

**OPTIMASI RUTE WISATA KABUPATEN MOJOKERTO
MENGUNAKAN ALGORITMA A-STAR UNTUK
PENENTUAN RUTE TERPENDEK**

SKRIPSI

**OLEH
SILVIA ROHMADINA
NIM. 200601110038**



**JURUSAN MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2024**

**OPTIMASI RUTE WISATA KABUPATEN MOJOKERTO
MENGUNAKAN ALGORITMA A-STAR UNTUK
PENENTUAN RUTE TERPENDEK**

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
Silvia Rohmadina
NIM. 200601110038**

**JURUSAN MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2024**

**OPTIMASI RUTE WISATA KABUPATEN MOJOKERTO
MENGUNAKAN ALGORITMA A-STAR UNTUK
PENENTUAN RUTE TERPENDEK**

SKRIPSI

**Oleh
Silvia Rohmadina
NIM. 200601110038**

**Telah Disetujui Untuk Diuji
Malang, 9 Desember 2024**

Dosen Pembimbing I



**Juhari, M.Si
NIPPPK. 19840209 202321 1 010**

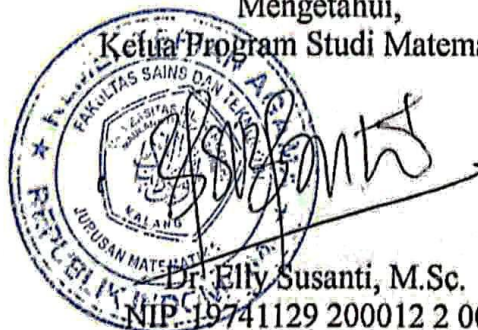
Dosen Pembimbing II



**Evawati Alisah, M.Pd
NIP. 19720604 199903 2 001**

Mengetahui,

Ketua Program Studi Matematika



Dr. Elly Susanti, M.Sc.

NIP. 19741129 200012 2 005

OPTIMASI RUTE WISATA KABUPATEN MOJOKERTO MENGUNAKAN ALGORITMA A-STAR UNTUK PENENTUAN RUTE TERPENDEK

SKRIPSI

Oleh
Silvia Rohmadina
NIM. 200601110038

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

Tanggal, 23 Desember 2024

Ketua Penguji : Dr. Usman Pagalay, M.Si.

Anggota Penguji I : Mohammad Jamhuri, M.Si.

Anggota Penguji II : Juhari, M.Si.

Anggota Penguji III : Evawati Alisah, M.Pd.

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 : Silvia Rohmadina
NIM : 200601110038
Program Studi : Matematika
Fakultas : Sains dan Teknologi
Judul Skripsi : Optimasi Rute Wisata Kabupaten Mojokerto Menggunakan
Algoritma *A-Star* Untuk Penentuan Rute Terpendek

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

Malang, 23 Desember 2024

Yang membuat pernyataan,



Silvia Rohmadina

NIM. 200601110038

MOTO

“Setetes keringat Ibuku yang keluar, terdapat seribu langkahku untuk maju.”

"Setiap peristiwa dalam hidup adalah bagian dari takdir Allah; percayalah, semuanya memiliki hikmah."

"Letakan aku dalam hatimu, maka aku akan meletakanmu dalam hatiku."
(Q.S Al-Baqarah:152)

“Aku belum pernah kecewa dalam berdoa kepada-Mu, wahai Tuhanku.”
(QS. Maryam: 4)

PERSEMBAHAN

Bismillaahirrahmaanirrahiim

Segala puji dan syukur tiada henti penulis ucapkan kepada Allah SWT, atas rahmat, petunjuk, serta bantuan dan mempermudah penulis dalam menyelesaikan setiap tahap skripsi ini. Dengan sepenuh hati, penulis mempersembahkan skripsi ini kepada:

Dua orang yang sangat berpengaruh dalam hidup penulis, yaitu Bapak Solikin dan Ibu Fitriyah, yang telah banyak berkorban serta memberikan nasihat dan doa restu yang tulus, sehingga penulis dapat menyelesaikan skripsi ini dengan baik. Penulis juga ingin mengapresiasi diri sendiri yang telah bertahan dan tidak menyerah dalam perjalanan ini, serta tetap percaya pada rencana indah dari Allah SWT. Tak lupa, penulis menyampaikan rasa terima kasih kepada teman-teman yang dengan ikhlas memberikan dukungan, semangat, dan doa dalam proses penyelesaian skripsi ini. Semoga semua usaha dan dukungan ini mendatangkan berkah bagi kita semua.

KATA PENGANTAR

Alhamdulillah, puji dan sukur atas kehadiran Allah Swt. Yang telah memberikan Rahmat dan hidayah-nya sehingga penulis dapat menyelesaikan penulisan skripsi yang berjudul “Optimasi Rute Wisata Kabupaten Mojokerto Menggunakan Algoritma *A-Star* untuk Mencari Rute Terpendek”. Sholawat dan salam senantiasa tetap tercurahkan kepada nabi Muhammad SAW, yang telah menuntun manusia menuju jalan kebenaran, yaitu agama islam dan kita mengharapkan syafaatnya kelak.

Dalam penyusunan skripsi ini, penulis menyadari sepenuhnya bahwa selesainya skripsi ini banyak mendapat bimbingan, arahan, dan sumbangan pemikiran dari berbagai pihak. Oleh karena itu penulis menyampaikan banyak terima kasih kepada:

1. Bapak Prof. Dr. H. M. Zainuddin, M.A., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim.
2. Ibu Prof. Dr. Hj. Sri Harini, M.Si., selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim.
3. Ibu Dr. Elly Susanti, S.Pd., M.Sc., selaku ketua Program Studi Matematika, Universitas Islam Negeri Maulana Malik Ibrahim.
4. Bapak Juhari, M.Si., selaku dosen pembimbing I yang telah banyak memberikan arahan, nasihat, motivasi, dan pengalaman berharga kepada penulis.
5. Ibu Evawati Alisah, M.Pd., selaku dosen pembimbing II yang telah banyak memberikan arahan dan berbagi ilmunya kepada penulis.
6. Bapak Dr. Usman Pagalay, M.Si., selaku ketua penguji yang telah memberikan saran dan dukungan kepada penulis.
7. Bapak Mohammad Jamhuri, M.Si., selaku dosen penguji 1 yang telah memberikan saran dan dukungan kepada penulis.
8. Ibu Intan Nisfulaila, M.Si., selaku dosen wali yang telah banyak memberikan bimbingan dan arahan semasa penulis menempuh perkuliahan.
9. Seluruh dosen Program Studi Matematika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim.

10. Segenap sivitas akademika Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
11. Kedua orang tua tercinta, Bapak Solikin dan Ibu Fitriyah, Mbah tercinta, Mbah Martinah, serta Kakak dan Ponakan tercinta, Muhammad Ainul Yakin, Riana Lestari, dan Arsakha Virendra Shafwan, serta seluruh keluarga besar tercinta yang senantiasa memberikan do'a, dukungan dan kasih sayang yang terbaik untuk penulis.
12. Seseorang terdekat penulis Andhika Surya Pratama, terimakasih karena selalu membantu, mendukung, memberikan apresiasi, serta menjaga penulis selama berada di Malang.
13. Sahabat terbaik penulis, Aini, Zakia, Salsa, Diasnita, Sisil, Silvia, Sephia, yang sudah selalu ada dan membantu dalam setiap proses penulis dalam menyusun skripsi, serta memberikan dukungan dan apresiasi kepada penulis.
14. Teman-teman angkatan program studi matematika 2020 yang berbagi pengalaman selama menuntut ilmu dibangku perkuliahan
15. Semua pihak yang tidak dapat disebutkan satu-persatu yang telah memberikan bantuan kepada penulis dalam menyelesaikan skripsi ini baik moril maupun materil.

Malang, 23 Desember 2024

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTO	vi
PERSEMBAHAN.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
DAFTAR SIMBOL	xiv
DAFTAR LAMPIRAN	xv
ABSTRAK	xvi
ABSTRACT	xvii
مستخلص البحث.....	xviii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	6
1.3 Tujuan	6
1.4 Manfaat Penelitian	6
1.5 Batasan Masalah	8
BAB II KAJIAN PUSTAKA	9
2.1 Teori Pendukung	9
2.1.1 Definisi Graf	9
2.1.2 Jenis-jenis Graf	9
2.1.3 Lintasan Terpendek	11
2.1.4 Algoritma <i>A-Star</i>	13
2.1.5 Kabupaten Mojokerto	19
2.2 Kajian Integrasi dengan Al-Quran dan Hadits	22
2.3 Kajian penerapan Algoritma <i>A-Star</i> dengan Teori Pendukung.....	27
BAB III METODE PENELITIAN	29
3.1 Jenis Penelitian.....	29
3.2 Sumber dan Data Penelitian.....	29
3.3 Tahap Penelitian.....	30
BAB IV HASIL DAN PEMBAHASAN.....	37
4.1 Deskripsi Data.....	37
4.2 Pengujian Pencarian Rute Terpendek	39
4.3 Hasil Pengujian	93
4.4 Evaluasi Pengujian.....	98
4.5 Implementasi Algoritma <i>A-Star</i> dalam Pandangan Islam.....	101
BAB V PENUTUP	104
5.1 Kesimpulan	104
5.2 Saran Untuk Penelitian Lanjutan	105
DAFTAR PUSTAKA.....	106

LAMPIRAN.....	109
RIWAYAT HIDUP.....	125

DAFTAR TABEL

Tabel 4.1 Tabel Destinasi Wisata Kabupaten Mojokerto.....	37
Tabel 4.2 Tabel Titik Awal.....	39
Tabel 4.3 Tabel Titik Tujuan Wisata Kabupaten Mojokerto.....	40
Tabel 4.4 Tabel Hasil Perbandingan dengan Algoritma Lainnya	93
Tabel 4.5 Tabel Hasil Proses Pengujian Rute Terpendek	97
Tabel 4.6 Tabel Hasil Nilai Efektifitas	99

DAFTAR GAMBAR

Gambar 2.1	Contoh dari Graf Sederhana	10
Gambar 2.2	Contoh dari Graf Tak Sederhana (Graf Ganda dan Graf Semu)	10
Gambar 2.3	Contoh dari Graf Tak Berarah	11
Gambar 2.4	Contoh dari Graf Berarah	11
Gambar 2.5	Contoh Tahapan dalam Pencarian <i>A-Star</i>	16
Gambar 4.1	Screenshoot data Lokasi dan Jalur Kabupaten Mojokerto	38
Gambar 4.2	Visualisasi Rute P1 Menuju W1	45
Gambar 4.3	Visualisasi Rute P1 Menuju W2	47
Gambar 4.4	Visualisasi Rute P1 Menuju W3	50
Gambar 4.5	Visualisasi Rute W1 Menuju W2	52
Gambar 4.6	Visualisasi Rute W1 Menuju W3	54
Gambar 4.7	Visualisasi Rute W3 Menuju W2	56
Gambar 4.8	Visualisasi Rute P2 Menuju W8	58
Gambar 4.9	Visualisasi Rute P2 Menuju W4	62
Gambar 4.10	Visualisasi Rute P2 Menuju W5	65
Gambar 4.11	Visualisasi Rute W8 Menuju W5	69
Gambar 4.12	Visualisasi Rute W8 Menuju W4	72
Gambar 4.13	Visualisasi Rute W5 Menuju W4	74
Gambar 4.14	Visualisasi Rute P6 Menuju W6	76
Gambar 4.15	Visualisasi Rute P6 Menuju W5	79
Gambar 4.16	Visualisasi Rute P6 Menuju W7	81
Gambar 4.17	Visualisasi Rute W5 Menuju W6	83
Gambar 4.18	Visualisasi Rute W5 Menuju W7	84
Gambar 4.19	Visualisasi Rute W6 Menuju W7	87
Gambar 4.20	Hasil Visualisasi Rute Terpendek	92
Gambar 4.21	Hasil Perhitungan Rute Penelitian di Google Maps	96
Gambar 4.22	Hasil Perhitungan Rute Pembandingan di Google Maps	97

DAFTAR SIMBOL

$f(n)$	: Biaya Solusi Termurah Melalui n ke tujuan
$g(n)$	: Biaya Jalur dari Titik Awal ke n
$h(n)$	: Biaya Jalur Termurah dari n ke Tujuan
n	: Node / Titik
$\Delta\phi$	: Perbedaan Lintang Antara Dua Titik (Radian)
$\Delta\lambda$	: Perbedaan Bujur Antara Dua Titik (Radian)
a	: Variabel Intermediate yang Menghitung Jarak Kuadrat dari Dua Titik
ϕ_1	: <i>Latitude</i> / Garis Lintang dari Titik Pertama
ϕ_2	: <i>Latitude</i> / Garis Lintang dari Titik Kedua
λ_1	: <i>Longitude</i> / Garis Bujur dari Titik Pertama
λ_2	: <i>Longitude</i> / Garis Bujur dari Titik Kedua
c	: Jarak Angular (Radian)
d	: Jarak Antara Dua Titik di Permukaan Bumi (KM)
R	: Jari-jari Bumi (6.371 KM)
$P1, P2, P3, \dots, P10$	: Titik Awal Perhitungan <i>A-Star</i>
$C1, C2, C3, \dots, C99$	: Titik Persimpangan Jalan
$W1, W2, W3, \dots, W27$	: Titik Tujuan Wisata

DAFTAR LAMPIRAN

Lampiran 1. Data Jarak Antar Titik	109
Lampiran 2. Data Heuristik Menggunakan Formula Haversine	111
Lampiran 3. Data Titik Awal	115
Lampiran 4. Script Code	115
Lampiran 5. Hasil Visualisasi Rute Terpendek	118

ABSTRAK

Rohmadina, Silvia. 2024. **Optimasi Rute Wisata Kabupaten Mojokerto Menggunakan Algoritma *A-Star* untuk Penentuan Rute Terpendek**. Skripsi. Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (1) Juhari, M.Si. (2) Evawati Alisah, M.Pd.

Kata Kunci: Algoritma *A-Star*, Nilai Heuristik, Formula *Haversine*, rute terpendek.

Berwisata menjadi pilihan populer untuk mengisi waktu liburan sambil menikmati keindahan alam dan mengunjungi tempat-tempat bersejarah. Kabupaten Mojokerto, yang terletak di Jawa Timur, menjadi salah satu destinasi favorit karena kaya akan wisata budaya. Dengan lebih dari 25 destinasi wisata alam dan sejarah, wisatawan sering kali ingin mengunjungi beberapa tempat sekaligus dalam satu perjalanan. Oleh karena itu, diperlukan metode yang membantu merencanakan rute kunjungan agar lebih efisien. Penelitian ini memanfaatkan Algoritma *A-Star* untuk menemukan rute terpendek antara destinasi-destinasi wisata, dengan mengandalkan perhitungan nilai heuristik yang diperoleh dari jarak antar titik menggunakan formula *Haversine*. Prosesnya dimulai dengan pemilihan titik awal perjalanan, dilanjut menentukan urutan destinasi yang menghasilkan rute terpendek menggunakan algoritma *A-Star*. Proses ini terus berlanjut hingga semua destinasi yang diinginkan terkunjungi. Efektivitas rute yang dihasilkan kemudian diuji terhadap rute yang disediakan oleh Google Maps. Dari 10 uji coba rute wisata, Algoritma *A-Star* menghasilkan rute rata-rata 32,1% lebih pendek. Pengujian juga menunjukkan bahwa *A-Star* mengungguli metode *Greedy* dan *Dijkstra* dalam menemukan rute terpendek. Algoritma ini diharapkan membantu wisatawan merancang perjalanan yang lebih efisien, menghemat waktu dan energi.

ABSTRACT

Rohmadina, Silvia. 2024. **Optimization of Tourist Routes in Mojokerto Regency Using the *A-Star* Algorithm for Determining the Shortest Path.** Thesis. Mathematics Study Program, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Advisors: (1) Juhari, M.Si. (2) Evawati Alisah, M.Pd.

Keywords: *A-Star* Algorithm, Heuristic Value, Haversine Formula, Shortest Path.

Traveling is a popular choice to fill your vacation time while enjoying the beauty of nature and visiting historical places. Mojokerto Regency, located in East Java, is one of the favorite destinations for its rich cultural tourism. With more than 25 natural and historical tourist destinations, tourists often want to visit several places in one trip. Therefore, there is a need for a method that helps plan visit routes more efficiently. This research utilizes the A-Star Algorithm to find the shortest route between tourist destinations, relying on the calculation of the heuristic value obtained from the distance between points using the haversine formula. The process begins with the selection of the starting point of the trip, followed by determining the sequence of destinations that produce the shortest route using the A-Star algorithm. This process continues until all desired destinations are visited. The effectiveness of the resulting route is then tested against the route provided by Google Maps. From 10 trials of tourist routes, the A-Star algorithm produces an average route that is 32.1% shorter. The test also shows that A-Star outperforms the Greedy and Dijkstra methods in finding the shortest route. This algorithm is expected to help tourists design more efficient trips, saving time and energy.

مستخلص البحث

روهمادينا، سيلفيا. ٢٠٢٤. تحسين الطرق السياحية في موجوكرتو ريجنسي باستخدام خوارزمية *A-Star* لتحديد أقصر طريق. البحث العلمي. قسم الرياضيات، كلية العلوم والتكنولوجيا، جامعة مولان مالك إبراهيم الإسلامية احلكومية يف مالنح. المشرف الأول: جوهري، الماجستير. المشرفة الثانية: إيفاواقي أليسا، الماجستير.

الكلمات المفتاحية: خوارزمية *A-Star*، القيمة الإرشادية، صيغة هافيرسين، أقصر طريق.

يُعد السفر خياراً شائعاً لملء وقت عطلتك أثناء الاستمتاع بجمال الطبيعة وزيارة الأماكن التاريخية. تُعد محافظة موجوكرتو، الواقعة في جاوة الشرقية، إحدى الوجهات المفضلة بسبب غنى معالمها الثقافية. مع وجود أكثر من ٢٥ وجهة سياحية طبيعية وتاريخية، غالباً ما يرغب المسافرون في زيارة عدة أماكن في رحلة واحدة. لذلك، هناك حاجة إلى طريقة تساعد في تخطيط مسارات الزيارة بشكل أكثر كفاءة. يستخدم هذا البحث خوارزمية *A-Star* لإيجاد أقصر طريق بين الوجهات السياحية، بالاعتماد على حساب القيمة الاستدلالية التي يتم الحصول عليها من المسافة بين النقاط باستخدام صيغة هافيرسين. وتبدأ العملية باختيار نقطة بداية الرحلة، يليها تحديد تسلسل الوجهات التي تنتج أقصر طريق باستخدام خوارزمية *A-Star*. تستمر هذه العملية حتى تتم زيارة جميع الوجهات المطلوبة. ثم يتم اختبار فعالية المسار الناتج مقابل المسار الذي توفره خرائط *Google*. من خلال ١٠ تجارب للطرق السياحية، أنتجت خوارزمية *A-Star* مساراً أقصر بنسبة ٣٢,١٪ في المتوسط. أظهرت الاختبارات أيضاً أن خوارزمية *A-Star* تفوقت على طريقتي *Dijkstra* و *Greedy* في إيجاد أقصر طريق. من المتوقع أن تساعد هذه الخوارزمية المسافرين على تصميم رحلات أكثر كفاءة، مما يوفر الوقت والطاقة.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kabupaten Mojokerto terletak di daerah yang bervariasi topografinya, mulai dari dataran rendah di sekitar sungai Brantas hingga dataran tinggi di lereng Gunung Penanggungan dan Gunung Welirang. Secara historis, daerah ini merupakan bagian penting dari Kerajaan Majapahit dan Kerajaan Kahuripan. Peninggalan sejarah dari masa Majapahit tersebar di hampir seluruh wilayah Mojokerto, sementara peninggalan dari masa Airlangga banyak ditemukan di lereng Gunung Penanggungan. Kabupaten Mojokerto memiliki potensi pariwisata yang sangat beragam, termasuk wisata alam, sejarah, dan wisata buatan, serta memiliki kekayaan budaya, kesenian, dan tradisi lokal yang bervariasi. Pengembangan pariwisata dan kebudayaan di Mojokerto diyakini tidak hanya mampu meningkatkan kesejahteraan masyarakat setempat, tetapi juga memperkenalkan keragaman wisata dan budaya kepada khalayak luas, serta melestarikan kebudayaan yang berkembang di wilayah tersebut untuk generasi mendatang (Taufik & Wandini, 2012). Dengan daya tarik unik yang dimilikinya, Mojokerto menjadi pilihan yang sempurna bagi para wisatawan yang ingin menjelajahi keindahan alam dan keberagaman budaya Indonesia dalam atmosfer yang dipenuhi pesona dan ketenangan.

