimum, sehingga W2 dapat kita masukkan ke dalam *list* tutup sebagai berikut:

$$List\ Tutup = [['L1', 'L2', 'L3', 'L4', 'L5', 'L6', 'W2'], 3.27]$$

Berdasarkan perhitungan di atas antara titik awal yaitu L1 terhadap masing-masing titik didapatkan bahwa destinasi terdekat untuk dikunjungi adalah W1. Sehingga W1 dapat kita masukkan dalam *list* rute.

$$List\ Rute = [['L1', 'L2', 'W1'], 1.65]$$

Langkah selanjutnya adalah menyeleksi dari titik W1 ke titik W2 dan W3. Akan ditentukan destinasi mana yang memiliki rute terpendek untuk dikunjungi terlebih dahulu.

• W1 → W3

W1 hanya terhubung dengan titik L2. Informasi tersebut menjadikan titik L2 otomatis masuk ke dalam *list* tutup.

$$List\ Tutup = [['W1', 'L2'], 0.15]$$

L2 terhubung dengan titik L1 dan L3. Kedua titik tersebut mengandung informasi sebagai berikut:

Terpilih = {'L1': (1.5, 2.2), 'L3': (0.6, 0.7)}

$$\begin{aligned} f(L1) &= g(L1) + h(L1) \\ &= (0.15 + 1.5) + 2.2 \\ &= 3.85 \end{aligned}$$

$$\begin{aligned} f(L3) &= g(L3) + h(L3) \\ &= (0.15 + 0.6) + 0.7 \\ &= 1.45 \end{aligned}$$

L3 memiliki nilai perhitungan paling minimum. Sehingga L3 dapat dimasukkan dalam *list* tutup.

List Tutup = [['W1', 'L2', 'L3'], 0.75]

L3 terkoneksi dengan titik L8 dan L4. Kedua titik tersebut memuat beberapa informasi sebagai berikut:

Terpilih = {'L8': (0.8, 0.6), 'L4': (0.5, 0.7)}

$$\begin{aligned} f(L8) &= g(L8) + h(L8) \\ &= (0.75 + 0.8) + 0.6 \\ &= 2.15 \end{aligned}$$

$$\begin{aligned} f(L4) &= g(L4) + h(L4) \\ &= (0.75 + 0.5) + 0.7 \\ &= 1.95 \end{aligned}$$

L4 memiliki perhitungan paling kecil. Selanjutnya L4 dapat dimasukkan kedalam *list* tutup menjadi:

List Tutup = [['W1', 'L2', 'L3', 'L4'], 1.25]

Terdapat titik L9 dan L5 yang terhubung dengan L4. Kedua titik tersebut memuat informasi sebagai berikut:

Terpilih = {'L9': (0.6, 0.1), 'L5': (0.4, 0.7)}

$$\begin{aligned} f(L9) &= g(L9) + h(L9) \\ &= (1.25 + 0.6) + 0.1 \\ &= 1.95 \end{aligned}$$

$$\begin{aligned} f(L5) &= g(L5) + h(L5) \\ &= (1.25 + 0.6) + 0.7 \\ &= 2.55 \end{aligned}$$

L9 memiliki proses perhitungan minimum, Sehingga L9 dapat dimasukkan ke dalam *list* tutup sebagai berikut:

List Tutup = [['W1', 'L2', 'L3', 'L4', 'L9'], 1.85]

L9 terhubung dengan L8, W3, dan L11. Kemudian ketiganya mengandung informasi tentang bobot dan nilai heuristiknya adalah:

Terpilih = {'L8': (0.4, 0.3), 'L11': (0.7, 0.5), 'W3': (0.1, 0)}

$$\begin{aligned} f(L8) &= g(L8) + h(L8) \\ &= (1.85 + 0.4) + 0.3 \\ &= 2.55 \end{aligned}$$

$$\begin{aligned} f(L10) &= g(L10) + h(L10) \\ &= (1.85 + 0.7) + 0.5 \\ &= 3.05 \end{aligned}$$

$$\begin{aligned} f(W3) &= g(W3) + h(W3) \\ &= (1.85 + 0.1) + 0 \\ &= 1.95 \end{aligned}$$

W3 memiliki proses perhitungan minimum dan memiliki nilai heuristik sama dengan nol. Hal ini menandakan bahwa proses pencarian destinasi telah sampai. Sehingga W3 dapat kita masukkan dalam *list* tutup sebagai berikut:

$$List\ Tutup = [['W1', 'L2', 'L3', 'L4', 'L9', 'W3'], 1.95]$$

• **W1 → W2**

W1 hanya terhubung dengan titik L2. Informasi tersebut menjadikan titik L2 otomatis masuk ke dalam *list* tutup.

$$List\ Tutup = [['W1', 'L2'], 0.15]$$

Kemudian L2 terhubung dengan titik L1 dan L3. Keduanya mengandung informasi berupa panjang rute dan nilai heuristiknya sebagai berikut:

$$Terpilih = \{ 'L1': (1.5, 2.1), 'L3': (0.6, 0.5) \}$$

Kemudian kita seleksi melalui sebuah perhitungan dan dipilih nilai yang paling minimum.

$$\begin{aligned} f(L1) &= g(L1) + h(L1) \\ &= (0.15 + 1.5) + 2.1 \\ &= 3.75 \end{aligned}$$

$$\begin{aligned} f(L3) &= g(L3) + h(L3) \\ &= (0.15 + 0.6) + 0.5 \\ &= 1.25 \end{aligned}$$

Hasil yang diperoleh menunjukkan bahwa L3 memiliki perhitungan yang minimum, sehingga kita masukkan ke dalam *list* tutup menjadi:

$$List\ Tutup = [['W1', 'L2', 'L3'], 0.75]$$

Sama halnya dengan proses sebelumnya, akan dicari titik yang terhubung dengan L3, kemudian didapatkan hasil:

$$\text{Terpilih} = \{ 'L4': (0.5, 0.3), 'L8': (0.9, 1.7) \}$$

$$\begin{aligned} f(L4) &= g(L4) + h(L4) \\ &= (0.75 + 0.5) + 0.3 \\ &= 1.55 \end{aligned}$$

$$\begin{aligned} f(L8) &= g(L8) + h(L8) \\ &= (0.75 + 0.9) + 1.7 \\ &= 3.35 \end{aligned}$$

Melalui proses yang sama didapatkan hasil bahwa titik yang memiliki perhitungan minimum adalah titik L4. L4 kita masukkan dalam *list* tutup.

$$\text{List Tutup} = [['W1', 'L2', 'L3', 'L4'], 1.25]$$

L4 terhubung dengan titik L5 dan L9. Kedua titik tersebut mengandung informasi sebagai berikut:

$$\text{Terpilih} = \{ 'L5': (0.4, 0.3), 'L9': (0.6, 1.5) \}$$

$$\begin{aligned} f(L5) &= g(L5) + h(L5) \\ &= (1.25 + 0.4) + 0.3 \\ &= 1.95 \end{aligned}$$

$$\begin{aligned} f(L9) &= g(L9) + h(L9) \\ &= (1.25 + 0.6) + 1.7 \\ &= 2.55 \end{aligned}$$

Berdasarkan proses perhitungan di atas, L5 memiliki hasil perhitungan minimum, sehingga L5 dapat dimasukkan dalam *list* tutup.

$$\text{List Tutup} = [['W1', 'L2', 'L3', 'L4', 'L5'], 1.65]$$

Titik L5 terhubung dengan titik L6 dan L10. Kedua titik tersebut memiliki informasi sebagai berikut

$$\text{Terpilih} = \{ 'L6': (0.02, 0.3), 'L10': (0.6, 0.7) \}$$

$$\begin{aligned} f(L6) &= g(L6) + h(L6) \\ &= (1.65 + 0.02) + 0.3 \\ &= 1.97 \end{aligned}$$

$$\begin{aligned} f(L10) &= g(L10) + h(L10) \\ &= (1.65 + 0.6) + 0.7 \\ &= 2.95 \end{aligned}$$

L6 memiliki proses perhitungan paling minimum, sehingga L6 dapat dimasukkan dalam *list* tutup.

$$\text{List Tutup} = [['W1', 'L2', 'L3', 'L4', 'L5', 'L6'], 1.67]$$

L6 terhubung dengan titik L13 dan W2. Kedua titik tersebut memuat informasi sebagai berikut:

$$\text{Terpilih} = \{ 'L13': (0.8, 1.3), 'W2': (0.25, 0) \}$$

$$\begin{aligned} f(L13) &= g(L13) + h(L13) \\ &= (1.67 + 0.8) + 1.3 \\ &= 2.77 \end{aligned}$$

$$\begin{aligned} f(W2) &= g(W2) + h(W2) \\ &= (1.67 + 0.25) + 0 \end{aligned}$$

$$= 1.92$$

W2 memiliki nilai heuristik 0 dan memiliki perhitungan paling minimum, sehingga W2 dapat kita masukkan ke dalam *list* tutup sebagai berikut:

$$List\ Tutup = [['L1', 'L2', 'L3', 'L4', 'L5', 'L6', 'W2'], 1.92]$$

Berdasarkan perhitungan di atas, dari titik W1, rute terdekat yang dapat dikunjungi terlebih dahulu adalah destinasi wisata W2. Sehingga kita dapat masukkan W2 ke dalam *list* rute

$$List\ Rute = [['L1', 'L2', 'W1'], ['W1', 'L2', 'L3', 'L4', 'L5', 'L6', 'W2'], 3.57]$$