Pencarian jalur terpendek ketika sedang dalam berkunjung ke suatu daerah wisata adalah hal yang sangat penting untuk dipertimbangkan ketika merencanakan

destinasi wisata. Sebagaimana yang telah difirmankan Allah SWT pada surah Surah Al-Kahfi(18) ayat 19 yang berbunyi (Kemenag, 2024):

وَكَذَلِكَ بَعَثْنَاهُمْ لِيَتَسَاءَلُوا بَيْنَهُمْ قَالَ قَائِلٌ مِّنْهُمْ كَمْ لَبِثْتُمْ قَالُوا لَبِثْنَا يَوْمًا أَوْ بَعْضَ يَوْمٍ قَالُوا رَبُّكُمْ
أَعْلَمُ بِمَا لَبِثْتُمْ فَابْعَثُوا أَحَدَكُمْ بِوَرِقِكُمْ هَذِهِ إِلَى الْمَدِينَةِ فَلْيَنْظُرْ أَيُّهَا أَزْكَى طَعَامًا فَلْيَأْتِكُمْ بِرِزْقٍ مِنْهُ وَلْيَتَلَطَّفْ
وَلَا يُشْعِرَنَّ بِكُمْ أَحَدًا

Artinya:

"Dan demikianlah Kami bangkitkan mereka, agar mereka saling bertanya di antara mereka sendiri. Ada seorang yang berkata: "Berapa lama kamu tinggal (di tempat ini)?" Mereka menjawab: "Kami tinggal sehari atau sebagian hari." Orang itu berkata: "Tuhanmu lebih mengetahui tentang lamanya kamu tinggal, maka kirimkanlah seorang di antara kamu dengan uang perakmu ini ke kota, supaya ia mencari makanan yang baik-baik dari padanya dan memperhatikan sopan-santunnya, dan jangan sekali-kali ia memberitahukan tentang kamu kepada seorang pun."

Tafsir Jalalin menjelaskan bagaimana Ashabul Kahfi, yang telah ditidurkan dalam gua oleh Allah, dibangun kembali untuk saling bertanya tentang berapa lama mereka telah tinggal di sana. Ketika mereka terbangun, beberapa dari mereka menduga bahwa mereka hanya tinggal sehari atau setengah hari, sementara yang lain menyerahkan pengetahuan tersebut kepada Allah. Kemudian, mereka mengutus salah satu dari mereka untuk membeli makanan dengan uang perak yang mereka miliki, dengan syarat untuk tidak mengungkapkan keberadaan mereka kepada siapapun (Tafsir, 2024). Ayat diatas memberikan pelajaran yang dalam tentang perencanaan perjalanan yang bijak dan strategis. Menyajikan kisah tentang sekelompok orang yang melakukan perjalanan dan bertanya kepada sesama mereka tentang berapa lama mereka telah berada di suatu tempat. Salah seorang dari mereka menyebutkan bahwa mereka telah tinggal di tempat tersebut selama sehari atau sebagian hari. Namun, seseorang di antara mereka menyarankan agar mereka mengirimkan seorang perwakilan dengan uang perak mereka ke kota untuk mencari

makanan yang baik dan menyelidiki keadaan di sana tanpa menarik perhatian orang lain. Meskipun ayat ini tidak secara eksplisit menyebutkan tentang mencari rute terpendek, namun pesan yang terkandung di dalamnya adalah pentingnya perencanaan yang matang dalam perjalanan. Dengan mengoptimalkan sumber daya yang tersedia, termasuk waktu dan uang, serta memilih jalur atau rute yang paling efisien, mereka dapat mencapai tujuan mereka dengan lebih baik.

Hal ini menunjukkan betapa pentingnya bagi setiap individu untuk mempertimbangkan dengan cermat setiap langkah yang diambil dalam perjalanan mereka, baik secara fisik maupun spiritual. Dengan demikian, ayat ini mengajarkan nilai-nilai tentang kebijaksanaan, perencanaan yang matang, dan pengelolaan sumber daya dengan efektif, yang merupakan prinsip-prinsip yang dapat diterapkan dalam berbagai aspek kehidupan, termasuk dalam konteks perjalanan fisik maupun perjalanan hidup secara keseluruhan.

Penentuan akses atau rute perjalanan merupakan persoalan yang secara erat terhubung dengan konsep Teori Graf. Teori Graf adalah himpunan dari titik-titik $V(G)$ dan himpunan sisi $E(G)$ yang menghubungkan beberapa titik, yang sering disebut sebagai edge (Munir, 2016). Hubungan penelitian dalam teori ini dengan destinasi wisata Kabupaten Mojokerto terkait dengan pencarian rute yang paling efisien untuk mengunjungi berbagai objek wisata di wilayah tersebut. Berbagai algoritma dapat digunakan untuk menentukan jalur terpendek, di antaranya adalah algoritma *A-Star*, *Dijkstra*, *Greedy*, dan yang lainnya (Wayahdi, Ginting, & dkk, 2021).

Algoritma *A-Star*, menurut penelitian (Martins & dkk, 2022), merupakan pengembangan dari algoritma *Dijkstra* yang menggunakan heuristik berdasarkan

pengetahuan tentang masalah yang dihadapi. Namun, *A-Star* berbeda dari *Dijkstra* karena fungsi heuristik $h(n)$ ditambahkan ke fungsi biaya jalur $g(n)$ untuk mendefinisikan fungsi baru, yaitu $f(n)$, fungsi biaya jalur perkiraan. Sementara itu, penelitian (Hartatik, Rahim, & dkk, 2023) menunjukkan bahwa algoritma greedy dapat digunakan untuk menentukan rute terdekat pada setiap langkah hingga mencapai tujuan, dengan mencari nilai minimum sementara. Meskipun tidak selalu menghasilkan solusi terbaik, algoritma greedy umumnya memiliki kompleksitas waktu yang baik, dalam hal ini terdapat perbedaan dengan penelitian yang penulis gunakan, dikarenakan penelitian penulis menggunakan algoritma A^* dengan pendekatan Haversine formula, yang memungkinkan pencarian rute terpendek secara lebih menyeluruh dengan mempertimbangkan jarak ke tujuan dan biaya dari titik awal. Sementara itu, penelitian Hartatik, dkk mengandalkan algoritma greedy yang memprioritaskan pilihan rute terdekat di setiap langkah. Meskipun metode tersebut memiliki keunggulan tersendiri, pendekatannya lebih fokus pada efisiensi lokal dibandingkan keseluruhan perjalanan. Di sisi lain, Penelitian penulis menggunakan algoritma *A-Star* yang dilengkapi dengan heuristik *Haversine* formula, memungkinkan pencarian rute terpendek secara lebih efisien dengan mempertimbangkan estimasi jarak langsung ke tujuan. Pendekatan ini membantu dalam proses optimasi rute wisata di Kabupaten Mojokerto, khususnya dengan fokus pada keakuratan dan kecepatan dalam pencarian rute.

Sebaliknya, penelitian (Bunaen, Pratiwi, & Riti, 2022) menggunakan algoritma *Dijkstra*, yang secara efektif menyelesaikan masalah pencarian rute terpendek dengan membandingkan setiap vertex untuk menemukan jalur terpendek dari awal hingga tujuan. Meskipun *Dijkstra* sangat baik untuk mencari rute tanpa

memperhitungkan jarak estimasi ke tujuan, algoritma *A-Star* memberikan keuntungan tambahan dengan memadukan informasi jarak langsung, sehingga lebih cocok untuk aplikasi yang membutuhkan hasil lebih cepat di lingkungan wisata yang luas.

Selanjutnya pada penelitian yang dilakukan oleh (Sinlae, Nuraini, & dkk, 2023) menunjukkan bahwa Hasil pengujian algoritma *A-Star* lebih unggul dibandingkan dengan algoritma greedy. Keunggulan *A-Star* ini didasarkan pada pertimbangan nilai heuristik yang ada di setiap node, sehingga *A-Star* lebih efektif dalam menentukan jalur yang akan dilalui. Meskipun penelitian Penulis dan penelitian Sinlae, dkk sama-sama menggunakan algoritma *A-Star*, perbedaannya terletak pada konteks aplikasinya, di mana penulis mengkhususkan penerapan pada rute wisata dengan perhitungan jarak menggunakan Haversine formula, sedangkan penelitian Sinlae menggunakan *Heuristic Euclidean Distance*.

Dalam konteks dunia nyata pariwisata di Kabupaten Mojokerto, masalah aksesibilitas dan kurangnya promosi di Trowulan menyebabkan rendahnya kunjungan wisatawan dibandingkan dengan daerah Pacet dan Trawas (Suwandi, Haryati, & Suroso, 2023). Oleh karena itu, untuk menemukan cara yang paling efektif merata dan mempermudah dalam mengunjungi tempat-tempat wisata yang berada di Kabupaten Mojokerto, penelitian ini menggunakan algoritma *A-Star* untuk menghitung rute terpendek dengan penggunaan waktu yang efisien. Pemilihan metode ini didasarkan pada kemampuannya untuk mengidentifikasi jalur terpendek dan menganalisis semua opsi lintasan yang efisien dalam sebuah graf yang dibuat berdasarkan tiap tepi dari seluruh simpul yang dibuat. Karenanya, metode ini dianggap cocok untuk menyelesaikan masalah penentuan jalur

terpendek, khususnya dalam konteks pencarian lokasi wisata di Kabupaten Mojokerto.

1.2 Rumusan Masalah

Berdasarkan permasalahan diatas, rumusan masalah yang diangkat dalam penelitian ini adalah bagaimana efektivitas penerapan algoritma *A-Star* dalam mengoptimasi rute terpendek destinasi wisata di Kabupaten Mojokerto?

1.3 Tujuan

Penelitian ini bertujuan untuk mengetahui seberapa efisien algoritma *A-Star* dalam mengoptimasi rute terpendek menuju destinasi wisata di Kabupaten Mojokerto.

1.4 Manfaat Penelitian

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

1. Kepariwisataan:

Penelitian ini dapat membantu dalam pengembangan pariwisata di Kabupaten Mojokerto dengan memberikan rekomendasi rute terpendek ke destinasi wisata. Hal ini dapat meningkatkan daya tarik destinasi dan meningkatkan jumlah kunjungan wisatawan. Selain itu juga, dengan memberikan informasi tentang rute terpendek, wisatawan dapat mengoptimalkan waktu mereka untuk menikmati berbagai objek wisata

yang ada di Kabupaten Mojokerto, meningkatkan pengalaman wisata mereka.

2. Pemerintah:

Penelitian ini dapat membantu pemerintah dalam merencanakan pengelolaan sumber daya pariwisata dengan lebih efisien. Dengan mengetahui rute terpendek, pemerintah dapat mengalokasikan sumber daya secara lebih tepat untuk memperbaiki infrastruktur, mengatur lalu lintas, dan meningkatkan keterjangkauan ke destinasi wisata. Selain itu, penelitian ini dapat memberikan dasar yang kuat bagi pemerintah untuk memprioritaskan pembangunan atau perbaikan infrastruktur yang mendukung keterjangkauan ke destinasi wisata.

3. Wisatawan:

Wisatawan dapat memanfaatkan hasil penelitian ini untuk merencanakan perjalanan mereka dengan lebih baik, memilih rute terpendek dan tercepat untuk mencapai destinasi wisata yang mereka tuju. Selain itu juga, dengan menggunakan rute terpendek, wisatawan dapat menghemat waktu dan biaya perjalanan, sehingga memberikan pengalaman liburan yang lebih efisien dan ekonomis.

4. Ilmiah:

Penelitian ini berperan penting dalam mengembangkan dan menguji Algoritma *A-Star* untuk menentukan rute terpendek dalam pariwisata, memastikan algoritma tersebut efektif dan dapat digunakan dengan baik. Selain memperkaya literatur ilmiah, penelitian ini menguji efektivitas algoritma tersebut di dunia nyata, khususnya di Kabupaten Mojokerto,

memberikan gambaran praktis tentang kinerjanya di berbagai kondisi geografis. Hasilnya juga bisa menjadi referensi bagi penelitian lanjutan dalam optimasi rute di berbagai sektor. Lebih lanjut, penelitian ini membuka peluang pengembangan sistem pendukung keputusan berbasis algoritma yang dapat diterapkan di bidang seperti logistik, transportasi, dan manajemen bencana.

1.5 Batasan Masalah

Agar penyelesaian masalah yang diangkat bisa diselesaikan dengan maksimal, maka diperlukan suatu batasan-batasan pada permasalahan sedang dibahas. Adapun batasan masalah yang ditentukan dalam penelitian ini, yaitu:

1. Destinasi wisata alam dan bersejarah yang menjadi data penelitian terletak Kabupaten Mojokerto, bukan Kota Mojokerto.
2. Rute yang dipilih didasarkan pada jalan utama yang mengarah ke lokasi destinasi wisata Kabupaten Mojokerto.
3. Penelitian ini hanya mempertimbangkan jarak dalam perhitungannya, sedangkan waktu perjalanan diabaikan karena ketidakmampuan untuk memprediksi kemacetan jalan Kabupaten Mojokerto.
4. Data yang digunakan dalam penelitian ini diperoleh melalui Google Maps.

BAB II

KAJIAN PUSTAKA

2.1 Teori Pendukung

2.1.1 Definisi Graf

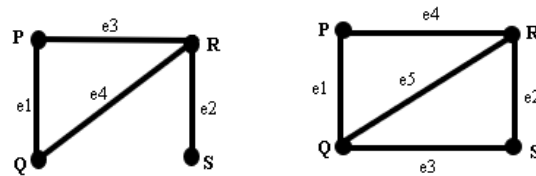
Suatu graf merupakan kumpulan tidak kosong dari titik-titik yang disebut sebagai *vertex*, serta kumpulan pasangan tidak berurutan dari titik-titik tersebut yang dikenal sebagai sisi atau *edge*. Untuk menyimbolkan suatu graf, kita gunakan notasi G . Himpunan titik pada graf G diwakili oleh $V(G)$, sedangkan himpunan sisi pada graf G diwakili oleh $E(G)$. Jika jumlah titik dan sisi pada G bersifat terhingga, maka G disebut sebagai graf terhingga. Ketika terdapat dua sisi atau lebih yang menghubungkan satu pasang titik, fenomena tersebut disebut sebagai sisi rangkap atau *multiple edges*. Istilah yang digunakan untuk menyebut sebuah sisi yang memiliki titik ujung yang sama disebut *loop*. Suatu graf dikategorikan sebagai graf sederhana apabila tidak memiliki sisi rangkap dan loop (Rahayuningsih, 2018).

2.1.2 Jenis-jenis Graf

Graf dapat dibagi menjadi beberapa kategori tergantung pada cara kita melihatnya, seperti apakah terdapat sisi ganda atau kalang (*loop*), jumlah simpul, atau orientasi arah pada sisi (Juhari, 2021). Berdasarkan keberadaan kalang (*loop*) atau sisi ganda, secara umum ada dua jenis graf:

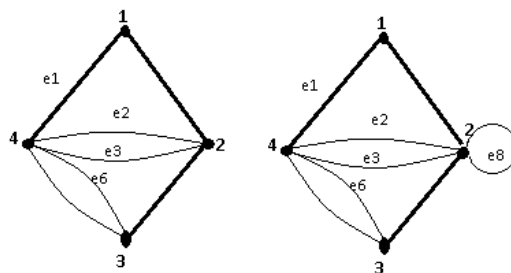
1. Graf sederhana (*simple graph*). Graf ini tidak memiliki kalang (*loop*) atau sisi ganda. Dalam graf sederhana, setiap sisi direpresentasikan oleh pasangan simpul yang tidak ber-urutan. Artinya, jika kita punya sisi (u, v) ,

itu sama dengan (v, u) . Graf sederhana $G = (V, E)$ terdiri dari simpul-simpul yang tidak kosong dan E himpunan pasangan tidak ber-urutan yang berbeda yang disebut sisi.



Gambar 2.1 Contoh dari Graf Sederhana

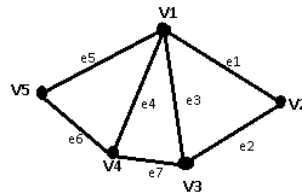
2. Graf tak-sederhana (*unsimple graph*). Graf ini mengandung sisi ganda atau kalang (*loop*). Ada dua jenis graf tak-sederhana: graf ganda dan graf semu. Graf ganda memiliki sisi ganda, yang artinya ada lebih dari satu sisi yang menghubungkan sepasang simpul. Graf ganda $G = (V, E)$ terdiri dari himpunan tidak kosong simpul-simpul serta E himpunan-ganda (multiset) yang mengandung sisi ganda. Sedangkan graf semu mengandung kalang (*loop*), graf ini lebih umum daripada graf ganda dikarenakan sisi graf semu dapat terhubung ke dirinya sendiri. Kita menyebut jumlah simpul dalam graf sebagai kardinalitas graf ($n = |V|$) dan jumlah sisi sebagai ($m = |E|$).



Gambar 2.2 Contoh dari Graf Tak Sederhana (Graf Ganda dan Graf Semu)

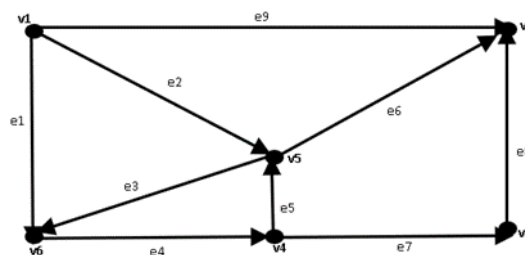
Sisi dalam graf juga bisa memiliki arah. Berdasarkan arahnya, ada dua jenis graf:

1. Graf Tak-Berarah (*undirected graph*). Graf yang sisinya tidak memiliki arah. Dalam graf tak-berarah, urutan simpul yang terhubung oleh sisi tidak diperhitungkan. Jadi, $(u, v) = (v, u)$ dianggap sama.



Gambar 2.3 Contoh dari Graf Tak Berarah

2. Graf berarah (*directed graph*). Graf yang sisinya memiliki arah. Sisi dalam graf berarah biasa disebut busur. Dalam graf berarah, urutan simpul pada busur (u, v) dan (v, u) dianggap berbeda, artinya $(u, v) \neq (v, u)$. Pada busur (u, v) simpul u disebut simpul asal, sedangkan v disebut simpul terminal. Graf berarah sering digunakan untuk menggambarkan aliran proses, peta lalu lintas, dan sebagainya. Kalang (*loop*) diperbolehkan dalam graf berarah, tapi tidak sisi ganda. Definisi graf bisa diperluas untuk mencakup graf-ganda berarah, di mana kalang (*loop*) dan sisi ganda diperbolehkan.