Destinasi yang tersisa adalah destinasi W3. Pencarian rute menuju W3 akan dimulai dari titik W2. Di sini tidak akan dilakukan perbandingan kembali karena W3 menjadi destinasi terakhir. W2 terhubung dengan titik L14 dan L6. Kedua titik tersebut memuat informasi sebagai berikut:

$$Terpilih = \{ 'L14': (0.3, 1.8), 'L6': (0.25, 1) \}$$

$$f(L14) = g(L14) + h(L14)$$

$$= (0 + 0.3) + 1.8$$

$$= 2.2$$

$$f(L6) = g(L6) + h(L6)$$

$$= (0 + 0.25) + 1$$

$$= 1.25$$

L6 memiliki nilai perhitungan minimum sehingga dapat dimasukkan ke dalam *list* tutup. Hal ini dapat dituliskan sebagai:

$$List\ Tutup = [['W2', 'L6'], 0.25]$$

L6 memiliki keterhubungan dengan titik L5 dan L13. Berdasarkan kedua titik tersebut, terdapat informasi:

$$\text{Terpilih} = \{ 'L5': (0.05, 0.6), 'L13': (0.8, 1.2) \}$$

$$\begin{aligned} f(L13) &= g(L13) + h(L13) \\ &= (0.25 + 0.8) + 1.2 \\ &= 2.25 \end{aligned}$$

$$\begin{aligned} f(L5) &= g(L5) + h(L5) \\ &= (0.25 + 0.05) + 0.6 \\ &= 0.9 \end{aligned}$$

L6 memiliki nilai perhitungan minimum sehingga dapat dimasukkan ke dalam *list* tutup. Hal ini dapat dituliskan sebagai:

$$\text{List Tutup} = [['W2', 'L6', 'L5'], 0.3]$$

L5 terhubung dengan L4 dan L10. Kedua titik tersebut masing-masing memiliki informasi bobot dan nilai heuristik yang berbeda.

$$\text{Terpilih} = \{ 'L4': (0.4, 0.6), 'L10': (0.23, 0.5) \}$$

$$\begin{aligned} f(L4) &= g(L4) + h(L4) \\ &= (0.3 + 0.4) + 0.6 \\ &= 1.3 \end{aligned}$$

$$\begin{aligned} f(L10) &= g(L10) + h(L10) \\ &= (0.3 + 0.23) + 0.5 \\ &= 1.03 \end{aligned}$$

L10 memiliki nilai perhitungan paling kecil. Akibatnya, L10 dapat dimasukkan ke dalam *list* tutup.

$$List\ Tutup = [['W2', 'L6', 'L5', 'L10'], 0.53]$$

L10 memiliki keterhubungan dengan dua titik. Titik tersebut adalah L12 dan L9. Kedua titik tersebut memuat informasi bobot dan nilai heuristik masing-masing terhadap W3.

$$Terpilih = \{ 'L12': (1.3, 1.4), 'L9': (0.6, 0.1) \}$$

$$\begin{aligned} f(L9) &= g(L9) + h(L9) \\ &= (0.53 + 0.6) + 0.1 \\ &= 1.23 \end{aligned}$$

$$\begin{aligned} f(L12) &= g(L12) + h(L12) \\ &= (0.53 + 1.3) + 1.4 \\ &= 3.23 \end{aligned}$$

L9 memiliki nilai perhitungan paling minimum. Akibatnya, L9 dapat dimasukkan ke dalam *list* tutup.

$$List\ Tutup = [['W2', 'L6', 'L5', 'L10', 'L9'], 1.13]$$

L9 terhubung dengan L8, W3, dan L11. Kemudian ketiganya mengandung informasi tentang bobot dan nilai heuristiknya adalah:

$$Terpilih = \{ 'L8': (0.4, 0.3), 'L4': (0.7, 0.5), 'W3': (0.1, 0) \}$$

$$\begin{aligned} f(L8) &= g(L8) + h(L8) \\ &= (1.13 + 0.4) + 0.3 \\ &= 1.83 \end{aligned}$$

$$\begin{aligned}
 f(L4) &= g(L4) + h(L4) \\
 &= (1.13 + 0.7) + 0.8 \\
 &= 2.63
 \end{aligned}$$

$$\begin{aligned}
 f(W3) &= g(W3) + h(W3) \\
 &= (1.13 + 0.1) + 0 \\
 &= 1.23
 \end{aligned}$$

W3 memiliki nilai heuristik 0. Disisi lain W3 juga memiliki perhitungan paling kecil. Kondisi ini menyebabkan W3 masuk ke *list* tutup sebagai berikut:

$$List\ Tutup = [['W2', 'L6', 'L5', 'L10', 'L9', 'W3'], 1.23]$$

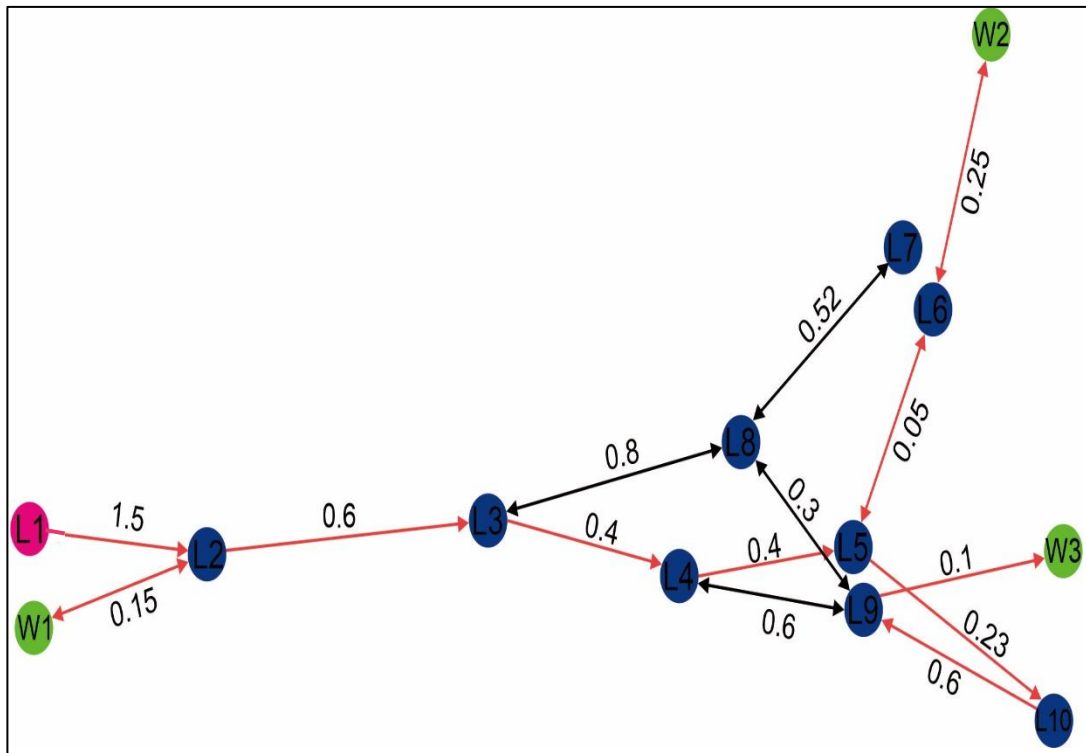
Rute terakhir yang ada pada *list* tutup di atas dapat dimasukkan ke dalam *list* rute. Kemudian dijumlahkan total jarak minimum yang dapat ditempuh.