Gambar 2.4 Contoh dari Graf Berarah

2.1.3 Lintasan Terpendek

Lintasan terpendek adalah rute paling minimum yang dibutuhkan untuk mencapai suatu lokasi dari lokasi awal. Rute minimum ini bisa ditemukan dengan

menggunakan graf, yang merupakan kumpulan titik dalam bidang dua dimensi yang saling terhubung melalui edge. Sebuah graf tersusun dari sejumlah titik yang dihubungkan oleh edge. Beberapa istilah yang sering digunakan dalam konteks graf di antaranya:

1. Titik - titik tersebut disebut *verteks*.
2. Garis - garis yang menghubungkan antar *verteks* disebut *edge*.
3. *Adjacent* artinya bertetangga. Maksudnya jika ada dua *vertex* disebut *adjacent*, jika mempunyai *edge* yang sama.
4. *Path* adalah lintasan yang melalui *edge* dan *verteks* dalam graf.
5. *Cycle* adalah lintasan yang dimulai dan berakhir pada *verteks* yang sama.
6. *Direct* pada *directed* graf adalah graf dimana edge-edgenya mempunyai suatu arah.

Graf yang digunakan merupakan graf-graf yang berbobot, yaitu graf yang setiap sisinya diberikan suatu nilai. Ada beberapa macam persoalan lintasan terpendek, antara lain:

1. Lintasan terpendek antara dua buah simpul tertentu (*a pair shortest path*).
2. Lintasan terpendek antara semua pasangan simpul (*all pairs shortest path*).
3. Lintasan terpendek dari *verteks* tertentu ke semua *verteks* yang lain (*single source shortest path*).
4. Lintasan terpendek antara dua buah *verteks* yang melalui beberapa *verteks* tertentu (*intermediate shortest path*) (Mukti & Mulyono, 2018).

Masalah lintasan terpendek merupakan salah satu tantangan mendasar dalam komputasi dan teori graf, yang secara intuitif mudah dipahami dan dijelaskan. Dalam konteks pemetaan, ini mengacu pada masalah mencari rute tercepat dari satu titik ke titik lainnya. Secara formal, dalam sebuah graf yang memiliki simpul-simpul yang terhubung oleh sisi-sisi dengan nilai atau biaya tertentu, tantangannya adalah menemukan jalur dengan biaya paling rendah antara dua simpul tersebut (Talan & Bamnote, 2015).

2.1.4 Algoritma *A-Star*

Algoritma *A-Star* diperkenalkan pertama kali pada tahun 1968 oleh Peter Hart, Nils Nilsson, dan Bertram Raphael. Algoritma ini termasuk dalam kategori pencarian heuristik, yang digunakan untuk menghitung estimasi jalur dengan biaya terkecil dari suatu titik awal ke titik berikutnya hingga mencapai titik tujuan (Marcelina & Yulianti, 2020). Algoritma *A-Star* merupakan salah satu jenis algoritma yang digunakan untuk menyelesaikan kasus yang berhubungan dengan *path finding* (pencarian jalan). Dalam hasil pencariannya *A-Star* dikatakan complete dan optimal. Algoritma *A-Star* menggunakan cara pencarian secara *Breadth First Search*, dimana pencarian dilakukan dengan cara melebar ke setiap node pada level yang sama, dan nantinya akan menemukan rute terbaik dari titik awal sampai tujuan. Algoritma *A-Star* juga dilengkapi dengan suatu fungsi heuristik. Heuristik adalah tahap awal dalam penelitian sejarah yang melibatkan kegiatan pencarian, pengumpulan, dan pengadaan sumber-sumber sejarah secara ekstensif sebagai landasan untuk analisis selanjutnya (Sayono, 2021). Fungsi

heuristik yang terdapat pada algoritma *A-Star* digunakan sebagai optimasi dalam menentukan node tujuan yang akan dipilih.

Algoritma *A-Star* merupakan algoritma *breadth-first search* yang menggunakan fungsi heuristik yang kompleks untuk memilih path. Algoritma *breadth-first search* selalu memperluas jalan untuk menuju node yang akan dilewati, tetapi tidak memperhitungkan cost ke simpul tersebut (Hutabalian, Sunanto, & Amien, 2021).

Algoritma *Breadth First Search* (BFS) sendiri merupakan salah satu algoritma pencarian yang saat ini banyak digunakan. Algoritma BFS bekerja dengan cara melakukan pencarian node secara preorder dalam skala besar. Strategi pencarian algoritma ini diterapkan dengan menjelajahi node dari satu node ke node lainnya, hingga akhirnya mencakup semua node yang menjadi tetangga dari node yang sedang dicari. BFS adalah pendekatan pencarian graf yang digunakan dalam berbagai aplikasi saat ini. Teknik pencarian ini dimulai dari node akar, kemudian berlanjut dengan menelusuri semua node di sekitarnya. Pada proses ini, node-node terdekat yang belum diperiksa akan dicari. Langkah ini diulangi terus-menerus hingga node tujuan akhir ditemukan dan dicapai (Indriyono & Widyatmoko, 2021).

Fungsi heuristik adalah suatu pedoman yang dipakai untuk mendapatkan solusi terkait penelusuran rute. Fungsi heuristik ini berperan dalam menuntun penelusuran rute serta mengevaluasi titik-titik pada peta (Arsyad, Supriyadi, & dll, 2019). Nilai heuristik dapat dimanfaatkan dengan tujuan untuk meminimalisir pencarian ruang. Algoritma *A-Star* dapat mengkontruksi rute tercepat mulai dari tempat permulaan menuju tempat yang diinginkan. Algoritma ini mempunyai formula sebagai berikut (Avini, Alamin, & dkk, 2024):

$$f(n) = h(n) + g(n) \quad (2.1)$$

Keterangan:

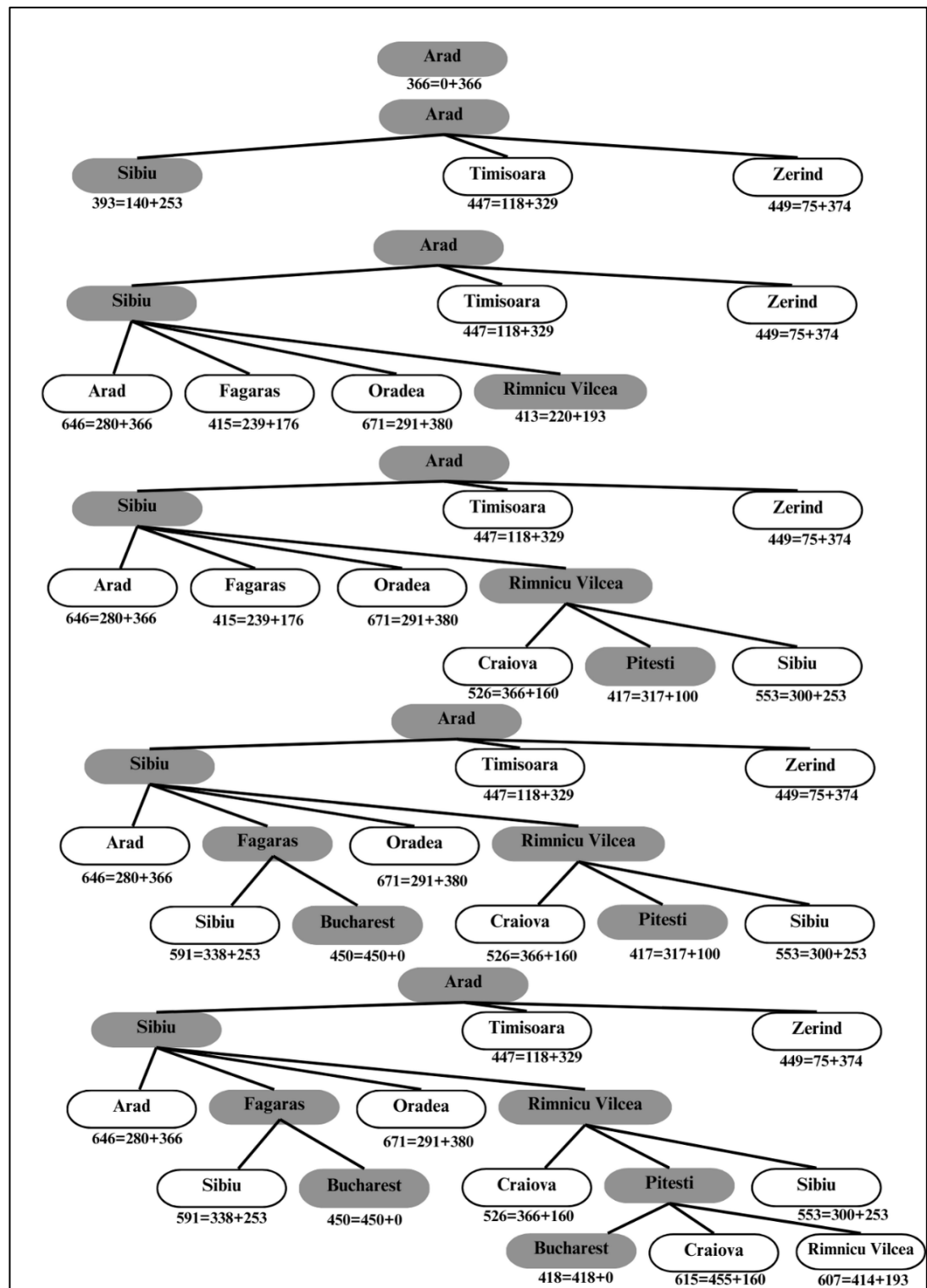
$h(n)$ = Biaya jalur termurah dari titik n ke titik tujuan

$g(n)$ = Biaya jalur dari titik awal ke titik n

$f(n)$ = Biaya solusi termurah melalui titik n ke titik tujuan

n = Titik / node

Tahapan algoritma *A-Star*, mulai dari pemilihan node awal, penentuan nilai g (biaya dari node awal ke node saat ini), h (perkiraan jarak ke tujuan), hingga pemilihan jalur dengan nilai f terendah ($g + h$) untuk menemukan rute terpendek, akan dijelaskan. Proses ini akan digambarkan pada Gambar 2.5.



Gambar 2.5 Contoh Tahapan dalam Pencarian *A-Star*
 Sumber: (Russel & Norvig, 2010).

Dimana $h(n)$ dapat dicari melalui perhitungan dengan formula *haversine*.

Formula *haversine* menghitung jarak antara dua titik atau node berdasarkan

koordinat lintang dan bujurnya. Berikut dibawah ini merupakan rumus dari Formula *haversine* (Shishkov, 2024):

$$\text{Radian} = \text{Derajat} \cdot \left(\frac{\pi}{180}\right)$$

$$\Delta\varphi = (\varphi_2 - \varphi_1)$$

$$\Delta\lambda = (\lambda_2 - \lambda_1)$$

$$a = \sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos \varphi_1 \cdot \cos \varphi_2 \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right)$$

$$c = 2 \cdot \arctan2(\sqrt{a}, \sqrt{1-a})$$

$$d = R \cdot c \tag{2.2}$$

Keterangan:

$\Delta\varphi$ = Perbedaan lintang antara dua titik (radian)

$\Delta\lambda$ = Perbedaan bujur antara dua titik (radian)

φ_1 = *Latitude* / garis lintang dari titik pertama

φ_2 = *Latitude* / garis lintang dari titik kedua

λ_1 = *Longitude* / garis bujur dari titik pertama

λ_2 = *Longitude* / garis bujur dari titik kedua

c = Jarak angular (radian)

d = Jarak antara dua titik dipermukaan bumi (km)

R = Jari-jari bumi (6.371 km)

Formula ini tidak memperhitungkan faktor-faktor seperti ketinggian bukit, kedalaman lembah, dan ketidaksempurnaan bentuk bumi. Oleh karena itu, formula ini berasumsi bahwa permukaan bumi datar dan berbentuk bulat sempurna (Mulya, Trisanto, & dkk, 2022). Berdasarkan penelitian (Miftahuddin, Umaroh, & Karim,

2020), *Haversine* terbukti lebih akurat dibandingkan dengan Euclidean. Dari analisis tersebut, disimpulkan bahwa perhitungan jarak di permukaan bumi harus mempertimbangkan kemiringan bumi.

Semakin tepat nilai heuristik yang dipakai, maka semakin sedikit pula perulangan yang dihitung oleh algoritma *A-Star*, sehingga semakin cepat pula perhitungan bisa dilakukan. Diasumsikan jika nilai heuristik yang ditemukan tepat (selalu mengembalikan jarak rute minimum antara dua node sangat tepat), maka algoritma *A-Star* akan menghasilkan pada hasil perhitungan yang akurat sehingga algoritma bisa menjadi $O(p)$, di mana p adalah banyak jumlah langkah dalam path (rute). Pemilihan nilai heuristik yang akurat atau tidaknya tergantung pada apakah nilai heuristik yang telah di hitung tersebut menjadi terlalu rendah atau terlalu tinggi (Fernando, Mutsaqof, & Megawaty, 2020).

Dalam penerapannya, Algoritma *A-Star* memiliki beberapa terminologi dasar diantaranya *starting point*, simpul (*nodes*), *A*, *open list*, *closed list*, biaya (*cost*), halangan (*unwalkable*).

1. *Starting point* adalah sebuah terminologi untuk posisi awal sebuah benda.
2. *A* adalah simpul yang sedang dijalankan dalam algoritma pencarian jalan terpendek.
3. Simpul adalah petak-petak kecil sebagai representasi dari area pathfinding. Bentuknya dapat berupa persegi, lingkaran, maupun segitiga.
4. *Open list* adalah tempat menyimpan data simpul yang mungkin diakses dari *starting point* maupun simpul yang sedang dijalankan.
5. *Closed list* adalah tempat menyimpan data simpul sebelum *A* yang juga merupakan bagian dari jalur terpendek yang telah berhasil didapatkan.

6. Biaya (*cost*) adalah nilai yang diperoleh dari penjumlahan, jumlah nilai tiap simpul dalam jalur terpendek dari starting point ke A, dan jumlah nilai perkiraan dari sebuah simpul ke simpul tujuan.
7. Simpul tujuan yaitu simpul yang dituju.
8. Halangan adalah sebuah atribut yang menyatakan bahwa sebuah simpul tidak dapat dilalui oleh A (Dalem, 2018).

2.1.5 Kabupaten Mojokerto

Kabupaten Mojokerto adalah salah satu kabupaten di Provinsi Jawa Timur, dengan total luas wilayah mencapai 969,360 km² atau sekitar 2,09% dari total luas Provinsi Jawa Timur. Penggunaan lahan di Kabupaten Mojokerto terus mengalami perubahan setiap tahunnya, seperti alih fungsi lahan pertanian menjadi lahan pemukiman, pekarangan, bangunan, industri, dan sebagian lagi menjadi jalan. Kabupaten Mojokerto berbatasan dengan Kabupaten Lamongan dan Kabupaten Gresik di sebelah utara, Kabupaten Sidoarjo dan Kabupaten Pasuruan di sebelah timur, Kota Batu dan Kota Malang di sebelah selatan, serta Kabupaten Jombang di sebelah barat. Di tengah wilayahnya terdapat Kota Mojokerto.

Secara geografis, Kabupaten Mojokerto berada di antara 111°20'13" hingga 111°40'47" Bujur Timur dan 7°18'35" hingga 7°47' Lintang Selatan. Secara administratif, wilayah ini termasuk dalam cakupan Badan Koordinasi Wilayah Pemerintahan dan Pembangunan Bojonegoro. Dalam Rencana Tata Ruang Wilayah Jawa Timur, Kabupaten Mojokerto masuk dalam kawasan pengembangan "Gerbang Kertosusila." Kabupaten ini terdiri atas 18 kecamatan, 299 desa, dan 5 kelurahan. Dilihat dari karakteristik tanahnya, wilayah Kabupaten Mojokerto

cenderung cekung di bagian tengah, sementara bagian selatan dan utara lebih tinggi. Di selatan, wilayah pegunungan dengan tanah subur meliputi Kecamatan Pacet, Trawas, Gondang, dan Jatirejo. Bagian tengah adalah dataran sedang, sedangkan bagian utara berupa perbukitan kapur yang cenderung kurang subur. Sekitar 30% wilayah Kabupaten Mojokerto memiliki kemiringan lebih dari 15 derajat, sedangkan sisanya berupa dataran dengan kemiringan di bawah 15 derajat. Secara umum, ketinggian wilayah di Kabupaten Mojokerto rata-rata berada di bawah 500 meter di atas permukaan laut, kecuali Kecamatan Pacet dan Trawas yang memiliki wilayah dengan ketinggian lebih dari 700 meter di atas permukaan laut. Berikut adalah potensi pengembangan yang ada di Kabupaten Mojokerto:

1. Bidang industri. Kabupaten Mojokerto dilihat dari posisi geografis merupakan wilayah yang potensi untuk pengembangan industri disebabkan:
 - a. Kemudahan akses transportasi ekspor-impor didukung oleh kedekatan dengan akses transportasi udara (Bandara Juanda), laut (Pelabuhan Tanjung Perak), serta akses darat melalui pembangunan Tol Sumo (Surabaya-Mojokerto).
 - b. Penyediaan kawasan peruntukan industri yang luas dan mudah diakses, antara lain kawasan industri di Kecamatan Ngoro seluas sekitar 500 ha, kawasan industri di Kecamatan Jetis, Kecamatan Kemlagi, dan Kecamatan Dawarblandong seluas sekitar 10.000 ha, serta kawasan industri di Kecamatan Mojoanyar seluas sekitar 500 ha.
 - c. Penyediaan kawasan peruntukan industri di luar kawasan industri juga tersebar di berbagai kecamatan lainnya.

2. Bidang Infrastruktur. Dalam mendukung pengembangan kawasan industri di Kabupaten Mojokerto, diperlukan fasilitas penunjang, terutama pengembangan infrastruktur jalan dan jembatan yang menghubungkan akses ke kawasan industri. Selain itu, pembangunan jalan juga difokuskan pada akses menuju kawasan wisata guna mengembangkan dan meningkatkan sektor pariwisata di Kabupaten Mojokerto, serta pembangunan jalan lingkungan untuk mendorong pertumbuhan ekonomi masyarakat pedesaan.
3. Bidang pendidikan. Pemerintah Kabupaten Mojokerto, dalam upaya memenuhi kebutuhan tenaga kerja bagi industri di wilayahnya, melaksanakan pembangunan Sumber Daya Manusia (SDM) melalui penyediaan sarana dan prasarana pendidikan, dengan fokus pada pengembangan pendidikan Sekolah Menengah Kejuruan (SMK). Salah satu langkah yang diambil adalah mendirikan SMK di beberapa kecamatan, sesuai dengan kebutuhan tenaga kerja yang diperlukan oleh industri di Kabupaten Mojokerto.
4. Bidang kesehatan. Kesehatan merupakan salah satu indikator penting dalam pembangunan manusia, oleh karena itu Pemerintah Kabupaten Mojokerto juga memprioritaskan pengembangan di bidang kesehatan. Untuk meningkatkan layanan kesehatan masyarakat, direncanakan Puskesmas di wilayah Kabupaten Mojokerto akan menambah fasilitas rawat inap, yang

perlu didukung oleh sarana prasarana serta tenaga paramedis yang memadai sebagai penunjang pelayanan rawat inap tersebut.

5. Bidang pariwisata. Kabupaten Mojokerto memiliki potensi pariwisata yang apabila dikelola dengan benar dapat menyumbangkan Pendapatan Asli Daerah pada Kabupaten Mojokerto cukup besar. Potensi tersebut antara lain objek wisata Petirtaan Jolotundo Trawas, Air Terjun Coban Cangu, Air Terjun Dlundung Trawas, Wana Wisata dan Kolam Air Panas Padusan Pacet, dan Ekowisata Waduk Tanjungan Kemlagi. Pemerintah Kabupaten Mojokerto berencana untuk meningkatkan sarana dan prasarana pada semua objek wisata yang ada untuk menjamin kenyamanan pengunjung, sehingga dapat menarik lebih banyak lagi wisatawan untuk datang ke objek-objek wisata tersebut (Dinas Komunikasi dan Informatika, 2024).

2.2 Kajian Integrasi dengan Al-Quran dan Hadits

Perjalanan Perjalanan (Safar) merupakan Aktivitas aktif seseorang sebagai makhluk hidup memperkuat keberadaannya di dunia ini. Imam Syafi'i menegaskan pandangan ini dalam sajak-sajak di kitab Diwan-nya, bahwa seseorang yang memiliki jiwa progresif dan akal yang sempurna tentu akan merasa bosan jika hanya berdiam di kampung halamannya tanpa menjalani aktivitas dinamis yang membuatnya lebih produktif. Beliau menganjurkan agar individu dengan karakter seperti ini melakukan perjalanan, meninggalkan kampung halaman untuk mencari suasana baru dan ilmu pengetahuan. Lebih dari itu, hakikat kehidupan adalah perjuangan untuk meraih kemuliaan di dunia dan akhirat. Setiap orang pasti

melakukan perjalanan, baik itu jarak pendek maupun jauh, perjalanan yang bersifat situasional atau rutinitas mingguan, bulanan, tahunan, atau bahkan harian (Ali, Izzuddin, & dkk, 2013).

Sebagaimana Allah SWT berfirman dalam Surah Al-Mulk(67) ayat 15 yang berbunyi (Katsir, 2015). Dapat dikatakan bahwa mayoritas atau bahkan seluruh populasi manusia di seluruh dunia telah mengalami pengalaman melakukan perjalanan (safar). Rentang jarak yang ditempuh dalam melakukan hal tersebut juga bervariasi, beberapa melakukannya dari tempat yang jauh sementara yang lain hanya dari tempat yang dekat, dengan tingkat frekuensi dan intensitas yang berbeda tergantung pada individu masing-masing. Kebutuhan setiap individu bervariasi, mulai dari kebutuhan yang bersifat duniawi, seperti berwisata, hingga kebutuhan yang bersifat ukhrawi, seperti perjalanan religi. Dalam Islam, hukum asal bepergian ke berbagai penjuru bumi adalah mubah (diperbolehkan) selama tidak melanggar syariat.

Sebagaimana Allah SWT berfirman dalam Surah Al-Mulk(67) ayat 15 yang berbunyi

هُوَ الَّذِي جَعَلَ لَكُمُ الْأَرْضَ ذُلُولًا فَامْشُوا فِي مَنَاكِبِهَا

Artinya:

“Dialah Yang menjadikan bumi itu mudah bagi kamu, maka berjalanlah di segala penjurunya.”

Maksudnya, berjalanlah kalian ke mana pun yang kalian kehendaki di berbagai kawasan, serta lakukanlah perjalanan mengelilingi semua daerah untuk keperluan mata pencaharian dan perniagaan. Ketahuilah bahwa usaha kalian tidak

akan memberikan manfaat apa pun bagi kalian, kecuali jika Allah memudahkan segala sesuatunya.

Allah SWT berfirman:

وَكُلُوا مِنْ رِزْقِهِ

Artinya:

“dan makanlah sebagian dari rezeki-Nya.”

Maka berupaya dengan menempuh sarananya tidaklah bertentangan dengan citra tawakal kepada Allah.

قَالَ الْإِمَامُ أَحْمَدُ: حَدَّثَنَا أَبُو عَبْدِ الرَّحْمَنِ، حَدَّثَنَا حَيْوَةُ، أَخْبَرَنِي بَكْرُ بْنُ عَمْرٍو، أَنَّهُ سَمِعَ عَبْدَ اللَّهِ بْنَ هُبَيْرَةَ يَقُولُ: إِنَّهُ سَمِعَ أَبَا تَمِيمٍ الْجَيْشَانِي يَقُولُ: إِنَّهُ سَمِعَ عُمَرَ بْنَ الْخَطَّابِ يَقُولُ: أَنَّهُ سَمِعَ رَسُولَ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ يَقُولُ: لَوْ أَنَّكُمْ تَتَوَكَّلُونَ عَلَى اللَّهِ حَقَّ تَوَكُّلِهِ، لَرَزَقَكُمْ كَمَا يَرْزُقُ الطَّيْرَ، تَغْدُو خِمَاصًا وَتَرُوحُ بِطَانًا

Imam Ahmad mengatakan, telah menceritakan kepada kami Abu Abdur Rahman, telah menceritakan kepada kami Haiwah, telah menceritakan kepadaku Bakar ibnu Amr; ia pernah mendengar Abdullah ibnu Hubairah mengatakan bahwa ia pernah mendengar Abus Sahm Al-Habsyani mengatakan bahwa ia pernah mendengar Umar ibnul Khattab mengatakan bahwa ia pernah mendengar Rasulullah Saw. bersabda:

“Seandainya kalian bertawakal kepada Allah dengan sebenar-benar tawakal, niscaya Dia akan memberimu rezeki sebagaimana Dia memberi rezeki kepada burung; burung pergi di pagi hari dalam keadaan perut kosong dan pulang di petang hari dalam keadaan perut kenyang.”

Imam Imam Turmuzi, Imam Nasai, dan Imam Ibnu Majah telah meriwayatkannya melalui hadis Ibnu Hubairah; Imam Turmuzi menyatakan bahwa

hadis ini hasan sahih. Dalam hadis ini ditekankan pentingnya melakukan perjalanan di pagi dan petang untuk mencari rezeki, disertai dengan rasa tawakal kepada Allah Swt., karena Dialah yang menundukkan, yang memperjalankan, dan yang menjadikan segala sesuatu sebagai penyebab adanya rezeki.

وَالِيهِ النُّشُورُ

Artinya:

“Dan hanya kepada-Nyalah kamu (kembali setelah) dibangkitkan.”

Yakni dikembalikan kelak di hari kiamat. Ibnu Abbas, Mujahid, As-Saddi, dan Qatadah mengatakan bahwa *manakibuha* artinya daerah-daerah yang jauh, daerah-daerah pedalamannya, dan seluruh kawasannya. Ibnu Abbas dan Qatadah mengatakan pula bahwa *manakibuha* artinya gunung-gunungnya.

Ibnu Abu Hatim mengatakan, telah menceritakan kepada kami ayahku, telah menceritakan kepada kami Amr ibnu Hakkam Al-Azdi, telah menceritakan kepada kami Syu'bah, dari Qatadah, dari Yunus ibnu Jubair, dari Basyir ibnu Ka'b, bahwa ia membaca ayat ini *“maka berjalanlah di segala penjurunya.”* (Al-Mulk: 15).

Lalu ia berkata kepada budak perempuan yang telah melahirkan anak darinya, *"Jika engkau mengetahui makna manakibuha, berarti engkau merdeka."* Lalu budak perempuannya itu menjawab, *"Manakibuha artinya pegunungannya."* Lalu Basyir ibnu Ka'b bertanya kepada Abu Darda mengenai maknanya, maka Abu Darda menjawab, *"Manakibuha artinya daerah pegunungannya."*

Selain itu, Islam sebagai agama Rahmatan lil 'Alamin telah memberikan panduan lengkap mengenai berbagai aspek kehidupan manusia, termasuk dalam hal berhemat. Islam memiliki peran penting sebagai agama yang mulia, dengan Al-Qur'an dan hadits sebagai pedoman hidup yang membantu menjalani kehidupan

sesuai syari'at Islam. Mematuhi Al-Qur'an dan hadits adalah kewajiban yang berdampak signifikan pada perilaku kita dan dalam menyelesaikan masalah. Di era modern yang serba canggih dan mudah ini, gaya hidup manusia cenderung menjadi boros dan hedonis demi mencapai kepuasan pribadi. Untuk menghindari pola hidup seperti itu, kita harus melatih diri untuk hidup hemat. Seperti yang tercantum dalam Al-Qur'an surat Al-Isra' ayat 29 yang berbunyi (Kemenag, 2024):

وَلَا تَجْعَلْ يَدَكَ مَغْلُولَةً إِلَىٰ عُنُقِكَ وَلَا تَبْسُطْهَا كُلَّ الْبَسْطِ فَتَقْعُدَ مَلُومًا مَّحْسُورًا

Artinya:

"Janganlah engkau jadikan tanganmu terbelenggu pada lehermu (kikir) dan jangan (pula) engkau mengulurkannya secara berlebihan sebab nanti engkau menjadi tercela lagi menyesal."

Selanjutnya, dalam ayat ini, Allah Swt. menjelaskan cara-cara yang baik dalam membelanjakan harta. Allah menerangkan keadaan orang-orang yang kikir dan pemboros dengan menggunakan ungkapan "jangan menjadikan tangan terbelenggu pada leher, tetapi juga jangan terlalu mengulurkannya." Kedua ungkapan ini umum digunakan oleh masyarakat Arab. Ungkapan pertama melambangkan larangan untuk bersikap bakhil atau kikir, sehingga enggan memberikan harta kepada orang lain, meskipun hanya sedikit. Sementara itu, ungkapan kedua melarang seseorang untuk berlaku boros dalam membelanjakan harta, sehingga melebihi kemampuan yang dimilikinya. Kebiasaan memboroskan harta dapat menyebabkan seseorang tidak memiliki simpanan atau tabungan yang dapat digunakan ketika dibutuhkan (Kementerian Agama RI, 2011).

Dari ayat ini dapat dipahami bahwa cara yang baik dalam membelanjakan harta adalah dengan cara yang hemat, layak, dan wajar, tidak terlalu bakhil dan tidak terlalu boros. Sikap yang terlalu bakhil akan membuat seseorang tercela, sementara

sikap yang terlalu boros dapat menyebabkan pelakunya mengalami kebangkrutan atau pailit.

Sebagai contoh, beberapa penjelasan yang diperoleh dari hadis-hadis Nabi dapat disampaikan, di mana Imam Ahmad dan para ahli hadis lainnya meriwayatkan dari Ibnu Abbas bahwa Rasulullah SAW bersabda:

وَمَا عَالٍ مِّنْ اِقْتَصَادٍ

Artinya:

“Tidak akan menjadi miskin orang yang berhemat.”

Hadis ini menjelaskan pentingnya berhemat, sehingga Nabi mengatakan bahwa orang yang selalu berhemat tidak akan menjadi beban orang lain atau menjadi miskin. Imam al-Baihaqi meriwayatkan sebuah hadis dari Ibnu Abbas bahwa Rasulullah saw bersabda:

الْاِقْتِصَادُ فِي النَّفَقَةِ نِصْفُ الْمَعِيشَةِ

Artinya:

“Berlaku hemat dalam membelanjakan harta, separuh dari penghidupan.”

Hal ini mencerminkan prinsip-prinsip kebijaksanaan dan tanggung jawab dalam mengelola harta yang diajarkan dalam ajaran Islam.

2.3 Kajian penerapan Algoritma *A-Star* dengan Teori Pendukung

Penelitian yang dilakukan oleh (Fauzi, Supriyadi, & Yanitasari, 2018) berjudul "Implementasi Algoritme *A-Star* untuk Pemetaan Koordinat Tumbuhan Langka Berbasis Web" menghasilkan temuan bahwa algoritme *A-Star* mampu menentukan rute terpendek menuju lokasi tumbuhan yang diinginkan. Selain

fasilitas pencarian rute, aplikasi yang dikembangkan juga mendukung pengelolaan informasi mengenai tumbuhan di dalam kawasan PT Pupuk Kujang. Rute yang dihasilkan berupa graph dua dimensi yang akan ditampilkan ketika pengguna berada dalam kawasan tersebut. Berdasarkan hasil penelitian, aplikasi ini diusulkan untuk dilengkapi dengan sistem mobile berbasis Android agar dapat memandu pengguna langsung ke lokasi tumbuhan, serta menambahkan teknologi Augmented Reality untuk membuat program lebih interaktif dan menarik.

Penelitian (Setyawan, Irawan, & Faisol, 2023) dengan judul "Rancang Bangun Aplikasi Penentuan Rute Terpendek Distribusi Jamu Tradisional Menggunakan Metode A*" menunjukkan bahwa hasil pengujian metode *A-Star* memiliki akurasi yang cukup tinggi, meskipun terdapat selisih beberapa meter dibandingkan hasil dari Google Maps. Jarak antara PJ Ismoyo dengan beberapa toko jamu tradisional dalam metode *A-Star* adalah 6,21 km, sedikit berbeda dengan hasil Google Maps yang menunjukkan 6,7 km. Pengujian blackbox membuktikan bahwa aplikasi berjalan sesuai dengan harapan, dan mayoritas pengguna menyatakan aplikasi sudah berfungsi dengan baik. Sementara itu, penelitian (Lubis, Triase, & Alda, 2024) dalam "Implementasi Algoritma A* (*A-Star*) pada Sistem Informasi Geografis Berbasis Web untuk Menentukan Jalur Terpendek Halte Bus Transmetro Deli" menjelaskan bahwa aplikasi berbasis *A-Star* ini mampu menemukan halte bus terdekat dengan mempertimbangkan node-node yang ada. Jika solusi ditemukan, aplikasi akan menampilkan halte dengan jarak terpendek, namun jika tidak, pesan akan muncul menandakan bahwa tidak ada halte yang dapat dijangkau.

BAB III

METODE PENELITIAN

3.1 Jenis Penelitian

Penelitian ini menggunakan dua pendekatan utama, yaitu evaluasi kuantitatif dan studi lapangan dengan bantuan Google Maps, Google Earth, dan Google Colab. Evaluasi kuantitatif bertujuan untuk menilai performa sistem melalui data numerik dan statistik. Ini melibatkan pengujian dan analisis untuk memahami efektivitas sistem serta mengidentifikasi kelebihan dan kelemahannya. Pendekatan studi lapangan melibatkan eksplorasi langsung menggunakan sumber daya informasi seperti pemetaan geografis dan analisis visual dari area yang diteliti. Data dari kedua pendekatan ini diolah untuk menghasilkan kesimpulan komprehensif tentang performa sistem, menggabungkan analisis numerik dengan wawasan dari lapangan.

3.2 Sumber dan Data Penelitian

Data yang dipakai sebagai objek dalam penelitian ini adalah data sekunder berupa data titik koordinat jalan yang dilewati untuk pergi ke destinasi wisata Kabupaten Mojokerto yang didapatkan dari Google Maps dan Google Earth, serta data 27 lokasi wisata alam dan bersejarah yang didapatkan dari website resmi pariwisata Kabupaten Mojokerto, dengan sumber sebagai berikut:
<https://pariwisata.mojokertokab.go.id/>

3.3 Tahap Penelitian

Tahapan penelitian menggunakan algoritma *A-Star* tersusun dari beberapa tahap. Tahapan-tahapan yang harus dilakukan adalah sebagai berikut:

1. Mengambil informasi rute yang akan dilalui dari Google Maps dalam bentuk tangkapan layar. Langkah awal yang perlu dilakukan adalah mengobservasi dan mengumpulkan data tentang objek yang akan menjadi fokus penelitian. Data yang dikumpulkan mencakup jarak berdasarkan grid dan titik-titik yang diperoleh dari data spasial Google Maps. Penelitian ini akan membahas beberapa aspek penting, termasuk proses pengambilan gambar (capturing) yang melibatkan pengambilan tangkapan layar titik-titik destinasi alam dan bersejarah. Titik-titik lokasi yang telah dikumpulkan akan dijadikan node dalam sebuah graf, sedangkan edge dalam graf akan ditentukan dari rute perjalanan. Selanjutnya, akan dilakukan analisis untuk menentukan apakah edge tersebut bersifat dua arah atau satu arah. Semua ini akan menjadi pertimbangan utama dalam proses pengumpulan dan analisis data yang diperlukan.
2. Mengidentifikasi dan Mengumpulkan Data Koordinat dari Lokasi Destinasi Wisata yang Dibutuhkan. Tahapan dalam menentukan lokasi node adalah menentukan koordinat geografis dari destinasi wisata dan persimpangan jalan. Koordinat ini diperoleh dengan mengambil nilai longitude dan latitude dari Google Maps. Setelah mendapatkan koordinat tersebut, akan digunakan sebagai referensi dalam menghitung nilai heuristik yang dibutuhkan dalam proses pencarian rute terpendek menggunakan algoritma

A-Star. Pada tahap ini, titik awal dilambangkan dengan simbol P_1 , $P_1, P_2, \dots, P_{10}$, titik tujuan atau destinasi wisata dilambangkan dengan simbol $W_1, W_2, W_3, \dots, W_{27}$, dan titik persimpangan dilambangkan dengan simbol $C_1, C_2, C_3, \dots, C_{99}$.

3. Perhitungan Nilai Heuristik. Dalam proses pencarian serta perhitungan nilai heuristik, diperlukan suatu pendekatan untuk mengevaluasi data dari kondisi tertentu guna menentukan seberapa relevan hal itu dalam mencapai solusi yang efisien untuk data yang diinginkan. Fungsi heuristik bisa bervariasi tergantung pada tujuan akhir yang diinginkan, sehingga fungsi ini sering menjadi parameter kunci dalam menyelesaikan suatu masalah. Penggunaan fungsi heuristik digunakan untuk menentukan jarak antara dua titik. Dalam pencarian nilai heuristik ini, digunakan formula *haversine* untuk pencarian hasil jarak. Terdapat beberapa langkah dalam perhitungan untuk mencari berapa jarak antar titik persimpangan, langkah langkahnya sebagai berikut:

- a. Definisikan titik koordinat, setiap titik di permukaan bumi dinyatakan dengan koordinat geografis, yaitu:

- *Latitude* (Lintang): Nilai dalam derajat°
- *Longitude* (Bujur): Nilai dalam derajat°

Misalkan kita memiliki dua titik, yaitu:

- Titik 1 ($\text{Lat1}, \text{Lon1}$) atau dalam bentuk simbol (φ_1, λ_1)
- Titik 2 ($\text{Lat2}, \text{Lon2}$) atau dalam bentuk simbol (φ_2, λ_2)

- b. Konversi derajat ke radian, karena fungsi trigonometri seperti sin dan cos bekerja dengan radian, kita perlu mengonversi nilai derajat ke radian. Rumus konversi adalah:

$$\text{Radian} = \text{Derajat} \cdot \left(\frac{\pi}{180}\right)$$

Untuk setiap koordinat, lakukan konversi:

- Lat1 (φ_1) ke radian
 - Lon1(λ_1) ke radian
 - Lat2 (φ_2) ke radian
 - Lon2 (λ_2) ke radian
- b. Hitung perbedaan koordinat atau perbedaan lintang dan bujur antara dua titik:

$$\Delta\varphi = (\varphi_2 - \varphi_1)$$

$$\Delta\lambda = (\lambda_2 - \lambda_1)$$

- c. Terapkan formula *haversine*. Formula *haversine* adalah sebagai berikut:

$$a = \sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos\varphi_1 \cdot \cos\varphi_2 \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right)$$

Variabel a adalah komponen antara yang digunakan untuk menghitung jarak sudut antara dua titik pada bola bumi. Nilai a dihasilkan dari perhitungan trigonometri yang mempertimbangkan perbedaan lintang dan bujur antara dua titik. Setelah menghitung nilai a , langkah berikutnya adalah menghitung c :

$$c = 2 \cdot \arctan2(\sqrt{a}, \sqrt{1-a})$$

Fungsi dari c adalah untuk mengubah nilai a menjadi jarak sudut dalam satuan radian. Jadi, c mewakili sudut (dalam radian) antara dua titik jika diukur dari pusat bumi. Fungsi dari c adalah untuk mengubah nilai a menjadi jarak sudut dalam satuan radian. Jadi, c mewakili sudut (dalam radian) antara dua titik jika diukur dari pusat bumi.