List Rute

$$\begin{aligned}
 &= [['L1', 'L2', 'W1'], ['W1', 'L2', 'L3', 'L4', 'L5', 'L6', 'W2'], ['W2', 'L6', 'L5', 'L10', \\
 &\quad 'L9', 'W3'], 4.8]
 \end{aligned}$$

Berdasarkan *list* rute di atas, melalui titik L1, Destinasi pertama yang menjadi tujuan adalah W1. Setelah itu dilanjutkan ke destinasi kedua yaitu W2. Selesai dari W2, perjalanan dilanjutkan ke destinasi terakhir yaitu W3. Hasil ini akan tetap walaupun ketika memasukkan titik tujuan secara acak, baik W1 terlebih dahulu, W2 terlebih dahulu, bahkan W3 terlebih dahulu. Karena ketika masuk dalam proses perhitungan akan diproses sesuai dengan algoritma yang telah dibuat sebelumnya.

c. Visualisasi Rute



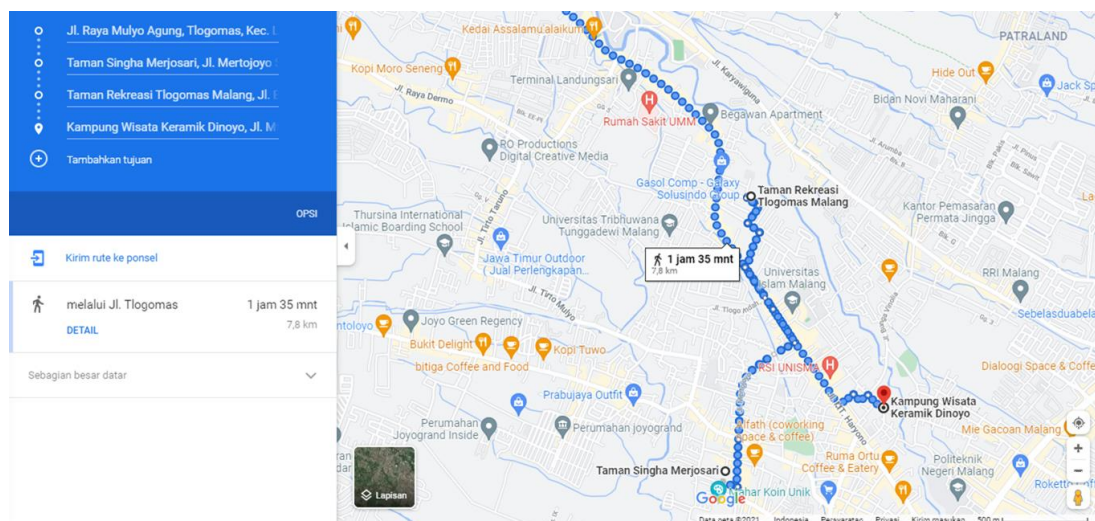
Gambar 4.3 Hasil Pengujian Rute Terpendek

Visualisasi graf ini pada umumnya berwarna biru untuk semua titik dan warna hitam untuk setiap sisi. Garis yang berpanah menunjukkan graf berarah itu sendiri. Selain itu, untuk output hasil akan diberikan warna tersendiri untuk dibedakan. Titik berwarna merah muda menyimbolkan titik awal dalam melakukan perjalanan. Titik ini merupakan titik awal yang telah dimasukkan pada tahap awal. Titik yang berwarna hijau menandakan titik yang akan menjadi tujuan destinasi wisata. Warna sisi yang menjadi rute diberi warna merah agar menjadi pembeda dengan sisi yang tidak dijadikan rute perjalanan.

4.3 Hasil Pengujian

Penelitian ini juga akan menguji rute terpendek pada beberapa destinasi melalui google maps. Pengujian ini akan disesuaikan dengan pengujian menggunakan algoritma *A-star*. Misalnya, pada perhitungan yang telah dijelaskan

di atas adalah dari L1 ke W2, W1, dan W3. Maka pada *google maps* akan dilakukan penelusuran melalui L1 menuju W3 (Taman Merjosari), W1 (Tlogomas Park), dan W2 (Kampung Keramik), dan. Satuan jarak pada penelitian ini menggunakan satuan kilo meter. Salah satu hasil yang ditampilkan pada *google maps* berdasarkan contoh di atas tersaji dalam gambar 4.4.



Gambar 4.4 Visualisasi Rute Pada Maps

Gambar di atas menunjukkan bahwa rute yang terbentuk adalah rute yang berurut sesuai dengan yang telah di masukkan pada kolom tujuan. Tidak terdapat penyeleksian destinasi mana yang terdekat dengan titik awal sehingga dapat dikunjungi terlebih dahulu. Jarak yang ditempuh adalah 7,8 km. Jarak ini lebih panjang daripada yang dihasilkan oleh Algoritma *A-star*.

Proses pengujian ini akan dilakukan sebanyak 30 kali secara acak. Walaupun secara acak, pengujian ini akan dilakukan menyeluruh. Arti menyeluruh di sini adalah seluruh destinasi wisata akan mendapatkan bagian untuk dilakukan proses pengujian sesuai dengan program yang telah dibuat sebelumnya. Hasil dari proses pengujian ini akan ditampilkan melalui tabel 4.9.

Tabel 4.9 Hasil Percobaan Pengujian Rute

No	Titik Awal	Tujuan	Hasil Penelitian	Jarak pada penelitian (km)	Rute Maps	Jarak pada maps (km)
1	L1	W3,W1, W2	L1,W1,W2, W3	4,80	W3,W2, W1	7,80
2	L12	W6,W5,W8	L12,W8,W5, W6	5,75	W6,W5, W8	7,90
3	B6	W16,W4,W11	B6,W11,W4, W16	12,51	W16,W4, W11	18,20
4	D16	W9,W17,W36	D16,W17, W36,W9	14,29	W9,W17, W36	30,60
5	K7	W12,W8,W38	K7,W8,W38, W12	6,32	W12,W8, W38	12,40
6	K18	W19,W7,W10	K18,W7, W10,W19	20,38	W19,W7, W10	28,80
7	K25	W20,W13, W37	K25,W37, W20,W13	5,27	W20,W13, W37	10,10
8	B8	W23,W30, W15	B8,W15,W23, W30	4,72	W23,W30, W15	8,70
9	S4	W26,W16, W37	S4,W16,W37, W26	5,04	W26,W16, W37	8,00
10	K73	W24,W26, W15	K73,W24, W26, W15	2,72	W24,W26, W15	3,50
11	D7	W32,W25, W21,W27	D7,W21, W25,W27, W32	5,29	W32,W25, W21,W27	10,50
12	L46	W28,W24, W26,W21	L46,W21, W28,W26, W24	5,30	W28,W24, W26,W21	12,20
13	L55	W35,W10, W18,W22	L55,W22,W3, W18,W10	16,41	W35,W10, W18,W22	26,30
14	S12	W27,W24, W29,W23	S12,W24, W23,W29, W27	7,12	W27,W24, W29,W23	10,20
15	L43	W22,W35, W31,W11	L43,W11,W2, W31,W35	11,51	W22,W35, W31,W11	17,40
16	K8	W4,W6,W8, W12	K8,W8,W4, W6,W12	7,58	W4,W6, W8,W12	22,80
17	L4	W19,W23, W24,W13	L4,W24,W23, W13,W19	16,41	W19,W23, W24,W13	27,40
18	B9	W3,W5,W7, W9	B9,W9,W5, W3,W7	12,13	W3,W5, W7,W9	22,50
19	K1	W10,W16,W18, W31	K1,W31, W16,W18, W10	15,88	W10,W16, W18,W31	30,40
20	L11	W2,W9,W12, W32	L11,W2,W12, W9,W32	14,81	W2,W9, W12,W32	21,60

No	Titik Awal	Tujuan	Hasil Penelitian	Jarak pada penelitian (km)	Rute Maps	Jarak pada maps (km)
21	S5	W8,W16, W24,W25, W26	S5,W25,W26, W24,W8, W16	5,65	W8,W16, W24,W25, W26	12,80
22	K27	W13,W15, W29,W30, W31	K27,W31, W30,W29, W15,W13	6,85	W13,W15, W29,W30, W31	13,20
23	B27	W32,W35, W37,W29, W31	B27,W29,W3, W31,W35, W32	9,35	W32,W35, W37,W29, W31	16,20
24	K39	W7,W27, W37,W8, W9	K39,W27, W37,W8,W7, W9	10,03	W7,W27, W37,W8, W9	24,70
25	L25	W25,W5, W15,W16, W17	L25,W5,W15, W25,W16, W17	16,83	W25,W5, W15,W16, W17	30,50
26	L44	W30,W7,W27, W1,W4,W6	L44,W6,W4, W7,W30,W27, W1	13,91	W30,W7, W27,W1, W4,W6	24,70
27	K23	W10,W9,W11, W12,W14, W15	K23,W15, W14,W12,W9, W10,W11	8,61	W10,W9, W11,W12, W14,W15	26,90
28	B20	W21,W13, W18,W26, W27,W30	B20,W13,W2, W26,W27, W30,W18	10,25	W21,W13, W18,W26, W27,W30	19,80
29	S6	W17,W18, W20,W31, W32,W33	S6,W20,W33, W31,W18, W17	12,15	W17,W18, W20,W31, W32,W33	20,60
30	D6	W27,W26, W28,W11, W21,W23	D6,W21,W28, W26,W23, W27,W11	11,14	W27,W26, W28,W11, W21,W23	23,40

4.4 Evaluasi Pengujian

Setelah dilakukan pengujian sebanyak 30 kali didapatkan hasil bahwa pengujian yang dilakukan dengan algoritma *A-star* memiliki jarak yang relatif lebih pendek daripada jarak yang ditampilkan melalui google maps. Persentase kehematan/ efektivitas jarak tempuh hasil pengujian didapatkan melalui

$$\text{nilai efektivitas} = \frac{JM-JP}{JM} \times 100\% \quad (3)$$

Keterangan:

JM = Jarak yang ditampilkan pada google maps

JP = Jarak ditampilkan pada penelitian

Hasil perhitungan nilai efektifitas tersebut tersaji pada tabel 4.10.