Perlu diketahui bahwa $\arctan2(y, x)$ merupakan fungsi arctangen empat kuadran, yang didefinisikan memiliki rentang $(-\pi, \pi)$, memberikan satu nilai sudut yang spesifik untuk setiap posisi titik (x, y) pada lingkaran satuan. Fungsi $\arctan2(y, x)$ dapat didefinisikan dalam bentuk arctangen dua kuadran atau lebih detailnya dapat didefinisikan menggunakan fungsi $\arctan(y, x)$ dengan tambahan aturan khusus untuk menentukan kuadran tempat titik (y, x) berada (Betz, 2015).

$$\arctan2(y, x) \triangleq \begin{cases} \arctan\left(\frac{y}{x}\right), & x > 0 \\ \arctan\left(\frac{y}{x}\right) + \pi, & y \geq 0, x < 0 \\ \arctan\left(\frac{y}{x}\right) - \pi, & y < 0, x < 0 \\ \frac{\pi}{2}, & y > 0, x = 0 \\ -\frac{\pi}{2}, & y < 0, x = 0 \\ \text{undefined}, & y = 0, x = 0 \end{cases}$$

- d. Hitung jarak antara dua titik pada permukaan bumi dihitung dengan rumus:

$$d = R \cdot c$$

Dimana, d adalah jarak antara dua titik, dan R adalah jari-jari bumi, yaitu sekitar 6.371 kilometer.

- e. Kalkulasi akhir, setelah mendapatkan nilai c , kalikan dengan jari-jari bumi untuk mendapatkan jarak antar dua titik. Jarak ini biasanya dalam satuan kilometer.

4. Melakukan percobaan untuk mencari rute terpendek. Setelah data terkumpul berdasarkan titik koordinat yang telah diperoleh, langkah berikutnya adalah melakukan serangkaian eksperimen untuk menguji rute terpendek. Dalam pengujian ini, beberapa lokasi destinasi wisata akan ditentukan sebagai titik-titik yang harus dilewati, dan kemudian ditetapkan titik awal dan akhir dari rute. Proses pengujian akan diulang beberapa kali untuk mencakup semua destinasi wisata di Kabupaten Mojokerto. Rute dalam pengujian ini akan ditentukan menggunakan algoritma *AStar* dengan tambahan seleksi destinasi khusus wisata alam dan sejarah, sebagai pengembangan dari penelitian sebelumnya. Berikut merupakan langkah-langkah menentukan jalur terpendek menggunakan algoritma *A-Star*:

- a. Tentukan simpul awal dan simpul tujuan. Tetapkan nilai awal untuk fungsi heuristik dan jarak dari simpul awal ke simpul tujuan.
- b. Hitung nilai fungsi total $f(n)$ untuk setiap simpul, dengan menggunakan persamaan:

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

Keterangan:

$h(n)$ = Biaya jalur termurah dari titik n ke titik tujuan

$g(n)$ = Biaya jalur dari titik awal ke titik n

$f(n)$ = Biaya solusi termurah melalui titik n ke titik tujuan

- c. Pilih simpul dengan nilai $f(n)$ terkecil dan tetapkan sebagai simpul yang sedang diproses.
 - d. Jika simpul yang dipilih adalah simpul tujuan, maka rute terpendek telah ditemukan.
 - e. Jika simpul yang dipilih bukan simpul tujuan, maka perluas simpul tersebut dengan menghitung nilai $g(n)$ dan $h(n)$ untuk setiap tetangga.
 - f. Jika simpul tetangga memiliki nilai $f(n)$ yang lebih kecil dari sebelumnya, perbarui nilainya dan tentukan simpul tersebut sebagai bagian dari jalur optimal.
 - g. Ulangi langkah-langkah c hingga f sampai simpul tujuan ditemukan atau semua kemungkinan jalur telah dievaluasi.
 - h. Setelah simpul tujuan ditemukan, rekonstruksi jalur terpendek dengan mengikuti urutan simpul dari tujuan kembali ke simpul awal.
5. Evaluasi Pengujian. Setelah pengujian rute terpendek selesai, langkah berikutnya adalah menganalisis dan mengevaluasi hasil tersebut terkait penghematan dan efisiensi yang diperoleh dari algoritma *A-Star*. Rincian perhitungan persentase penghematan atau efektivitas jarak tempuh dari hasil pengujian ini diperoleh melalui metode yang dijelaskan di bawah ini:
- a. Mengumpulkan data jarak tempuh yang dihasilkan oleh Google Maps dan algoritma *A-Star* untuk setiap rute yang diuji.

- b. Membandingkan setiap hasil untuk menentukan seberapa banyak algoritma *A-Star* mampu mengurangi jarak tempuh dibandingkan dengan Google Maps.

- c. Menghitung persentase penghematan jarak tempuh dengan formula:

$$\text{Nilai Efektifitas} = \frac{\text{Jarak Google Maps} - \text{Jarak AStar}}{\text{Jarak Google Maps}} \times 100\%$$

- d. Menyajikan hasil perhitungan dalam bentuk tabel atau grafik untuk memudahkan analisis dan pemahaman (Syihabuddin, Jauhari, & dkk, 2022).

BAB IV

HASIL DAN PEMBAHASAN

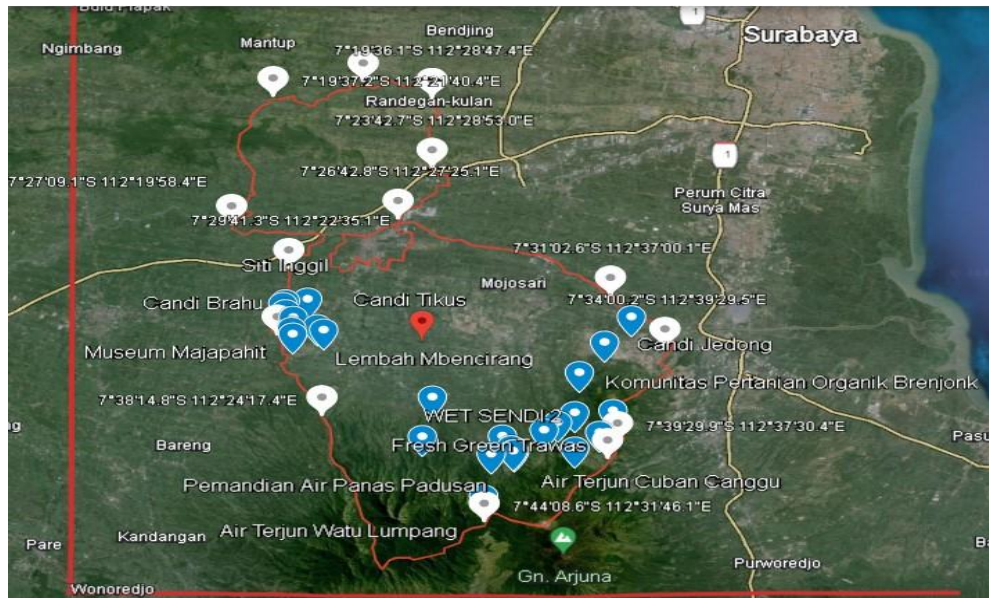
4.1 Deskripsi Data

Sebanyak 27 lokasi wisata di wilayah Kabupaten Mojokerto telah diidentifikasi dan dipilih untuk penelitian ini. Lokasi-lokasi tersebut akan berfungsi sebagai simpul pada graf penelitian, yang menjadi dasar analisis lebih lanjut. Berikut adalah daftar dari 27 lokasi wisata yang akan digunakan dalam studi ini:

Tabel 4.1 Tabel Destinasi Wisata Kabupaten Mojokerto

No.	Lokasi Wisata	Simbol
1.	Candi Brahu	W5
2.	Siti Inggil	W6
3.	Maha Vihara Majapahit	W8
4.	Gapura Wringin Lawang Trowulan	W7
5.	Museum Majapahit	W4
6.	Candi Kedaton (Sumur Upas)	W9
7.	Makam Troloyo	W10
8.	Candi Bajang Ratu	W14
9.	Candi Tikus	W15
10.	Lembah Mbencirang	W16
11.	Akar Seribu	W23
12.	Wet Sendi 2	W24
13.	Air Terjun Watu Lumpang	W25
14.	Air Terjun Grenjengan	W26
15.	De'qoem Qoem	W27
16.	Pemandian Air Panas Padusan	W17
17.	Air Terjun Cuban Canggal	W18
18.	Taman Wisata Ubalan Waterpark Pacet Mojokerto	W19
19.	Air Terjun Watu Gedhek	W20
20.	Alas Veenuz Trawas	W21
21.	Air Terjun Dlundung	W22
22.	Komunitas Pertanian Organik Brenjonk	W3
23.	Wisata Sawah Sumber Gempong	W1
24.	Fresh Green Trawas	W2
25.	Petirtaan Jolotundo	W11
26.	Candi Jedong	W12
27.	Wisata Sumber Dhuwur Sidorejo	W13

Berikut ini adalah daftar lengkap dari berbagai titik lokasi wisata yang berada di Kabupaten Mojokerto. Daftar ini telah diperoleh setelah melakukan pencarian mendetail dengan menggunakan Google Maps, yang membantu dalam mengidentifikasi dan memastikan keberadaan lokasi-lokasi wisata tersebut untuk keperluan penelitian ini:



Gambar 4.1 Screenshoot data Lokasi dan Jalur Kabupaten Mojokerto

Dalam gambar di atas, titik-titik biru menunjukkan lokasi-lokasi destinasi wisata yang telah dipilih untuk penelitian ini. Sedangkan titik-titik putih menunjukkan perbatasan jalan yang memisahkan Kabupaten Mojokerto dari kabupaten-kabupaten yang berbatasan langsung dengannya. Garis merah pada peta berfungsi sebagai penanda batas keseluruhan antara Kabupaten Mojokerto dan Kota Mojokerto, serta kabupaten-kabupaten tetangga seperti Kabupaten Lamongan, Kabupaten Sidoarjo, Kabupaten Jombang, Kabupaten Pasuruan, dan Kabupaten Malang. Penanda ini memberikan gambaran yang jelas tentang wilayah studi dan batas administratif yang relevan dalam konteks penelitian.

4.2 Pengujian Pencarian Rute Terpendek

Pengujian rute terpendek dari beberapa titik awal yang diambil sebagai sampel akan dilakukan dalam tahapan-tahapan penelitian yang terinci, menggunakan Algoritma *A-Star* sebagaimana dijelaskan dalam bab 3. Proses ini melibatkan serangkaian langkah yang sistematis dan terstruktur untuk menghitung pengujian rute terpendek. Berikut adalah alur lengkap dari proses yang akan dilakukan dalam penelitian untuk menghitung pengujian rute terpendek:

1. Memilih Titik Awal dan Titik Tujuan

Langkah awal dalam penelitian ini melibatkan proses seleksi titik awal perjalanan menggunakan Google Maps yang memerlukan pertimbangan yang cermat. Titik-titik awal ini dipilih dengan memperhatikan beberapa faktor, termasuk sebagai jalur utama yang sering dikunjungi oleh wisatawan. Namun, dalam penentuan titik-titik awal ini, perlu diperhatikan juga kondisi setiap jalur yang akan dilalui serta mempertimbangkan batas administratif antara kota dan kabupaten yang berbatasan. Penentuan batas destinasi wisata budaya yang dapat dikunjungi didasarkan pada analisis jumlah titik pada deskripsi data yang telah disajikan sebelumnya. Berikut adalah daftar titik-titik awal yang digunakan untuk perhitungan dalam penelitian ini:

Tabel 4.2 Tabel Titik Awal

No.	Nama Jalan	Simbol	Koordinat	
			<i>Latitude</i>	<i>Longitude</i>
1.	Jl. Raya Slep	P1	-7.67475	112.61813
2.	Jl. Totok Kerot	P2	-7.559440	112.369550
3.	Jl. Raya Kejagan	P6	-7.541012	112.380153

Lokasi destinasi wisata dapat dipilih untuk beberapa lokasi wisata atau kelompok sekaligus. Hal ini disebabkan oleh jumlah data lokasi destinasi wisata yang terbatas pada 27 destinasi dalam dataset penelitian ini. Format penulisan untuk beberapa titik wisata adalah dengan memisahkan mereka menggunakan tanda koma (,). Contoh-contoh dari penentuan beberapa titik awal dan beberapa lokasi tujuan dapat ditemukan dalam Tabel 4.3.

Tabel 4.3 Tabel Titik Tujuan Wisata Kabupaten Mojokerto

Titik Awal	Titik-titik Tujuan Wisata
P1	W3, W2, W1
P2	W4, W8, W5
P6	W7, W6, W5

2. Perhitungan Jarak Antar Titik

Setelah pemilihan titik awal dan titik akhir dilakukan, tahap berikutnya adalah menghitung jarak antara setiap titik. Perhitungan ini menggunakan rumus Haversine, yang berfungsi untuk menghitung jarak antara dua titik berdasarkan koordinat geografisnya. Berikut ini disajikan perhitungan manual dari hasil perhitungan jarak. Adapun langkah-langkah menghitung jarak antara dua titik koordinat menggunakan rumus Haversine adalah sebagai berikut:

- Diketahui:

1. C15 (Titik awal): $\varphi_1 = -7.55716$, $\lambda_1 = 112.37203$
2. C9 (Titik tujuan): $\varphi_2 = -7.55583$, $\lambda_2 = 112.3738$

Penyelesaian:

1. Konversi derajat desimal ke radian

a. C15: $\varphi_1 = \frac{-7.55716 \times \pi}{180} = -0.131897$ radian

b. C15: $\lambda_1 = \frac{112.37203 \times \pi}{180} = 1.96126$ radian

c. C9: $\varphi_2 = \frac{-7.55583 \times \pi}{180} = -0.131874$ radian

d. C9: $\lambda_2 = \frac{112.3738 \times \pi}{180} = 1.96129$ radian

2. Hitung perbedaan bujur ($\Delta\varphi$) dan perbedaan lintang ($\Delta\lambda$)

a. $\Delta\varphi = \varphi_2 - \varphi_1 = (-0.131874) - (-0.131897) = 0.000023$
radian

b. $\Delta\lambda = \lambda_2 - \lambda_1 = 1.96129 - 1.96126 = 0.00003$ radian

3. Hitung $\sin^2\left(\frac{\Delta\varphi}{2}\right)$ dan $\sin^2\left(\frac{\Delta\lambda}{2}\right)$

a. $\sin\left(\frac{\Delta\varphi}{2}\right) = \sin\left(\frac{0.000023}{2}\right) = \sin(0.0000115) = 0.0000115$

b. $\sin^2\left(\frac{\Delta\varphi}{2}\right) = (0.0000115)^2 = 1.3225 \times 10^{-10}$

c. $\sin\left(\frac{\Delta\lambda}{2}\right) = \sin\left(\frac{0.00003}{2}\right) = \sin(0.000015) = 0.000015$

d. $\sin^2\left(\frac{\Delta\lambda}{2}\right) = (0.000015)^2 = 2.25 \times 10^{-10}$

4. Hitung a dalam rumus *haversine*

a. $\cos \varphi_1 = \cos(-0.131897) = 0.991314$

b. $\cos \varphi_2 = \cos(-0.131874) = 0.991317$

c. $a = \sin^2\left(\frac{\Delta\lambda}{2}\right) + \cos \varphi_1 \cdot \cos \varphi_2 \cdot \sin^2\left(\frac{\Delta\varphi}{2}\right)$
 $= 1.3225 \times 10^{-10} + 0.991314 \cdot 0.991317 \cdot 2.25 \times 10^{-10}$
 $= 3.5335894462105 \times 10^{-10}$

5. Hitung c (*Angular distance* dalam radian)

a. $c = 2 \cdot$

$$\arctan2(\sqrt{3.5335894462105 \times 10^{-10}}, \sqrt{1 - 3.5335894462105 \times 10^{-10}})$$

$$= 2 \cdot \arctan2(1.87978 \times 10^{-5}, 1)$$

$$= 2 \cdot \arctan \frac{1.87978 \times 10^{-5}}{1}$$

$$= 2 \cdot (1.87978 \times 10^{-5}) = 3.75956 \times 10^{-5} \text{ radian}$$

6. Hitung jarak d dengan radius bumi $R = 6371$ km

a. $d = 6371 \cdot 3.75956 \times 10^{-5} = 0.239518 \approx 0.24$ km

Jadi, hasil pengukuran jarak antara titik C15 menuju C9 menggunakan perhitungan rumus Haversine menunjukkan jarak sebesar 0,24 km.

- Diketahui:

1. C9 (Titik awal): $\varphi_1 = -7.55583$, $\lambda_1 = 112.3738$
2. W8 (Titik tujuan): $\varphi_2 = -7.55349$, $\lambda_2 = 112.37475$

Penyelesaian:

1. Konversi derajat desimal ke radian

a. C9: $\varphi_1 = \frac{-7.55583 \times \pi}{180} = -0.13187411$ radian

b. C9: $\lambda_1 = \frac{112.3738 \times \pi}{180} = 1.9612928$ radian

c. W8: $\varphi_2 = \frac{-7.55349 \times \pi}{180} = -0.13183327$ radian

d. W8: $\lambda_2 = \frac{112.37475 \times \pi}{180} = 1.96130938$ radian

2. Hitung perbedaan bujur ($\Delta\varphi$) dan perbedaan lintang ($\Delta\lambda$)

a. $\Delta\varphi = \varphi_2 - \varphi_1 = (-0.13183327) - (-0.13187411)$
 $= 0.00004084$ radian

b. $\Delta\lambda = \lambda_2 - \lambda_1 = 1.96130938 - 1.9612928$
 $= 0.00001658$ radian

3. Hitung $\sin^2\left(\frac{\Delta\varphi}{2}\right)$ dan $\sin^2\left(\frac{\Delta\lambda}{2}\right)$

$$a. \sin\left(\frac{\Delta\varphi}{2}\right) = \sin\left(\frac{0.00004084}{2}\right) = \sin(0.00002041) = 0.00002041$$

$$b. \sin^2\left(\frac{\Delta\varphi}{2}\right) = (0.00002041)^2 = 4.16568 \times 10^{-10}$$

$$c. \sin\left(\frac{\Delta\lambda}{2}\right) = \sin\left(\frac{0.00001658}{2}\right) = \sin(0.00000829) = 0.00000829$$

$$d. \sin^2\left(\frac{\Delta\lambda}{2}\right) = (0.00000829)^2 = 6.87241 \times 10^{-11}$$

4. Hitung a dalam rumus *haversine*

$$a. \cos \varphi_1 = \cos(-0.13187411) = 0.991317$$

$$b. \cos \varphi_2 = \cos(-0.13183327) = 0.991323$$

$$\begin{aligned} c. a &= \sin^2\left(\frac{\Delta\lambda}{2}\right) + \cos \varphi_1 \cdot \cos \varphi_2 \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right) \\ &= 4.16568 \times 10^{-10} + 0.991317 \cdot 0.991323 \cdot 6.87241 \times 10^{-11} \\ &= 4.84104 \times 10^{-10} \end{aligned}$$

5. Hitung c (*Angular distance* dalam radian)

$$\begin{aligned} a. c &= 2 \cdot \arctan2(\sqrt{4.84104 \times 10^{-10}}, \sqrt{1 - 4.84104 \times 10^{-10}}) \\ &= 2 \cdot \arctan2(2.20024 \times 10^{-5}, 1) \\ &= 2 \cdot \arctan \frac{2.20024 \times 10^{-5}}{1} \\ &= 2 \cdot (2.20024 \times 10^{-5}) = 4.40048 \times 10^{-5} \text{ radian} \end{aligned}$$

6. Hitung jarak d dengan radius bumi $R = 6371$ km

$$a. d = 6371 \cdot 4.40048 \times 10^{-5} = 0.280355 \approx 0.28 \text{ km}$$

Jadi, hasil pengukuran jarak antara titik C9 menuju W8 menggunakan perhitungan rumus Haversine menunjukkan jarak sebesar 0,28 km.

Di atas merupakan contoh perhitungan manual jarak antar titik yang dilakukan dengan menggunakan rumus *Haversine*. Mengingat terdapat banyak

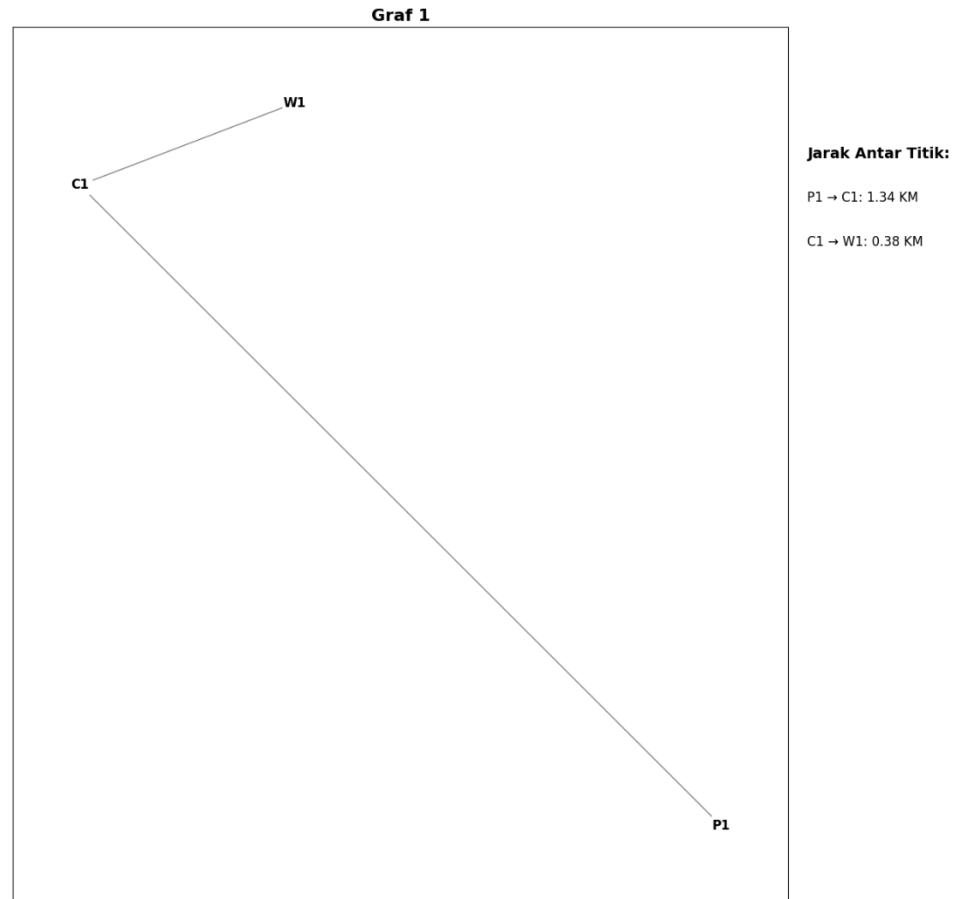
titik yang perlu dihitung dalam pencarian rute, perhitungan jarak antara setiap titik tersebut akan lebih efisien dengan bantuan Google Colab. Dengan menggunakan Google Colab, proses penghitungan jarak dapat dilakukan secara otomatis dan lebih cepat, sehingga memudahkan dalam analisis data dan perencanaan rute yang optimal. Hasil jarak antar titik ini akan disajikan di lampiran, sehingga memudahkan pembaca untuk merujuk pada data tersebut dan memahami perhitungan yang dilakukan.

3. Penentuan Rute

Proses penentuan rute terpendek akan dilakukan dengan menguji setiap titik awal terhadap setiap lokasi titik tujuan menggunakan algoritma *A-Star*. Penjelasan dan perhitungan terperinci mengenai beberapa sampel titik awal dan titik tujuan disajikan dalam konteks perhitungan dan penentuan rute yang paling efisien, sebagaimana ditunjukkan berikut dibawah ini.

1. Tittik Awal P1

- Penentuan Rute rute P1→W1



Gambar 4.2 Visualisasi Rute P1 Menuju W1

P1 merupakan titik awal untuk menuju destinasi, sehingga P1 dapat dimasukkan ke dalam *List* Tutup.

$$List\ Tutup = [['P1'], 0]$$

Karena hanya C1 yang sejalur dilewati, maka C1 dimasukkan ke dalam *List* tutup.

$$List\ Tutup = [['P1', 'C1'], 1.34]$$

Selanjutnya, akan ditentukan titik-titik mana saja yang rute jalannya terhubung dengan C1 serta dilakukan perhitungan nilai heuristik untuk masing-masing titik tersebut. Nilai heuristik ini dapat diperoleh melalui

perhitungan jarak menggunakan formula *haversine* seperti pada persamaan (2.2). Perhitungan ini memanfaatkan koordinat titik-titik yang tersedia di Google Maps. C1 terhubung dengan W1 dan C2, kemudian diperoleh hasil perhitungan berupa panjang rute dan nilai heuristiknya sebagai berikut:

$$\text{Terpilih} = \{ 'W1': (0.38, 0), 'C2': (0.98, 1.29) \}$$

Nilai 1.34 didapatkan dari perhitungan *List Tutup* Sebelumnya, dan 0.38 dari perhitungan jarak antara C1 ke W1 yang kemudian dijumlahkan dan menghasilkan $g(W1)$ yang merupakan biaya jalur dari titik awal P1 ke W1, lalu untuk heuristik $h(W1)$ menghasilkan 0 dikarenakan sudah mencapai titik tujuan. Selanjutnya nilai 1.34 sama saja didapatkan dari perhitungan *List Tutup* Sebelumnya, dan 0.98 dari perhitungan jarak antara C1 ke C2 yang kemudian dijumlahkan dan menghasilkan $g(C2)$ yang merupakan biaya jalur dari titik awal ke n , lalu untuk heuristik $h(C2)$ menghasilkan 1.29 yang merupakan perhitungan dari C2 ke tujuan. Kemudian, kita melakukan seleksi menggunakan perhitungan untuk menentukan nilai yang paling kecil di antara semua nilai yang ada.

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

$$= (1.34 + 0.38) + 0$$

$$= 1.72$$

$$f(C2) = g(C2) + h(C2)$$

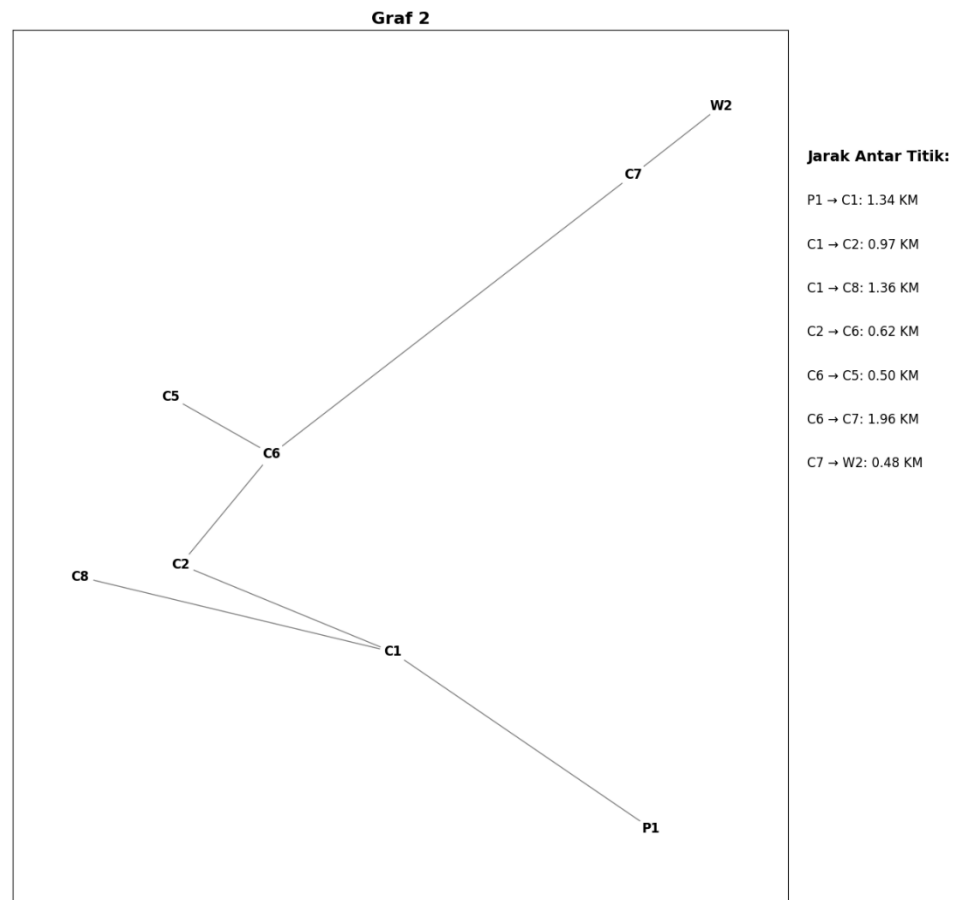
$$= (1.34 + 0.98) + 1.29$$

$$= 3.61$$

W1 dipilih karena memiliki nilai perhitungan terendah dan juga merupakan destinasi akhir, yang berarti telah mencapai tujuan wisata yang diinginkan. Dengan demikian, diperoleh hasil sebagai berikut:

$List\ Tutup = [['P1', 'C1', 'W1'], 1.72]$

- **Penentuan rute $P1 \rightarrow W2$**



Gambar 4.3 Visualisasi Rute P1 Menuju W2

Proses pencarian rute kali ini serupa dengan perhitungan rute sebelumnya. P1 berfungsi sebagai titik awal untuk mencapai destinasi, sehingga P1 dapat dimasukkan ke dalam *List Tutup*.

$List\ Tutup = [['P1'], 0]$

Karena hanya C1 yang sejalur dilewati, maka C1 dimasukkan ke dalam *List* tutup.

$$List\ Tutup = [['P1', 'C1'], 1.34]$$

Selanjutnya, akan ditentukan titik-titik mana saja yang memiliki jalur utama yang terhubung dengan C1, serta dilakukan perhitungan nilai heuristik untuk masing-masing titik tersebut. Setelah perhitungan, diperoleh hasil berupa panjang rute dan nilai heuristik sebagai berikut:

$$Terpilih = \{ 'C2': (0.97, 3.05), 'C8': (1.36, 3.41) \}$$

Selanjutnya, kita melakukan seleksi melalui perhitungan untuk memilih nilai yang terkecil.

$$f(C2) = g(C2) + h(C2)$$

$$= (1.34 + 0.97) + 3.05$$

$$= 5.36$$

$$f(C8) = g(C8) + h(C8)$$

$$= (1.34 + 1.36) + 3.41$$

$$= 6.11$$

Hasil perhitungan yang didapat menunjukkan bahwa C2 memiliki nilai terendah, sehingga dapat dimasukkan ke dalam *list* tutup sebagai berikut:

$$List\ Tutup = [['P1', 'C1', 'C2'], 2.31]$$

Karena hanya C6 yang sejalur dilewati, maka C6 dimasukkan ke dalam *List* tutup.

$$List\ Tutup = [['P1', 'C1', 'C2', 'C6'], 2.93]$$

Sama seperti dalam proses perhitungan sebelumnya, akan dicari titik-titik yang jalur utamanya terhubung dengan C6, dan hasilnya adalah sebagai berikut:

$$\text{Terpilih} = \{ 'C5': (0.5, 2.65), 'C7': (1.96, 0.48) \}$$

$$f(C5) = g(C5) + h(C5)$$

$$= (2.93 + 0.5) + 2.65$$

$$= 6.08$$

$$f(C7) = g(C7) + h(C7)$$

$$= (2.93 + 1.96) + 0.48$$

$$= 5.37$$

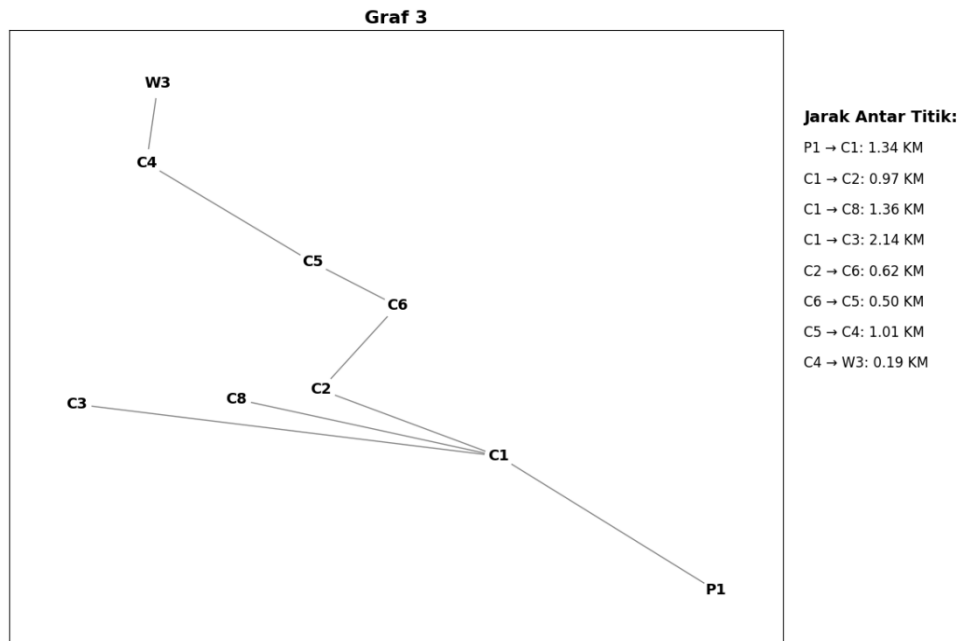
Dengan menggunakan proses yang sama, diperoleh hasil bahwa titik dengan perhitungan terendah adalah C7. Oleh karena itu, C7 dimasukkan ke dalam *List* tutup.

$$\text{List Tutup} = [['P1', 'C1', 'C2', 'C6', 'C7'], 4.89]$$

Sama halnya, karena hanya W2 yang sejalur dilewati dan merupakan jalur destinasi akhir yang menandakan telah sampainya dilokasi tujuan, maka W2 dimasukkan ke dalam *List* tutup sekaligus didapatkan rute terpendeknya:

$$\text{List Tutup} = [['P1', 'C1', 'C2', 'C6', 'C7', 'W2'], 5.37]$$

- **Penentuan rute $P1 \rightarrow W3$**



Gambar 4.4 Visualisasi Rute P1 Menuju W3

Proses pencarian rute kali ini serupa dengan perhitungan rute sebelumnya. P1 berfungsi sebagai titik awal untuk mencapai destinasi, sehingga P1 dapat dimasukkan ke dalam *List Tutup*.

$$List\ Tutup = [[P1], 0]$$

Karena hanya C1 yang sejalur dilewati, maka C1 dimasukkan ke dalam *List tutup*.

$$List\ Tutup = [[P1, C1], 1.34]$$

Selanjutnya, akan ditentukan titik-titik mana saja yang memiliki jalur utama yang terhubung dengan C1, serta dilakukan perhitungan nilai heuristik untuk masing-masing titik tersebut. Setelah perhitungan, diperoleh hasil berupa panjang rute dan nilai heuristik sebagai berikut:

$$Terpilih = \{C2: (0.97, 1.97), C8: (1.36, 1.88), C3: (2.14, 1.92)\}$$

Selanjutnya, kita melakukan seleksi melalui perhitungan untuk memilih nilai yang terkecil.

$$f(C2) = g(C2) + h(C2)$$

$$= (1.34 + 0.97) + 1.97$$

$$= 4.28$$

$$f(C8) = g(C8) + h(C8)$$

$$= (1.34 + 1.36) + 1.88$$

$$= 4.58$$

$$f(C3) = g(C3) + h(C3)$$

$$= (1.34 + 2.14) + 1.92$$

$$= 5.4$$

Hasil perhitungan yang didapat menunjukkan bahwa C2 memiliki nilai terendah, sehingga dapat dimasukkan ke dalam *List tutup* sebagai berikut:

$$List\ Tutup = [['P1', 'C1', 'C2'], 2.31]$$

Karena hanya C6 yang sejalur dilewati, maka C6 dimasukkan ke dalam *List tutup*, begitupun juga C5 dan C4.

$$List\ Tutup = [['P1', 'C1', 'C2', 'C6'], 2.93]$$

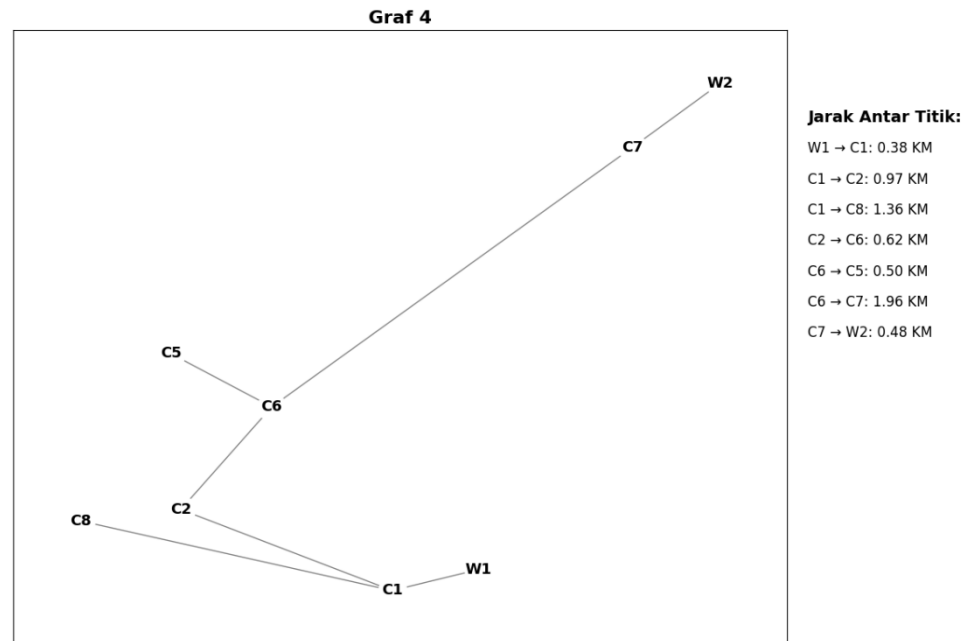
$$List\ Tutup = [['P1', 'C1', 'C2', 'C6', 'C5'], 3.43]$$

$$List\ Tutup = [['P1', 'C1', 'C2', 'C6', 'C5', 'C4'], 4.44]$$

Sama halnya diatas, karena hanya W3 yang sejalur dilewati dan merupakan jalur destinasi akhir yang menandakan telah sampainya dilokasi tujuan, maka W3 dimasukkan ke dalam *List tutup* sekaligus didapatkan rute terpendeknya:

$$List\ Tutup = [['P1', 'C1', 'C2', 'C6', 'C5', 'C4', 'W3'], 4.63]$$

- **Penentuan rute W1→W2**



Gambar 4.5 Visualisasi Rute W1 Menuju W2

W1 hanya terhubung dengan C1, oleh karena itu C1 dimasukkan ke dalam *List* tutup.