Tabel 4.10 Hasil Persentase Keefektifan Rute

Pengujian Ke-	Persentase (%)	Pengujian Ke-	Persentase (%)
1	38,46%	16	66,75%
2	27,22%	17	40,11%
3	31,26%	18	46,09%
4	53,30%	19	47,76%
5	49,03%	20	31,44%
6	29,24%	21	55,86%
7	47,82%	22	48,11%
8	45,75%	23	42,28%
9	37,00%	24	59,39%
10	22,29%	25	44,82%
11	49,62%	26	43,68%
12	56,56%	27	67,99%
13	37,60%	28	48,23%
14	30,20%	29	41,02%
15	33,85%	30	52,39%
Rata-rata			44,17%

Tabel di atas menunjukkan pengujian yang dilakukan menggunakan algoritma *A-star* memiliki kehematan jarak tempuh dengan rata-rata persentasenya adalah 44,17 %. Persentase kehematan paling tinggi terjadi pada percobaan ke-27. Sedangkan persentase paling rendah kehematan jarak tempuh terdapat pada percobaan ke-10. Rendahnya jarak tempuh yang dihasilkan oleh Algoritma *A-star* membuat algoritma ini lebih efektif.

Ada beberapa alasan yang membuat hasil pengujian rute terpendek menuju beberapa destinasi wisata jauh lebih efektif. Berikut merupakan penjelasannya:

a. Perhitungan Nilai Heuristik

Keunikan khas yang dimiliki oleh algoritma *A-star* adalah perhitungan nilai heuristiknya. Nilai heuristik merupakan jarak antara setiap titik yang dihubungkan dengan titik tujuan. Proses perhitungannya melalui perhitungan jarak euclidan yang diambil melalui titik koordinat masing-masing titik yang tersebar pada gambar.

Keefektifan nilai heuristik ini dapat diketahui pada pengujian ke-10. Pengujian yang dilakukan dengan titik awal adalah K73 dan penginputan destinasi tujuan W24 (Patung Chairil Anwar), W26 (Alun-Alun Merdeka), dan W15 (Kampung Jodipan). Hasil yang ditunjukkan oleh Algoritma *A-star* dan google maps sama-sama menunjukkan bahwa destinasi yang dituju pertama adalah Patung Chairil Anwar, kemudian Alun-alun Merdeka, dan terakhir adalah Kampung Jodipan.

Perbedaannya adalah pada jarak yang ditempuh. Jarak rute yang ditempuh berdasarkan algoritma *A-star* sebesar 2,72 km. Hal ini lebih rendah daripada jarak yang dihasilkan oleh google maps sebesar 3.5 km. Walaupun prioritas destinasi wisata yang dituju sama, namun rute yang dilalui berbeda. Hal ini merupakan efek dari perhitungan nilai heuristik dalam penentuan rute.

b. Penyeleksian Prioritas Rute Destinasi

Proses penelitian ini tidak berfokus pada satu tujuan destinasi saja, melainkan dilakukan pada beberapa titik tujuan destinasi wisata. Ketika dipilih terdapat tiga tujuan, maka dari ke tiga tujuan tersebut akan diseleksi masing-masing dengan algoritma *A-star*. Kemudian akan dipilih destinasi mana yang akan menjadi tujuan pertama. Untuk menentukan tujuan destinasi kedua juga akan

dilakukan seperti halnya yang pertama. Hal ini dilakukan sampai pada tujuan destinasi terakhir.

Akibat dari proses desain tersebut, pengujian rute tersebut lebih efektif dari pada pengujian rute yang dilakukan melalui *google maps*. Hal ini dikarenakan pada *google maps*, tidak adanya penyeleksian destinasi wisata. Jika seseorang memilih W10, W20, dan W30 maka pada *google maps* akan menampilkan destinasi pertama W10, kemudian W20, dan terakhir W30. Berbeda dengan pengujian dengan algoritma *A-star*. Walaupun yang dipilih adalah W10, W20, dan W30, belum tentu W10 menjadi tujuan destinasi pertama. Destinasi yang pertama dikunjungi adalah destinasi yang memiliki rute terpendek dari titik asal. Sehingga desain ini juga turut andil dalam menjadikan pemilihan rute lebih efektif.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan dapat disimpulkan bahwa penentuan rute terpendek menggunakan algoritma *A-star* dinilai lebih efektif dalam sisi jarak. Hal ini dibuktikan dengan pengujian rute terpendek yang telah dilakukan secara acak dan menyeluruh terhadap seluruh destinasi wisata sebanyak 30 kali. Hasilnya didapatkan bahwa rata-rata jarak yang dihasilkan lebih pendek/ efektif 44,17 % dibandingkan jarak rute yang dihasilkan pada *google maps*.

Penyebab keefektifan ini ada dua. Peratama adalah adanya nilai heuristik setiap titik terhadap titik tujuan. Kondisi ini menjadikan pemilihan rute lebih tepat, sehingga jarak yang ditempuh lebih pendek. Hal ini dapat dilihat dari percobaan ke-10 di mana destinasi yang menjadi prioritas adalah sama, namun rute yang dihasilkan algoritma *A-star* lebih pendek dibandingkan dengan *google maps*.

Penyebab keefektifan kedua adalah desain penelitian tentang pembuatan prioritas destinasi. Tujuan pembuatan prioritas destinasi adalah menyeleksi beberapa destinasi tujuan, kemudian menentukan destinasi mana yang akan dikunjungi terlebih dahulu yang memiliki jarak terdekat dengan titik awal. Dalam penyeleksian prioritas ini akan digunakan algoritma *A-star*. Inovasi desain penelitian ini menjadikan pemilihan rute jauh lebih efektif dibandingkan pemilihan rute yang dilakukan melalui *google maps*. *Google maps* sendiri tidak melakukan penyeleksian prioritas, hanya menampilkan rute sesuai urutan penginputan destinasi tujuan.

5.2 Saran

Penelitian ini telah menghasilkan modifikasi baru terhadap penentuan beberapa rute destinasi tujuan menggunakan algoritma *A-star*. Diharapkan adanya penelitian ini dapat menjadi pondasi untuk penelitian selanjutnya terutama dalam menentukan rute destinasi di luar wilayah Kota Malang. Selain itu diharapkan penelitian ini bisa dikembangkan lebih terutama dalam penampilan visualisasinya agar lebih rapi sesuai dengan gambar peta pada *google maps*.

DAFTAR PUSTAKA

- Abdussakir. 2009. *Teori Graf: Topik Dasar untuk Tugas Akhir/Skripsi*. Malang: UIN Press.
- Agarwal, Himanesh. 2016. *Time-Efficient A* Algorithm for Robot Path Planning* dalam *Procedia Technology 23* (hlm. 144 – 149) . India: Madhav Institute of Science & Technology.
- Al-Qur'an dan Terjemah. Cetakan ke-7: Al-Mizan Publishing House.
- Badan Perencanaan dan Pembangunan Daerah Kota Malang. 2021. *Peta Zonasi Kota Malang*, <https://si-petarungv2.malangkota.go.id/peta-zonasi>, diakses pada 15 Juni 2021 pukul 07.30.
- Badan Pusat Statistik Malang. 2019. *Konsep dan Definisi Statistik Kunjungan Wisatawan*, <https://malangkota.bps.go.id/subject/16/pariwisata.html>, diakses pada 15 Oktober 2020 pukul 10.27.
- Budayasa, I Ketut. 2016. *Teori Graph dan Aplikasinya*. Surabaya: Unesa University Press.
- Chartrand, L dan Leniak, L. 2000. *Graph & Digraph*. California: Chapman and Hall.
- Cipta. 2018. *Teori dan Algoritma Graph dengan Bahasa Java*. Yogyakarta: Andi
- Disporapar Malang Kota. 2019. *Data Pariwisata Kota Malang tahun 2019*, <https://disporapar.malangkota.go.id/peta-zonasi>, diakses pada 18 Maret 2021 pukul 09.30.
- Dona, dkk. 2020. *Aplikasi Pencarian Rute Terpendek Lokasi Kuliner Khas Palembang Menggunakan Algoritma Euclidean Distance dan A*(Star)* dalam *Jurnal SISFOKOM (Sistem Informasi dan Komputer) Volume 9 Nomor 2* (hlm. 195–202). Palembang: Universitas Indo Global Mandiri.
- Fernando, Yusra. 2020. *Penerapan Algoritma A-Star Pada Aplikasi Pencarian Lokasi Fotografi Di Bandar Lampung Berbasis Android* dalam *Jurnal TEKNOINFO Volume 14 Nomor 1* (hlm. 27–34) , Lampung : Universitas Lampung.
- Hadist riwayat Bukhori no.1 dan Muslim no. 1907.