$$List \text{ Tutup} = [['W1', 'C1'], 0.38]$$

C1 terhubung dengan C2 dan C8 yang selanjutnya akan dilakukan seleksi melalui perhitungan untuk memilih nilai yang terkecil.

$$Terpilih = \{ 'C2': (0.97, 3.05), 'C8': (1.36, 3.41) \}$$

$$f(C2) = g(C2) + h(C2)$$

$$= (0.38 + 0.97) + 3.05$$

$$= 4.4$$

$$f(C8) = g(C8) + h(C8)$$

$$= (0.38 + 1.36) + 3.41$$

$$= 5.15$$

Hasil perhitungan yang didapat menunjukkan bahwa C2 memiliki nilai terendah, sehingga dapat dimasukkan ke dalam *List* tutup sebagai berikut:

$$List\ Tutup = [['W1', 'C1', 'C2'], 1.35]$$

C2 hanya terhubung dengan C6, oleh karena itu C6 dimasukkan ke dalam *List* tutup.

$$List\ Tutup = [['W1', 'C1', 'C2', 'C6'], 1.97]$$

Selanjutnya, dicari titik yang terhubung dengan C6, yang kemudian didapatkan hasil:

$$Terpilih = \{ 'C5': (0.5, 2.65), 'C7': (1.96, 0.48) \}$$

$$f(C5) = g(C5) + h(C5)$$

$$= (1.97 + 0.49) + 2.65$$

$$= 5.11$$

$$f(C7) = g(C7) + h(C7)$$

$$= (1.97 + 1.96) + 0.48$$

$$= 4.41$$

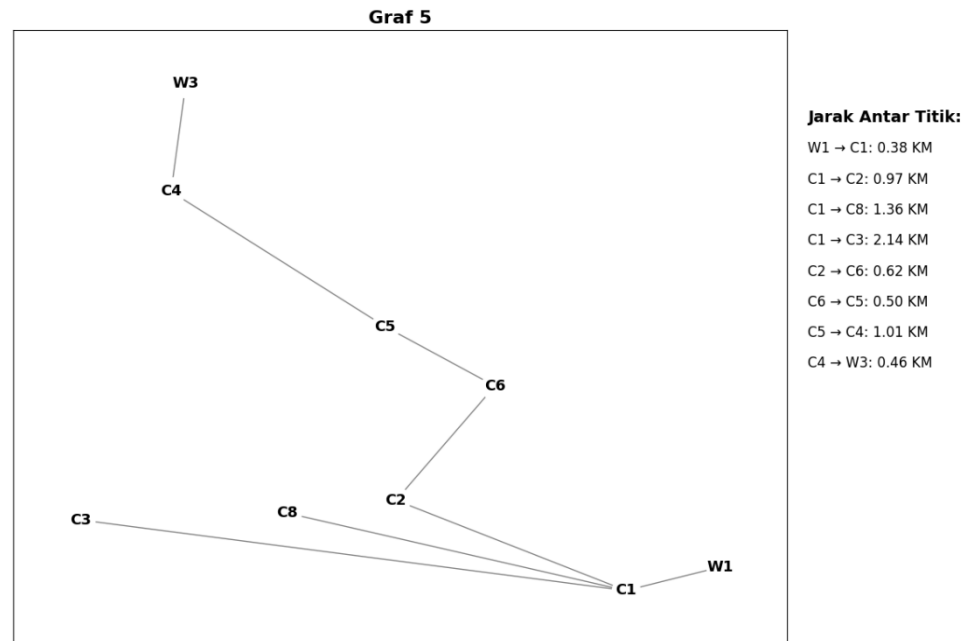
Hasil perhitungan yang didapat menunjukkan bahwa C7 memiliki nilai terendah, sehingga dapat dimasukkan ke dalam *List* tutup sebagai berikut:

$$List\ Tutup = [['W1', 'C1', 'C2', 'C6', 'C7'], 3.93]$$

Sama halnya perhitungan sebelumnya, karena hanya W2 yang sejalur dilewati dan merupakan jalur destinasi akhir yang menandakan telah sampainya dilokasi tujuan, maka W2 dimasukkan ke dalam *List* tutup sekaligus didapatkan rute terpendeknya:

$$Lit\ Tutup = [['W1', 'C1', 'C2', 'C6', 'C7', 'W2'], 4.41]$$

- **Penentuan rute W1→W3**



Gambar 4.6 Visualisasi Rute W1 Menuju W3

W1 hanya terhubung dengan C1, oleh karena itu C1 dimasukkan ke dalam *List* tutup.

$$List \text{ Tutup} = [['W1', 'C1'], 0.38]$$

C1 terhubung dengan C2 dan C8 yang selanjutnya akan dilakukan seleksi melalui perhitungan untuk memilih nilai yang terkecil.

$$Terpilih = \{ 'C2': (0.97, 1.97), 'C8': (1.36, 1.88), 'C3':$$

$$(2.14, 1.92) \}$$

$$f(C2) = g(C2) + h(C2)$$

$$= (0.38 + 0.97) + 1.97$$

$$= 3.32$$

$$f(C8) = g(C8) + h(C8)$$

$$= (0.38 + 1.36) + 1.88$$

$$= 3.62$$

$$\begin{aligned}
 f(C3) &= g(C3) + h(C3) \\
 &= (0.38 + 2.14) + 1.92 \\
 &= 4.44
 \end{aligned}$$

Hasil perhitungan yang didapat menunjukkan bahwa C2 memiliki nilai terendah, sehingga dapat dimasukkan ke dalam *List* tutup sebagai berikut:

$$List\ Tutup = [['W1', 'C1', 'C2'], 1.35]$$

Karena hanya C6 yang sejalur dilewati, maka C6 dimasukkan ke dalam *List* tutup, begitupun juga C5 dan C4.

$$List\ Tutup = [['W1', 'C1', 'C2', 'C6'], 1.97]$$

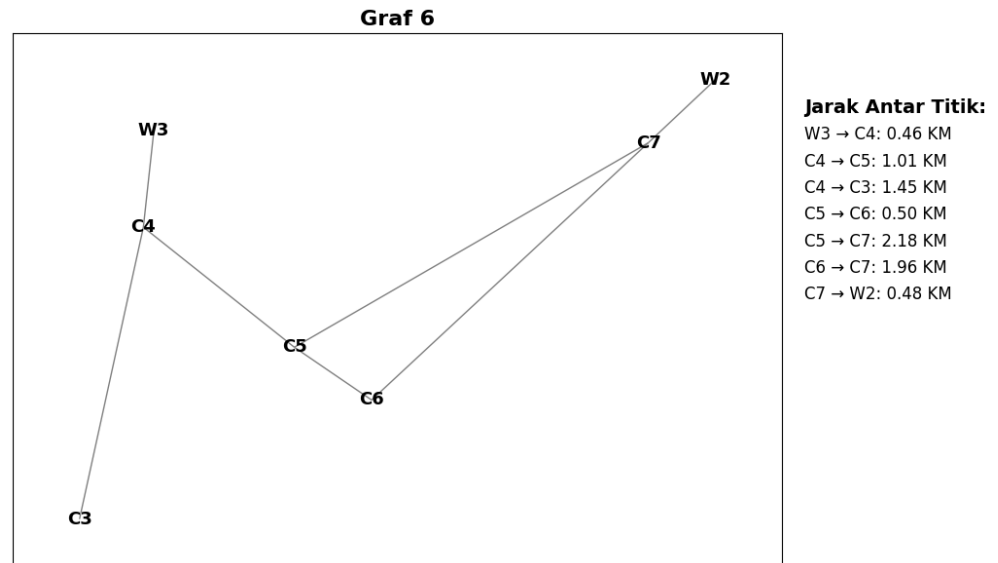
$$List\ Tutup = [['W1', 'C1', 'C2', 'C6', 'C5'], 2.47]$$

$$List\ Tutup = [['W1', 'C1', 'C2', 'C6', 'C5', 'C4'], 3.48]$$

Sama halnya diatas, karena hanya W3 yang sejalur dilewati dan merupakan jalur destinasi akhir yang menandakan telah sampainya dilokasi tujuan, maka W3 dimasukkan ke dalam *List* tutup sekaligus didapatkan rute terpendeknya:

$$List\ Tutup = [['W1', 'C1', 'C2', 'C6', 'C5', 'C4', 'W3'], 3.94]$$

- **Penentuan rute W3→W2**



Gambar 4.7 Visualisasi Rute W3 Menuju W2

W3 hanya terhubung dengan C4, oleh karena itu C4 dimasukkan ke dalam *List* tutup.

$$List\ Tutup = [['W3', 'C4'], 0.46]$$

C4 terhubung dengan C5 dan C3 yang selanjutnya akan dilakukan seleksi melalui perhitungan untuk memilih nilai yang terkecil.

$$Terpilih = \{ 'C5': (1.01, 2.65), 'C3': (1.45, 4.09) \}$$

$$f(C5) = g(C5) + h(C5)$$

$$= (0.46 + 1.01) + 2.65$$

$$= 4.56$$

$$f(C3) = g(C3) + h(C3)$$

$$= (0.46 + 1.45) + 4.09$$

$$= 6$$

Hasil perhitungan yang didapat menunjukkan bahwa C5 memiliki nilai terendah, sehingga dapat dimasukkan ke dalam *List* tutup sebagai berikut:

$$List\ Tutup = [['W3', 'C4', 'C5'], 1.47]$$

Sama seperti dalam proses perhitungan sebelumnya, akan dicari titik-titik yang jalur utamanya terhubung dengan C5, dan hasilnya adalah sebagai berikut:

$$Terpilih = \{ 'C6': (0.5, 2.44), 'C7': (2.18, 0.48) \}$$

$$f(C6) = g(C6) + h(C6)$$

$$= (1.47 + 0.5) + 2.44$$

$$= 4.41$$

$$f(C7) = g(C7) + h(C7)$$

$$= (1.47 + 2.18) + 0.48$$

$$= 4.13$$

Hasil perhitungan yang didapat menunjukkan bahwa C7 memiliki nilai terendah, sehingga dapat dimasukkan ke dalam *List* tutup sebagai berikut:

$$List\ Tutup = [['W3', 'C4', 'C5', 'C7'], 3.65]$$

Sama halnya diatas, karena hanya W2 yang sejalur dilewati dan merupakan jalur destinasi akhir yang menandakan telah sampainya dilokasi tujuan, maka W2 dimasukkan ke dalam *List* tutup sekaligus didapatkan rute terpendeknya:

$$List\ Tutup = [['W3', 'C4', 'C5', 'C7', 'W2'], 4.13]$$

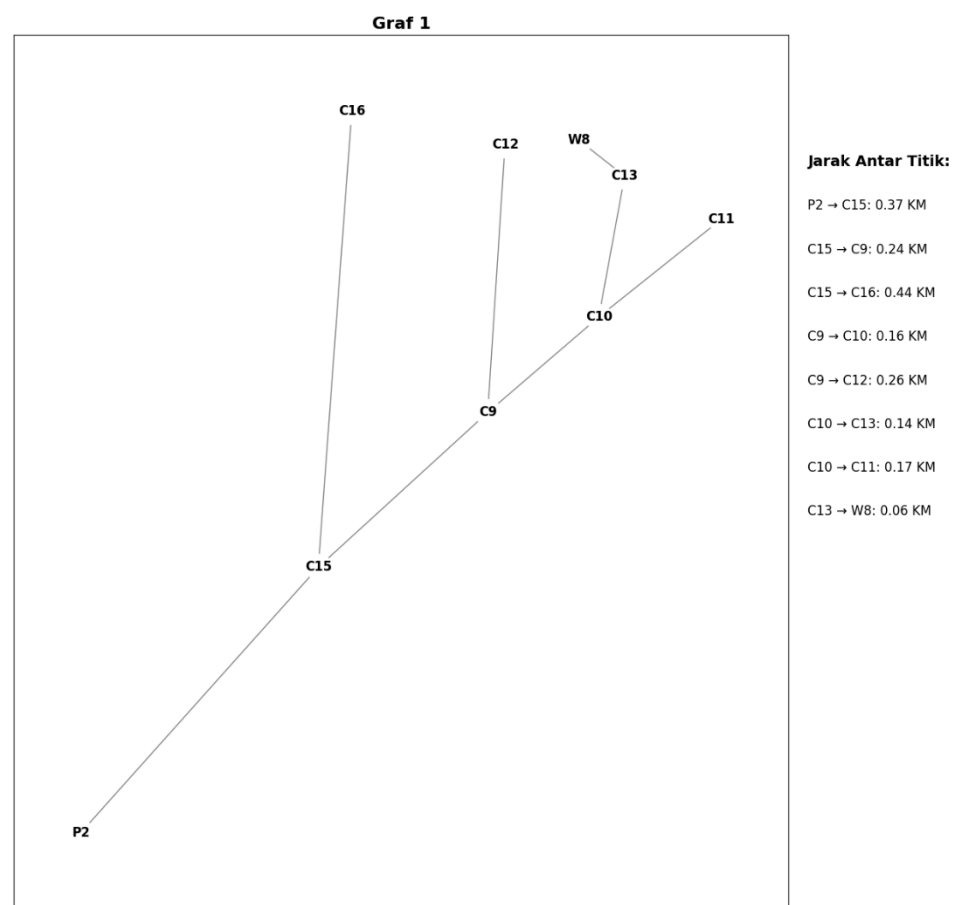
Rute terakhir yang tercantum dalam *List* tutup di atas dapat dimasukkan ke dalam list rute. Selanjutnya, total jarak minimum yang dapat ditempuh akan dijumlahkan.

$$List\ Rute = [['P1', 'C1', 'W1'], ['W1', 'C1', 'C2', 'C6', 'C5', 'C4', 'W3'], \\ ['W3', 'C4', 'C5', 'C7', 'W2'], 9.79]$$

Berdasarkan *List* rute di atas, perjalanan dimulai dari titik P1 dengan tujuan pertama menuju W1. Selanjutnya, perjalanan dilanjutkan ke destinasi kedua, yaitu W3. Setelah melewati W3, rute diteruskan ke destinasi terakhir, yaitu W2.

2. Tittik Awal P2

- **Penentuan rute P2→W8**



Gambar 4.8 Visualisasi Rute P2 Menuju W8

Proses pencarian rute kali ini serupa dengan perhitungan rute sebelumnya. P2 berfungsi sebagai titik awal untuk mencapai destinasi, sehingga P2 dapat dimasukkan ke dalam *List* Tutup.

$$List\ Tutup = [[P2'], 0]$$

Karena hanya C15 yang sejalur dilewati, maka C15 dimasukkan ke dalam *List* tutup.

$$List\ Tutup = [[P2', 'C15'], 0.37]$$

Selanjutnya, akan ditentukan titik-titik mana saja yang rute jalannya terhubung dengan C15 serta dilakukan perhitungan nilai heuristik untuk masing-masing titik tersebut. Nilai heuristik ini dapat diperoleh melalui perhitungan jarak menggunakan formula *haversine* seperti pada persamaan (2.2). Perhitungan ini memanfaatkan koordinat titik-titik yang tersedia di Google Maps. C15 terhubung dengan C9 dan C16, kemudian diperoleh hasil perhitungan berupa panjang rute dan nilai heuristiknya sebagai berikut:

$$Terpilih = \{ 'C9': (0.24, 0.28), 'C16': (0.44, 0.26) \}$$

Nilai 0.37 didapatkan dari perhitungan *List* Tutup Sebelumnya, dan 0.24 dari perhitungan jarak antara C15 ke C9 yang kemudian dijumlahkan dan menghasilkan $g(C9)$ yang merupakan biaya jalur dari titik awal P2 ke C9, lalu untuk heuristik $h(C9)$ menghasilkan 0.28 yang merupakan perhitungan dari C9 ke tujuan W8. Selanjutnya nilai 0.37 sama saja didapatkan dari perhitungan *List* Tutup Sebelumnya, dan 0.44 dari perhitungan jarak antara C15 ke C16 yang kemudian dijumlahkan dan menghasilkan $g(C16)$ yang merupakan biaya jalur dari titik awal ke C16, lalu untuk heuristik $h(C16)$ menghasilkan 0.26 yang merupakan perhitungan dari C16 ke tujuan W8. Kemudian, kita melakukan seleksi menggunakan perhitungan untuk menentukan nilai yang paling kecil di antara semua nilai yang ada.