- Hagberg, Aric, dkk. 2021. *Networkx Reference 2.6.2*. Python : NetworkxGoogleGroup.
- Hart, N. 1995. *Artificial Intelligence: A Modern Approach*. Hidayati, Nurul. 2017. *Makna Rihlah dan Safar dalam Al-Qur'an Studi Penafsiran Ibnu Katsir dan M. Quraish Syihab*. Skripsi. Prodi Ilmu Qur'an dan Tafsir. UIN Sunan Kalijaga. Yogyakarta.
- Maulana, Afrizal Adam. 2017. *Implementation of A* Algorithm in Web-Based Applications for Finding the Shortest Route as Navigation of Digital Indoor Map* dalam *Citec Journal Vol. 5 No. 1*. Jakarta: Teknik Informatika UDINUS.
- Munir, Rinaldi. 2016. *Matematika Diskrit*. Bandung: Program Studi Teknik Informatika ITB.
- Pekandayu. 2020. *Buku Panduan Kakang Mbakyu Kota Malang 2020*. Malang: Disporapar Kota Malang
- Rozikin, Muhammad Zainur. 2020. *Implementasi Algoritma Floyd-Warshall dalam Menentukan Jalur Terpendek Kunjungan Wisata di Kota Batu*. Skripsi. Prodi Matematika. UIN Maulana Malik Ibrahim. Malang
- Sandra, dkk. 2019. *Implementasi Rencana Perjalanan Wisata Di Kotabogor Menggunakan Algoritma Greedy dan A* Berbasis Website* dalam *Jurnal Ilmiah Teknologi dan Rekayasa Volume 24 No. 2 (hlm 114 - 130)*. Depok: Fakultas Teknologi Industri Universitas Gunadarma.
- TIK, Relawan. 2017. *Buku Panduan Pemograman Python*. Pematang: Pemerintah Kabupaten Pematang

LAMPIRAN

1. Data Titik Node

No	Titik 1	Titik 2	Jarak	No	Titik 1	Titik 2	Jarak	No	Titik 1	Titik 2	Jarak
1	L1	L2	1.5	32	B15	D6	0.48	63	K10	L57	0.4
2	L2	L1	1.5	33	B15	D16	1.5	64	K10	W29	0.4
3	L2	W1	0.15	34	B16	B10	1.3	65	K10	K63	0.3
4	L2	L3	0.6	35	B16	D1	0.6	66	K11	K16	0.5
5	W1	L2	0.15	36	B16	B12	1.7	67	K11	K12	0.3
6	L3	L2	0.6	37	B16	B17	0.45	68	K73	K11	0.45
7	L3	L4	0.5	38	B17	B16	0.45	69	K73	W35	0.3
8	L3	L8	0.8	39	B17	B18	1.2	70	K12	W35	0.25
9	L4	L3	0.4	40	B17	B29	1	71	K13	W30	0.3
10	L4	L5	0.4	41	B18	B17	1.2	72	K13	W31	0.25
11	L4	L9	0.6	42	B18	B9	0.1	73	K13	K24	0.4
12	L5	L4	0.4	43	B18	B20	0.6	74	K13	W34	0.5
13	L5	L6	0.05	44	B20	B18	0.6	75	W30	W8	0.3
14	L5	L10	0.23	45	B20	B22	0.13	76	W30	K12	0.3
15	L6	W2	0.25	46	B20	B27	0.8	77	W30	K13	0.3
16	L6	L13	0.8	47	B22	B20	0.13	78	K15	W30	0.3
17	L6	L5	0.05	48	B22	B25	0.13	79	K15	K20	0.2
18	W2	L14	0.3	49	B22	B23	0.15	80	K16	K17	0.3
19	W2	L6	0.25	50	B23	B22	0.15	81	K16	K15	0.1
20	L7	L8	0.52	51	B23	B24	0.3	82	K17	K18	0.2
21	L8	L7	0.52	52	B23	B26	0.1	83	K17	W29	0.3
22	L8	L3	0.8	53	B24	W12	0.1	84	W29	K10	0.4
23	L8	L9	0.3	54	B24	B23	0.3	85	W29	K17	0.3
24	L9	L4	0.6	55	W12	B24	0.1	86	W29	K62	0.4
25	L9	L10	0.6	56	W12	B25	0.1	87	K18	K17	0.2
26	L9	L8	0.3	57	B25	W12	0.1	88	K18	W24	0.5
27	L9	W3	0.1	58	B25	B22	0.13	89	K18	K60	0.5
28	W3	L9	0.1	59	B25	B8	0.82	90	K18	K19	0.2
29	W3	L11	0.4	60	B26	B23	0.1	91	K19	K18	0.2
30	L10	L9	0.6	61	B26	B5	0.4	92	K19	K21	0.4
31	L10	L5	0.23	62	B26	B27	0.1	93	K19	K20	0.1

No	Titik 1	Titik 2	Jarak	No	Titik 1	Titik 2	Jarak	No	Titik 1	Titik 2	Jarak
94	L10	L12	1.3	128	B27	B26	0.1	162	K20	K19	0.1
95	L11	W3	0.4	129	B27	B20	0.8	163	K20	K15	0.2
96	L11	S2	1.6	130	B27	B28	1.2	164	K20	K22	0.3
97	L11	L12	1.5	131	B27	B34	0.5	165	K20	W34	0.1
98	L12	L10	1.3	132	B28	B27	1.2	166	W34	K20	0.1
99	L12	L11	1.5	133	B28	B29	0.2	167	W34	K13	0.5
100	L12	L16	0.8	134	B28	B31	0.3	168	K21	W27	0.1
101	L12	L24	0.6	135	B29	B28	0.3	169	K21	K19	0.4
102	L13	L6	0.8	136	B29	B17	1	170	K21	K22	0.1
103	L13	L27	0.3	137	B29	B30	0.4	171	K22	K21	0.1
104	L13	K1	1.1	138	B30	B29	0.4	172	K22	K20	0.3
105	L14	W2	0.3	139	B31	B28	0.3	173	K22	K23	0.2
106	L14	L15	0.7	140	B31	B32	1.2	174	K23	K22	0.15
107	L14	L28	1	141	B31	B34	1.2	175	K23	W34	0.3
108	L15	L14	0.7	142	B32	B31	1.2	176	K23	K24	0.15
109	L15	L31	1.3	143	B32	B33	0.2	177	K24	K23	0.15
110	L16	L12	0.8	144	B32	B35	1.5	178	K24	K13	0.4
111	L16	K8	0.25	145	B33	B32	0.2	179	K24	K25	0.2
112	L16	L17	0.7	146	B34	B31	1.2	180	K25	K24	0.2
113	L17	L16	0.7	147	B34	B27	0.5	181	K25	K29	0.1
114	L17	W7	0.3	148	B34	B38	0.53	182	K25	K26	0.3
115	L17	W8	0.4	149	B34	B35	0.8	183	K26	K25	0.3
116	W7	L17	0.3	150	B35	B34	0.8	184	K26	K28	0.2
117	W7	K5	0.4	151	B35	B32	1.5	185	K26	K27	0.4
118	W7	L19	0.3	152	B35	B36	0.22	186	K27	K26	0.4
119	L19	W7	0.3	153	B36	B35	0.22	187	K27	K30	0.4
120	L19	L20	0.1	154	B36	B2	0.15	188	K27	K7	0.4
121	L20	L21	0.5	155	B36	B37	0.8	189	K27	K28	0.2
122	L20	K2	0.01	156	B37	B36	0.8	190	K28	K27	0.2
123	L21	L22	0.1	157	D1	B16	0.6	191	K28	K26	0.2
124	L21	L26	0.01	158	D1	D6	1.7	192	K28	W33	0.1
125	L22	L23	0.1	159	D1	D2	0.3	193	W33	K28	0.1
126	L23	L24	0.01	160	D2	D1	0.3	194	W33	K38	0.15
127	L23	L12	0.6	161	D2	D4	0.5	195	K29	K25	0.1

2. Script Code

```
#-----pendefinisian nilai heuristik-----#
def AmbilJarak(awal, akhir):
    return ((AmbilAbsis(awal) - AmbilAbsis(akhir))**2 +(AmbilOrdinat(awal) -
        AmbilOrdinat(akhir))**2)**(1/2)

#-----algoritma A-star-----#
def astar_riyan(G, asal, target, heuristik=None, bobot="bobot"):
    c = count()
    ListBuka = [(0, next(c), asal, 0, None)]
    Terseleksi = {}
    ListTutup = {}
    while ListBuka:
        _, __, node_terkini, dist, pangkal = heappop(ListBuka)
        # print('queue=',queue)
        if node_terkini == target:
            jalur_terpilih = [node_terkini]
            node = pangkal
            while node is not None:
                jalur_terpilih.append(node)
                node = ListTutup[node]
            jalur_terpilih.reverse()
            return jalur_terpilih
        if node_terkini in ListTutup:
            if ListTutup[node_terkini] is None:
                continue
            fx, h = Terseleksi[node_terkini]
            if fx < dist:
                continue
            ListTutup[node_terkini] = pangkal
        for tetangga, w in G[node_terkini].items():
            gx = dist + bobot(node_terkini, tetangga, w)
            if tetangga in Terseleksi:
                fx, h = Terseleksi[tetangga]
```

```

        if fx <= gx:
            continue
        else:
            h = heuristik(tetangga, target)
            Terseleksi[tetangga] = gx, h
            heappush(ListBuka, (gx + h, next(c), tetangga, gx, node_terkini))
    print('node', asal, 'atau target', target, 'tidak ditemukan')
#-----proses pengujian rute terpendek-----
#
def seleksi(A,node_tujuan,jalur,dist):
    seleksi1=[]
    if len(node_tujuan)==0:
        total=sum(dist)
        print('jalur : ',jalur)
        print('total jarak : ',total,' km')
    else :
        for i in node_tujuan:
            lintasan=astar_riyan(graf,A,i,heuristik=AmbilJarak,bobot='bobot')
            jarak=fungsi_jarak(graf,A,i,heuristik=AmbilJarak,bobot='bobot')
            seleksi1.append([lintasan,jarak])
        K1=min(seleksi1, key=lambda x: x[1])
        seleksi1.clear()
        jalur.append(K1[0])
        dist.append(K1[1])
        K2=str(K1[0][-1])
        node_tujuan.remove(K2)
    return seleksi(K2,node_tujuan,jalur,dist)

```

3. Hasil Pengujian Rute Penelitian

Pengujian Ke-	Hasil
1	Masukkan Node Awal:L1 Masukkan Node Tujuan:W2,W3,W1 jalur : [['L1', 'L2', 'W1'], ['W1', 'L2', 'L3', 'L4', 'L5', 'L6', 'W2'], ['W2', 'L6', 'L5', 'L10', 'L9', 'W3']] total jarak : 4.83 km
2	Masukkan Node Awal:L12 Masukkan Node Tujuan:W6,W5,W8 jalur : [['L12', 'L16', 'L17', 'W8'], ['W8', 'W35', 'K5', 'K4', 'K1', 'L13', 'L27', 'L28', 'L29', 'W5'], ['W5', 'L29', 'L30', 'L31', 'L32', 'L33', 'W6']] total jarak : 5.75 km
3	Masukkan Node Awal:B6 Masukkan Node Tujuan:W16,W4,W11 jalur : [['B6', 'B5', 'B4', 'B38', 'B3', 'B2', 'W10', 'B1', 'W11'], ['W11', 'B1', 'W10', 'B2', 'B3', 'B38', 'B4', 'B5', 'B6', 'L45', 'L43', 'L44', 'L46', 'L35', 'W5', 'L29', 'L30', 'W4'], ['W4', 'L30', 'L31', 'L32', 'L34', 'L35', 'L46', 'L47', 'L48', 'L50', 'L51', 'L52', 'L56', 'L57', 'W14', 'K69', 'K64', 'K62', 'W33', 'K38', 'W16']] total jarak : 12.509999999999998 km
4	Masukkan Node Awal:D16 Masukkan Node Tujuan:W9,W17,W36 jalur : [['D16', 'D15', 'D14', 'D13', 'W17'], ['W17', 'D13', 'D8', 'D7', 'D6', 'B15', 'B13', 'K69', 'K64', 'K62', 'W33', 'K28', 'K27', 'K30', 'W37', 'K31', 'W36'], ['W36', 'K31', 'W37', 'K32', 'K36', 'W31', 'K13', 'W34', 'K20', 'K19', 'K18', 'K17', 'W29', 'K10', 'L57', 'L56', 'L55', 'L54', 'L53', 'B7', 'B6', 'B5', 'B4', 'B38', 'B3', 'W9']] total jarak : 14.29 km
5	Masukkan Node Awal:K7 Masukkan Node Tujuan:W12,W8,W38 jalur : [['K7', 'K6', 'L17', 'W8'], ['W8', 'K35', 'K36', 'W31', 'K13', 'W34', 'K20', 'K19', 'K18', 'K60', 'W38'], ['W38', 'K62', 'K64', 'K69', 'W14', 'W13', 'B10', 'B9', 'B18', 'B20', 'B22', 'B25', 'W12']] total jarak : 6.32 km

Pengujian Ke-	Hasil
6	DiGraph with 236 nodes and 620 edges Masukkan Node Awal:K18 Masukkan Node Tujuan:W19,W7,W10 jalur : [['K18', 'K19', 'K20', 'K15', 'W30', 'K12', 'W35', 'K5', 'W7'], ['W7', 'L19', 'L20', 'K2', 'K3', 'K1', 'L13', 'L27', 'L28', 'L29', 'L30', 'L31', 'L32', 'L33', 'W6', 'L44', 'L43', 'L45', 'B6', 'B5', 'B4', 'B38', 'B3', 'B2', 'W10'], ['W10', 'B2', 'B3', 'B38', 'B4', 'B5', 'B6', 'B7', 'L53', 'L54', 'L55', 'L56', 'L57', 'W14', 'K69', 'K70', 'K71', 'W21', 'K72', 'B14', 'D17', 'D16', 'D15', 'D18', 'D19', 'D22', 'W19']] total jarak : 20.380000000000003 km
7	Masukkan Node Awal:K25 Masukkan Node Tujuan:W20,W13,W37 jalur : [['K25', 'K29', 'K30', 'W37'], ['W37', 'K32', 'K36', 'W31', 'K29', 'K25', 'K24', 'K23', 'K22', 'K21', 'W27', 'W25', 'K58', 'K54', 'K55', 'K72', 'W20'], ['W20', 'K72', 'W21', 'K71', 'K70', 'K69', 'W14', 'W13']] total jarak : 5.2700000000000005 km
8	Masukkan Node Awal:B8 Masukkan Node Tujuan:W23,W30,W15 jalur : [['B8', 'L53', 'L54', 'L55', 'L56', 'L57', 'W14', 'K69', 'K70', 'K71', 'W21', 'W15'], ['W15', 'W21', 'K71', 'K66', 'K67', 'K59', 'W23'], ['W23', 'K59', 'K60', 'K18', 'K19', 'K20', 'K15', 'W30']] total jarak : 4.7200000000000001 km
9	Masukkan Node Awal:S4 Masukkan Node Tujuan:W26,W16,W37 jalur : [['S4', 'K39', 'K38', 'W16'], ['W16', 'K38', 'W33', 'K28', 'K27', 'K30', 'W37'], ['W37', 'K32', 'K36', 'W31', 'K29', 'K25', 'K24', 'K23', 'K22', 'K21', 'W27', 'W25', 'K58', 'K54', 'K53', 'W26']] total jarak : 5.0400000000000001 km
10	Masukkan Node Awal:K73 Masukkan Node Tujuan:W24,W26,W15 jalur : [['K73', 'K11', 'K16', 'K17', 'K18', 'W24'], ['W24', 'W25', 'K58', 'K54', 'K53', 'W26'], ['W26', 'W25', 'K58', 'K54', 'K55', 'K72', 'W21', 'W15']] total jarak : 2.7199999999999998 km
11	Masukkan Node Awal:D7 Masukkan Node Tujuan:W32,W25,W21,W27 jalur : [['D7', 'D6', 'B15', 'B13', 'K70', 'K71', 'W21'], ['W21', 'K71', 'K66', 'K67', 'K59', 'W23', 'W24', 'W25'], ['W25', 'K58', 'K54', 'K53', 'K52', 'K41', 'W27'], ['W27', 'W25', 'K58', 'K54', 'K50', 'K47', 'K48', 'W32']] total jarak : 5.29 km

Pengujian Ke-	Hasil
12	<p>Masukkan Node Awal:L46 Masukkan Node Tujuan:W28,W24,W26,W21 jalur : [['L46', 'L47', 'L48', 'L50', 'L51', 'L52', 'L56', 'L57', 'W14', 'K69', 'K70', 'K71', 'W21'], ['W21', 'K72', 'W20', 'W28'], ['W28', 'K55', 'K56', 'K57', 'K58', 'K54', 'K53', 'W26'], ['W26', 'W25', 'W24']]</p> <p>total jarak : 5.300000000000001 km</p>
13	<p>asukkan Node Awal:L55 Masukkan Node Tujuan:W35,W10,W18,W22 jalur : [['L55', 'L56', 'L57', 'W14', 'K69', 'K64', 'K68', 'W22'], ['W22', 'K65', 'K67', 'K59', 'K60', 'K18', 'K19', 'K20', 'K15', 'W30', 'K12', 'W35'], ['W35', 'W8', 'K35', 'K36', 'W31', 'K29', 'K25', 'K24', 'K23', 'K22', 'K21', 'W27', 'W25', 'K58', 'K54', 'K55', 'K72', 'B14', 'D17', 'D16', 'D15', 'D14', 'W18'], ['W18', 'D14', 'D15', 'D16', 'B15', 'B13', 'K69', 'W14', 'L57', 'L56', 'L55', 'L54', 'L53', 'B7', 'B6', 'B5', 'B4', 'B38', 'B3', 'B2', 'W10']]</p> <p>total jarak : 16.41 km</p>
14	<p>Masukkan Node Awal:S12 Masukkan Node Tujuan:W27,W24,W29,W23 jalur : [['S12', 'S11', 'S10', 'W20', 'W28', 'K55', 'K56', 'K57', 'W24'], ['W24', 'W23'], ['W23', 'K59', 'K60', 'W38', 'K62', 'W29'], ['W29', 'K17', 'K18', 'K19', 'K21', 'W27']]</p> <p>total jarak : 7.120000000000001 km</p>
15	<p>Masukkan Node Awal:L43 Masukkan Node Tujuan:W22,W35,W31,W11 jalur : [['L43', 'L45', 'B6', 'B5', 'B4', 'B38', 'B3', 'B2', 'W10', 'B1', 'W11'], ['W11', 'B1', 'W10', 'B2', 'B3', 'B38', 'B4', 'B5', 'B6', 'B7', 'L53', 'L54', 'L55', 'L56', 'L57', 'W14', 'K69', 'K64', 'K68', 'W22'], ['W22', 'K65', 'K67', 'K59', 'K60', 'K18', 'K19', 'K20', 'W34', 'K13', 'W31'], ['W31', 'K13', 'W30', 'K12', 'W35']]</p> <p>total jarak : 11.51 km</p>
16	<p>Masukkan Node Awal:K8 Masukkan Node Tujuan:W4,W6,W8,W12 jalur : [['K8', 'K7', 'K6', 'L17', 'W8'], ['W8', 'W35', 'K5', 'K4', 'K1', 'L13', 'L27', 'L28', 'L29', 'L30', 'W4'], ['W4', 'L30', 'L31', 'L32', 'L33', 'W6'], ['W6', 'L44', 'L43', 'L45', 'B6', 'B5', 'B26', 'B23', 'B22', 'B25', 'W12']]</p> <p>total jarak : 7.580000000000001 km</p>

Pengujian Ke-	Hasil
17	Masukkan Node Awal:L4 Masukkan Node Tujuan:W19,W23,W24,W13 jalur : [['L4', 'L5', 'L6', 'L13', 'K1', 'K4', 'K73', 'K11', 'K16', 'K17', 'K18', 'W24'], ['W24', 'W23'], ['W23', 'K59', 'K60', 'W38', 'K62', 'K64', 'K69', 'W14', 'W13'], ['W13', 'W14', 'K69', 'K70', 'K71', 'W21', 'K72', 'B14', 'D17', 'D16', 'D15', 'D18', 'D19', 'D22', 'W19']] total jarak : 16.91 km
18	Masukkan Node Awal:B9 Masukkan Node Tujuan:W3,W5,W7,W9 jalur : [['B9', 'B18', 'B20', 'B22', 'B23', 'B26', 'B5', 'B4', 'B38', 'B3', 'W9'], ['W9', 'L41', 'L40', 'L39', 'L33', 'L34', 'L35', 'W5'], ['W5', 'L38', 'L36', 'L13', 'L6', 'L5', 'L10', 'L9', 'W3'], ['W3', 'L11', 'L12', 'L16', 'L17', 'W7']] total jarak : 12.129999999999999 km
19	Masukkan Node Awal:K1 Masukkan Node Tujuan:W10,W16,W18,W31 jalur : [['K1', 'K4', 'K5', 'W35', 'W8', 'K35', 'K36', 'W31'], ['W31', 'K29', 'K25', 'K26', 'K28', 'W33', 'K38', 'W16'], ['W16', 'K38', 'W33', 'K28', 'K26', 'K25', 'K24', 'K23', 'K22', 'K21', 'W27', 'W25', 'K58', 'K54', 'K55', 'K72', 'B14', 'D17', 'D16', 'D15', 'D14', 'W18'], ['W18', 'D14', 'D15', 'D16', 'B15', 'B13', 'K69', 'W14', 'L57', 'L56', 'L55', 'L54', 'L53', 'B7', 'B6', 'B5', 'B4', 'B38', 'B3', 'B2', 'W10']] total jarak : 15.879999999999999 km
20	Masukkan Node Awal:L11 Masukkan Node Tujuan:W2,W9,W12,W32 jalur : [['L11', 'W3', 'L9', 'L10', 'L5', 'L6', 'W2'], ['W2', 'L14', 'L28', 'L29', 'L30', 'L31', 'L32', 'L33', 'W6', 'L44', 'L43', 'L45', 'B6', 'B5', 'B26', 'B23', 'B22', 'B25', 'W12'], ['W12', 'B25', 'B22', 'B23', 'B26', 'B5', 'B4', 'B38', 'B3', 'W9'], ['W9', 'B3', 'B38', 'B4', 'B5', 'B6', 'B7', 'L53', 'L54', 'L55', 'L56', 'L57', 'W14', 'K69', 'K64', 'K68', 'W22', 'K65', 'K67', 'K59', 'W23', 'W24', 'W25', 'K58', 'K54', 'K50', 'K47', 'K48', 'W32']] total jarak : 14.809999999999999 km
21	Masukkan Node Awal:S5 Masukkan Node Tujuan:W8,W16,W24,W25,W26 jalur : [['S5', 'K43', 'K42', 'K40', 'K41', 'W27', 'W25'], ['W25', 'K58', 'K54', 'K53', 'W26'], ['W26', 'W25', 'W24'], ['W24', 'K18', 'K19', 'K20', 'K15', 'W30', 'W8'], ['W8', 'K35', 'K36', 'W31', 'K29', 'K25', 'K26', 'K28', 'W33', 'K38', 'W16']] total jarak : 5.65 km

Pengujian Ke-	Hasil
22	<p>Masukkan Node Awal:K27 Masukkan Node Tujuan:W13,W15,W29,W30,W31 jalur : [['K27', 'K30', 'W37', 'K32', 'K36', 'W31'], ['W31', 'K13', 'W30'], ['W30', 'K13', 'W34', 'K20', 'K19', 'K18', 'K17', 'W29'], ['W29', 'K62', 'K64', 'K68', 'W22', 'K66', 'K71', 'W21', 'W15'], ['W15', 'W21', 'K71', 'K70', 'K69', 'W14', 'W13']]</p> <p>total jarak : 6.85 km</p>
23	<p>Masukkan Node Awal:B27 Masukkan Node Tujuan:W32,W35,W37,W29,W31 jalur : [['B27', 'B26', 'B23', 'B22', 'B20', 'B18', 'B9', 'B10', 'W13', 'W14', 'K69', 'K64', 'K62', 'W29'], ['W29', 'K62', 'W33', 'K28', 'K27', 'K30', 'W37'], ['W37', 'K32', 'K36', 'W31'], ['W31', 'K13', 'W30', 'K12', 'W35'], ['W35', 'W8', 'K35', 'K36', 'W31', 'K29', 'K25', 'K24', 'K23', 'K22', 'K21', 'W27', 'W25', 'K58', 'K54', 'K50', 'K47', 'K48', 'W32']]</p> <p>total jarak : 9.35 km</p>
24	<p>Masukkan Node Awal:K39 Masukkan Node Tujuan:W7,W27,W37,W8,W9 jalur : [['K39', 'K40', 'K41', 'W27'], ['W27', 'K21', 'K22', 'K23', 'K24', 'K25', 'K29', 'K30', 'W37'], ['W37', 'K32', 'K34', 'W8'], ['W8', 'W35', 'K5', 'W7'], ['W7', 'L19', 'L20', 'K2', 'K3', 'K1', 'L13', 'L27', 'L28', 'L29', 'L30', 'L31', 'L32', 'L33', 'L39', 'L40', 'L41', 'W9']]</p> <p>total jarak : 10.030000000000001 km</p>
25	<p>Masukkan Node Awal:L25 Masukkan Node Tujuan:W25,W5,W15,W16,W17 jalur : [['L25', 'L26', 'K2', 'K3', 'K1', 'L13', 'L27', 'L28', 'L29', 'W5'], ['W5', 'L38', 'L47', 'L48', 'L50', 'L51', 'L52', 'L56', 'L57', 'W14', 'K69', 'K70', 'K71', 'W21', 'W15'], ['W15', 'W21', 'K71', 'K66', 'K67', 'K59', 'W23', 'W24', 'W25'], ['W25', 'W24', 'W23', 'K59', 'K60', 'W38', 'K62', 'W33', 'K38', 'W16'], ['W16', 'K38', 'W33', 'K28', 'K26', 'K25', 'K24', 'K23', 'K22', 'K21', 'W27', 'W25', 'K58', 'K54', 'K55', 'K56', 'W21', 'K71', 'K70', 'B13', 'B15', 'D6', 'D7', 'D8', 'D13', 'W17']]</p> <p>total jarak : 16.83 km</p>
26	<p>Masukkan Node Awal:L44 Masukkan Node Tujuan:W30,W7,W27,W1,W4,W6 jalur : [['L44', 'W6'], ['W6', 'L33', 'L34', 'L35', 'W5', 'L29', 'L30', 'W4'], ['W4', 'L30', 'L31', 'L32', 'L34', 'L35', 'W5', 'L38', 'L36', 'L13', 'K1', 'K4', 'K5', 'W7'], ['W7', 'L17', 'W8', 'K35', 'K36', 'W31', 'K13', 'W30'], ['W30', 'K13', 'K24', 'K23', 'K22', 'K21', 'W27'], ['W27', 'K21', 'K22', 'K20', 'K15', 'W30', 'K12', 'W35', 'K5', 'K4', 'K1', 'L13', 'L6', 'L5', 'L4', 'L3', 'L2', 'W1']]</p> <p>total jarak : 13.910000000000004 km</p>

Pengujian Ke-	Hasil
27	Masukkan Node Awal:K23 Masukkan Node Tujuan:W10,W9,W11,W12,W14,W15 jalur : [['K23', 'K22', 'K21', 'W27', 'W25', 'K58', 'K54', 'K55', 'K72', 'W21', 'W15'], ['W15', 'W21', 'K71', 'K70', 'K69', 'W14'], ['W14', 'W13', 'B10', 'B9', 'B18', 'B20', 'B22', 'B25', 'W12'], ['W12', 'B25', 'B22', 'B23', 'B26', 'B5', 'B4', 'B38', 'B3', 'W9'], ['W9', 'B3', 'B2', 'W10'], ['W10', 'B1', 'W11']] total jarak : 8.61 km
28	Masukkan Node Awal:B20 Masukkan Node Tujuan:W21,W13,W18,W26,W27,W30 jalur : [['B20', 'B18', 'B9', 'B10', 'W13'], ['W13', 'W14', 'K69', 'K70', 'K71', 'W21'], ['W21', 'K72', 'W20', 'W28', 'K55', 'K56', 'K57', 'K58', 'K54', 'K53', 'W26'], ['W26', 'W25', 'K58', 'K54', 'K53', 'K52', 'K41', 'W27'], ['W27', 'K21', 'K22', 'K20', 'K15', 'W30'], ['W30', 'K13', 'K24', 'K23', 'K22', 'K21', 'W27', 'W25', 'K58', 'K54', 'K55', 'K72', 'B14', 'D17', 'D16', 'D15', 'D14', 'W18']] total jarak : 10.25 km
29	Masukkan Node Awal:S6 Masukkan Node Tujuan:W17,W18,W20,W31,W32,W33 jalur : [['S6', 'S7', 'K49', 'K48', 'W32'], ['W32', 'K48', 'K47', 'S9', 'S10', 'W20'], ['W20', 'K72', 'W21', 'K71', 'K66', 'K67', 'K59', 'K60', 'W38', 'K62', 'W33'], ['W33', 'K28', 'K27', 'K30', 'W37', 'K32', 'K36', 'W31'], ['W31', 'K29', 'K25', 'K24', 'K23', 'K22', 'K21', 'W27', 'W25', 'K58', 'K54', 'K55', 'K72', 'B14', 'D17', 'D16', 'D15', 'D14', 'W18'], ['W18', 'D14', 'D13', 'W17']] total jarak : 12.149999999999999 km
30	Masukkan Node Awal:D6 Masukkan Node Tujuan:W27,W26,W28,W11,W21,W23 jalur : [['D6', 'B15', 'B13', 'K70', 'K71', 'W21'], ['W21', 'K72', 'W20', 'W28'], ['W28', 'K55', 'K56', 'K57', 'K58', 'K54', 'K53', 'W26'], ['W26', 'W25', 'W24', 'W23'], ['W23', 'W24', 'W25', 'K58', 'K54', 'K53', 'K52', 'K41', 'W27'], ['W27', 'K21', 'K19', 'K18', 'K17', 'W29', 'K10', 'L57', 'L56', 'L55', 'L54', 'L53', 'B7', 'B6', 'B5', 'B4', 'B38', 'B3', 'B2', 'W10', 'B1', 'W11']] total jarak : 11.14 km

DAFTAR RIWAYAT HIDUP



Riyan Fahmi Syihabuddin atau akrab dipanggil Riyan, lahir di Mojokerto pada tanggal 1 September 1998. Anak bungsu dari 2 bersaudara dari pasangan Bapak Abdul Kholiq dan Ibu Sofiatun. Seorang penulis artikel di Good News From Indonesia (GNFI) dan seorang yang gemar akan dunia desain grafis.

Pemuda yang berasal dari desa di kaki Gunung Welirang ini selalu bersemangat dalam mengembangkan potensi yang telah dianugerahkan oleh Tuhan. Hal ini dapat terwujud baik dalam Pendidikan formal maupun nonformal yang telah ditempuh. Dimulai dari MI Darussalam Pacet (2005-2011), MTs Pacet (2011-2014), MA Pacet (2014-2017), sampai UIN Maulana Malik Ibrahim Malang. Penulis juga mengenyam beberapa Pendidikan nonformal diantaranya TPQ Darussalam Pacet (2002-2010), Madin Al Khusaini Pacet (2010 – 2017), Eagle Englis Course (2013), Ma'had Sunan Ampel Al Aly (2017-2018), dan lain sebagainya.

Selama menempuh pendidikan dikuliah penulis juga mempunyai beberapa prestasi. Prestasi tersebut diantaranya adalah Medali Emas OSKI (Olimpiade Sains dan Karya Inovasi) PTKI bidang Matematika 2019 di Makasar, Medali Perak OSM (Olimpiade Sains Mahasiswa) POSI 2020, Top 7 Gus dan Yuk Kabupaten Mojokerto 2021, dan Pemenang III Duta Batik Kabupaten Mojokerto 2021. Selain itu penulis juga pernah menjadi asisten praktikum pemograman computer dan juga pernah aktif dalam kegiatan HMJ “Integral” Matematika bidang Penerbitan dan *Jurnalistik*, dan juga aktif di beberapa komunitas seperti MEC (Mathematics English Clubs) dan AlFarazy



KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI MAULANA MALIK
IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jalan Gajayana No 50 Dinoyo Malang Telp/Fax. (0341)588933

BUKTI KONSULTASI SKRIPSI

Nama : Riyan Fahmi Syihabuddin
NIM : 17610066
Fakultas/ Program Studi : Sains dan Teknologi/ Matematika
Judul Skripsi : Implementasi Algoritma A-Star dalam Menentukan Rute Terpendek Destinasi Wisata Kota Malang
Pembimbing I : Mohammad Nafie Jauhari, M.Si
Pembimbing II : Muhammad Khudzaifah, M.Si

No	Tanggal	Hal	Tanda Tangan
1	5 Februari 2021	Konsultasi Bab I dan II	1.
2	18 Februari 2021	Konsultasi Bab III	2.
3	3 Maret 2021	Konsultasi Kajian Keislaman	3.
4	12 Maret 2021	Konsultasi Bab I, II, dan III	4.
5	20 Maret 2021	Konsultasi PPT Seminar Prposal	5.
6	16 April 2021	Konsultasi Simulasi Seminar Proposal	6.
7	28 April 2021	Konsultasi Revisi Bab I, II, dan III	7.
8	24 Mei 2021	Konsultasi Revisi Kajian Keislaman	8.
9	10 Juni 2021	Konsultasi Gambaran Bab IV	9.
10	2 Juli 2021	Konsultasi Tentang Data Penelitian	10.
11	3 September 2021	Konsultasi Pemograman	11.
12	3 Oktober 2021	Konsultasi Revisi Pemograman	12.
13	29 Oktober 2021	Konsultasi Pendeskripsian Hasil Pemograman	13.
14	21 November 2021	Konsultasi Bab IV dan V	14.
15	28 November 2021	Konsultasi Revisi Bab IV dan V	15.
16	6 Desember 2021	ACC Keseluruhan	16.

Malang, 6 Desember 2021
Mengetahui
Ketua Program Studi



Dr. Elly Susanti, M.Sc.
NIP. 19741129 200012 2 005