$$f(C9) = g(C9) + h(C9)$$

$$= (0.37 + 0.24) + 0.28$$

$$= 0.89$$

$$f(C16) = g(C16) + h(C16)$$

$$= (0.37 + 0.44) + 0.26$$

$$= 1.07$$

Hasil perhitungan yang didapat menunjukkan bahwa C9 memiliki nilai terendah, sehingga dapat dimasukkan ke dalam *List* tutup sebagai berikut:

$$List\ Tutup = [[P2', C15', C9'], 0.61]$$

Sama seperti dalam proses perhitungan sebelumnya, akan dicari titik-titik yang jalur utamanya terhubung dengan C9, dan hasilnya adalah sebagai berikut:

$$Terpilih = \{C10': (0.16, 0.17), C12': (0.26, 0.08)\}$$

$$f(C10) = g(C10) + h(C10)$$

$$= (0.61 + 0.16) + 0.17$$

$$= 0.94$$

$$f(C12) = g(C12) + h(C12)$$

$$= (0.61 + 0.26) + 0.08$$

$$= 0.95$$

Hasil perhitungan yang didapat menunjukkan bahwa C10 memiliki nilai terendah, sehingga dapat dimasukkan ke dalam *List* tutup sebagai berikut:

$$List\ Tutup = [[P2', C15', C9', C10'], 0.77]$$

C10 terhubung dengan C13 dan C11, maka panjang rute dan nilai heuristinya adalah:

$$\text{Terpilih} = \{ 'C13': (0.14, 0.06), 'C11': (0.17, 0.18) \}$$

$$f(C13) = g(C13) + h(C13)$$

$$= (0.77 + 0.14) + 0.06$$

$$= 0.97$$

$$f(C11) = g(C11) + h(C11)$$

$$= (0.77 + 0.17) + 0.18$$

$$= 1.12$$

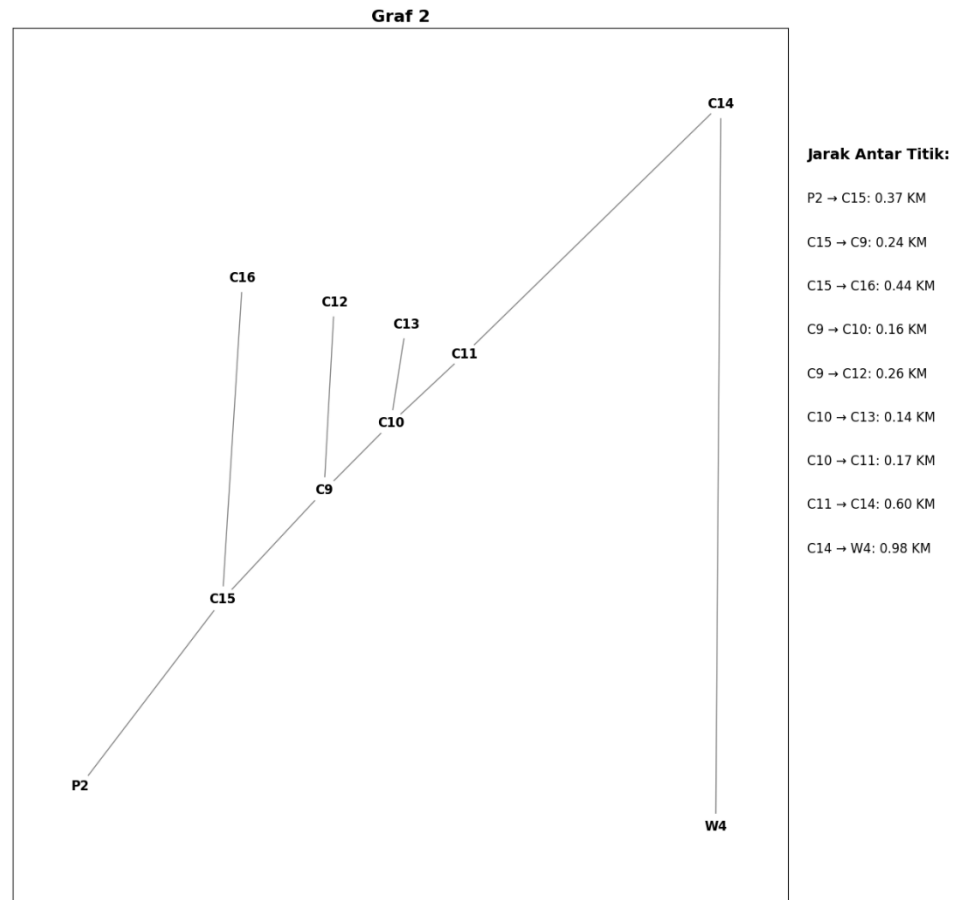
Hasil perhitungan yang didapat menunjukkan bahwa C13 memiliki nilai terendah, sehingga dapat dimasukkan ke dalam *List* tutup sebagai berikut:

$$\text{List Tutup} = [['P2', 'C15', 'C9', 'C10', 'C13'], 0.91]$$

Karena hanya W8 yang sejalur dilewati dan merupakan jalur destinasi akhir yang menandakan telah sampainya dilokasi tujuan, maka W8 dimasukkan ke dalam *List* tutup sekaligus telah sampai dilokasi *cluster* tujuan:

$$\text{List Tutup} = [['P2', 'C15', 'C9', 'C10', 'C13', 'W8'], 0.97]$$

- **Penentuan rute P2→W4**



Gambar 4.9 Visualisasi Rute P2 Menuju W4

Proses pencarian rute kali ini serupa dengan perhitungan rute sebelumnya. P2 berfungsi sebagai titik awal untuk mencapai destinasi, sehingga P2 dapat dimasukkan ke dalam *List Tutup*.

$$List\ Tutup = [['P2'], 0]$$

Karena hanya C15 yang sejalur dilewati, maka C15 dimasukkan ke dalam *List tutup*.

$$List\ Tutup = [['P2', 'C15'], 0.37]$$

Selanjutnya, akan ditentukan titik-titik mana saja yang memiliki jalur utama yang terhubung dengan C15, serta dilakukan perhitungan nilai

heuristik untuk masing-masing titik tersebut. Setelah perhitungan, diperoleh hasil berupa panjang rute dan nilai heuristik sebagai berikut:

$$\text{Terpilih} = \{ 'C9': (0.24, 0.88), 'C16': (0.44, 1.17) \}$$

Selanjutnya, kita melakukan seleksi melalui perhitungan untuk memilih nilai yang terkecil.

$$f(C9) = g(C9) + h(C9)$$

$$= (0.37 + 0.24) + 0.88$$

$$= 1.49$$

$$f(C16) = g(C16) + h(C16)$$

$$= (0.37 + 0.44) + 1.17$$

$$= 1.98$$

Hasil perhitungan yang didapat menunjukkan bahwa C9 memiliki nilai terendah, sehingga dapat dimasukkan ke dalam *List* tutup sebagai berikut:

$$\text{List Tutup} = [['P2', 'C15', 'C9'], 0.61]$$

C9 terhubung dengan C10 dan C12, maka panjang rute dan nilai heuristinya adalah:

$$\text{Terpilih} = \{ 'C10': (0.16, 0.83), 'C12': (0.26, 1.02) \}$$

$$f(C10) = g(C10) + h(C10)$$

$$= (0.61 + 0.16) + 0.83$$

$$= 1.6$$

$$f(C12) = g(C12) + h(C12)$$

$$= (0.61 + 0.26) + 1.02$$

$$= 1.89$$

Hasil perhitungan yang didapat menunjukkan bahwa C10 memiliki nilai terendah, sehingga dapat dimasukkan ke dalam *List* tutup sebagai berikut:

$$List\ Tutup = [['P2', 'C15', 'C9', 'C10'], 0.77]$$

C10 terhubung dengan C13 dan C11, maka panjang rute dan nilai heuristinya adalah:

$$Terpilih = \{ 'C13': (0.14, 0.9), 'C11': (0.17, 0.8) \}$$

$$f(C13) = g(C13) + h(C13)$$

$$= (0.77 + 0.14) + 0.9$$

$$= 1.81$$

$$f(C11) = g(C11) + h(C11)$$

$$= (0.77 + 0.17) + 0.8$$

$$= 1.74$$

Hasil perhitungan yang didapat menunjukkan bahwa C11 memiliki nilai terendah, sehingga dapat dimasukkan ke dalam *List* tutup sebagai berikut:

$$List\ Tutup = [['P2', 'C15', 'C9', 'C10', 'C11'], 0.94]$$

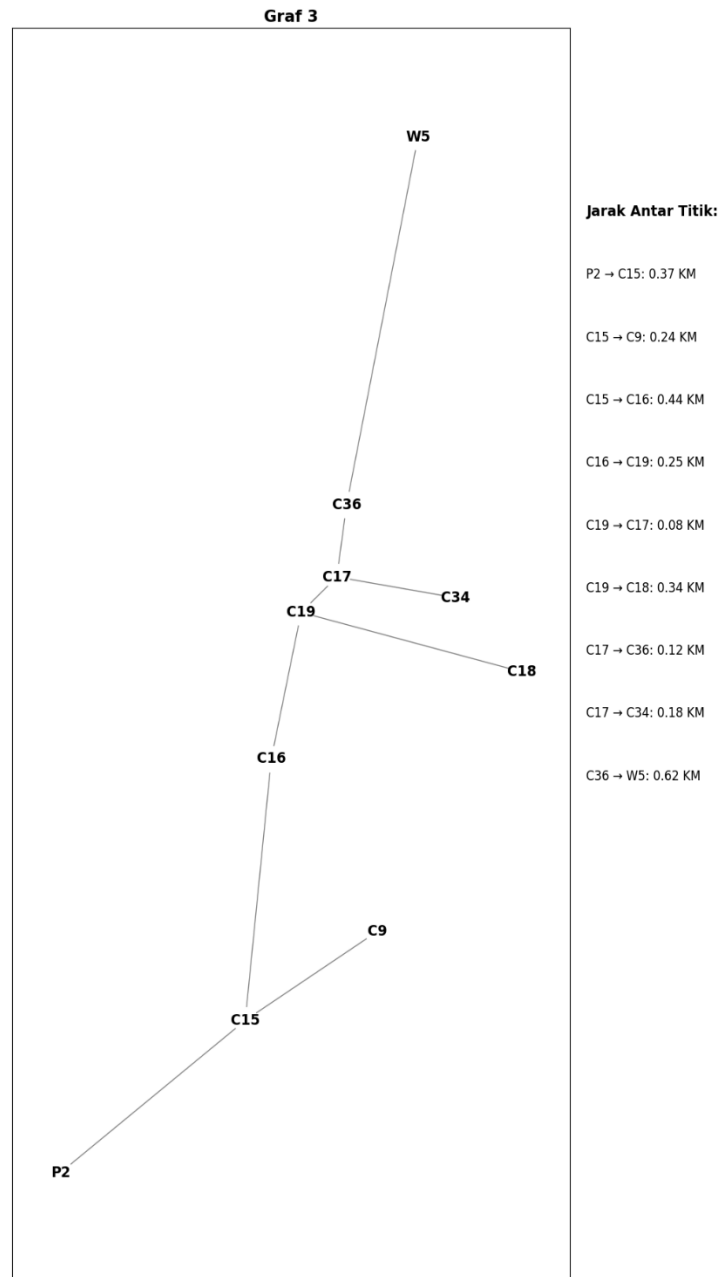
Karena hanya C14 yang sejalur dilewati, maka C14 dimasukkan ke dalam *List* tutup.

$$List\ Tutup = [['P2', 'C15', 'C9', 'C10', 'C11', 'C14'], 1.54]$$

Karena hanya W4 yang sejalur dilewati dan merupakan jalur destinasi akhir yang menandakan telah sampainya dilokasi tujuan, maka W4 dimasukkan ke dalam *List* tutup sekaligus telah sampai dilokasi cluster tujuan:

$$List\ Tutup = [['P2', 'C15', 'C9', 'C10', 'C11', 'C14', 'W4'], 2.52]$$

- **Penentuan rute P2→W5**



Gambar 4.10 Visualisasi Rute P2 Menuju W5

P2 merupakan titik awal untuk menuju destinasi, sehingga P2 dapat dimasukkan ke dalam *List* Tutup.

$$List\ Tutup = [[P2], 0]$$

Karena hanya C15 yang sejalur dilewati, maka C15 dimasukkan ke dalam *List* tutup.

$$List\ Tutup = [[P2', 'C15'], 0.37]$$

Selanjutnya, akan ditentukan titik-titik mana saja yang memiliki jalur utama yang terhubung dengan C15, serta dilakukan perhitungan nilai heuristik untuk masing-masing titik tersebut. Setelah perhitungan, diperoleh hasil berupa panjang rute dan nilai heuristik sebagai berikut:

$$Terpilih = \{ 'C9': (0.24, 1.32), 'C16': (0.44, 1.06) \}$$

Selanjutnya, kita melakukan seleksi melalui perhitungan untuk memilih nilai yang terkecil.

$$f(C9) = g(C9) + h(C9)$$

$$= (0.37 + 0.24) + 1.32$$

$$= 1.93$$

$$f(C16) = g(C16) + h(C16)$$

$$= (0.37 + 0.44) + 1.06$$

$$= 1.87$$

Hasil perhitungan yang didapat menunjukkan bahwa C16 memiliki nilai terendah, sehingga dapat dimasukkan ke dalam *List* tutup sebagai berikut:

$$List\ Tutup = [[P2', 'C15', 'C16'], 0.81]$$

Karena hanya C19 yang sejalur dilewati, maka C19 dimasukkan ke dalam *List* tutup.

$$List\ Tutup = [[P2', 'C15', 'C16', 'C19'], 1.06]$$

C19 terhubung dengan C17 dan C18, maka panjang rute dan nilai heuristinya adalah:

$$\text{Terpilih} = \{ 'C17': (0.08, 0.74), 'C18': (0.34, 0.9) \}$$

$$f(C17) = g(C17) + h(C17)$$

$$= (1.06 + 0.08) + 0.74$$

$$= 1.88$$

$$f(C18) = g(C18) + h(C18)$$

$$= (1.06 + 0.34) + 0.9$$

$$= 2.3$$

Hasil perhitungan yang didapat menunjukkan bahwa C17 memiliki nilai terendah, sehingga dapat dimasukkan ke dalam *List* tutup sebagai berikut:

$$\text{List Tutup} = [['P2', 'C15', 'C16', 'C19', 'C17'], 1.14]$$

Sama seperti dalam proses perhitungan sebelumnya, akan dicari titik-titik yang jalur utamanya terhubung dengan C17, dan hasilnya adalah sebagai berikut:

$$\text{Terpilih} = \{ 'C36': (0.12, 0.62), 'C34': (0.18, 0.76) \}$$

Selanjutnya, kita melakukan seleksi melalui perhitungan untuk memilih nilai yang terkecil.

$$f(C36) = g(C36) + h(C36)$$

$$= (1.14 + 0.12) + 0.62$$

$$= 1.88$$

$$f(C34) = g(C34) + h(C34)$$

$$= (1.14 + 0.18) + 0.76$$

= 2.08

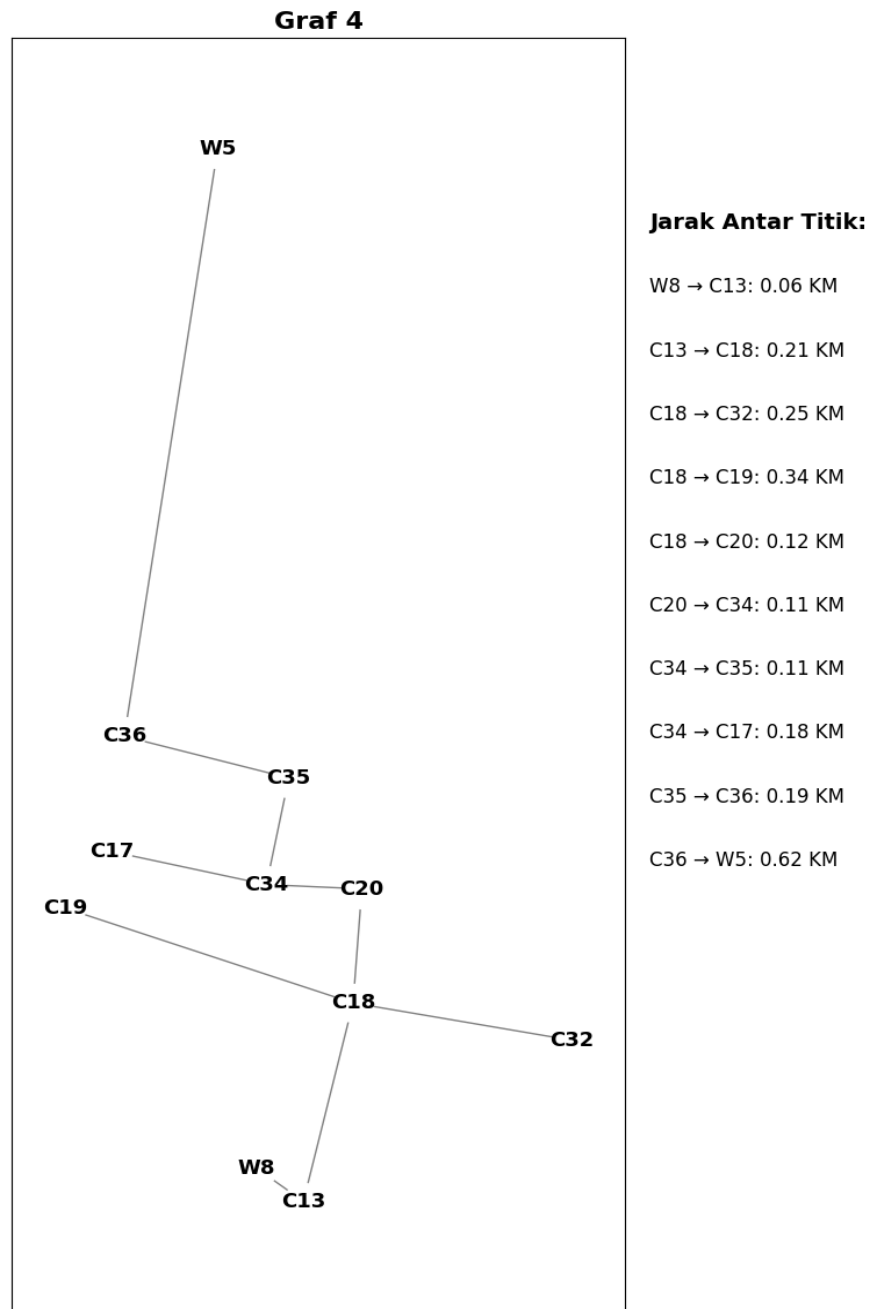
Hasil perhitungan yang didapat menunjukkan bahwa C36 memiliki nilai terendah, sehingga dapat dimasukkan ke dalam *List* tutup sebagai berikut:

List Tutup = [['P2',' C15',' C16',' C19',' C17',' C36'],1.26]

Karena hanya W5 yang sejalur dilewati dan merupakan jalur destinasi akhir yang menandakan telah sampainya dilokasi tujuan, maka W5 dimasukkan ke dalam *List* tutup sekaligus telah sampai dilokasi *cluster* tujuan:

List Tutup = [['P2',' C15',' C16',' C19',' C17' , C36',' W5']1.88]

- Penentuan rute W8→W5



Gambar 4.11 Visualisasi Rute W8 Menuju W5

W8 hanya terhubung dengan C13, oleh karena itu C13 dimasukkan ke dalam *List* tutup.

List Tutup = $[[\text{'W8'}, \text{'C13'}], 0.06]$

Karena hanya C18 yang sejalur dilewati, maka C18 dimasukkan ke dalam *List* tutup.

$$List\ Tutup = [['W8', 'C13', 'C18'], 0.27]$$

Selanjutnya, akan ditentukan titik-titik mana saja yang memiliki jalur utama yang terhubung dengan C18, serta dilakukan perhitungan nilai heuristik untuk masing-masing titik tersebut. Setelah perhitungan, diperoleh hasil berupa p

anjang rute dan nilai heuristik sebagai berikut:

$$Terpilih = \{ 'C32': (0.25, 1.01), 'C19': (0.34, 0.81), 'C20': (0.12, 0.79) \}$$

Selanjutnya, kita melakukan seleksi melalui perhitungan untuk memilih nilai yang terkecil.

$$f(C32) = g(C32) + h(C32)$$

$$= (0.27 + 0.25) + 1.01$$

$$= 1.53$$

$$f(C19) = g(C19) + h(C19)$$

$$= (0.27 + 0.34) + 0.81$$

$$= 1.42$$

$$f(C20) = g(C20) + h(C20)$$

$$= (0.27 + 0.12) + 0.79$$

$$= 1.18$$

Hasil perhitungan yang didapat menunjukkan bahwa C20 memiliki nilai terendah, sehingga dapat dimasukkan ke dalam *List* tutup sebagai berikut:

$$List\ Tutup = [['W8', 'C13', 'C18', 'C20'], 0.39]$$

Karena hanya C34 yang sejalur dilewati, maka C34 dimasukkan ke dalam *List* tutup.

$$List\ Tutup = [['W8', 'C13', 'C18', 'C20', 'C34'], 0.5]$$

C34 terhubung dengan C35 dan C17, maka panjang rute dan nilai heuristinya adalah:

$$Terpilih = \{ 'C35': (0.11, 0.66), 'C17': (0.18, 0.74) \}$$

$$f(C35) = g(C35) + h(C35)$$

$$= (0.5 + 0.11) + 0.66$$

$$= 1.27$$

$$f(C17) = g(C17) + h(C17)$$

$$= (0.5 + 0.18) + 0.74$$

$$= 1.42$$

Hasil perhitungan yang didapat menunjukkan bahwa C35 memiliki nilai terendah, sehingga dapat dimasukkan ke dalam *List* tutup sebagai berikut:

$$List\ Tutup = [['W8', 'C13', 'C18', 'C20', 'C34', 'C35'], 0.61]$$

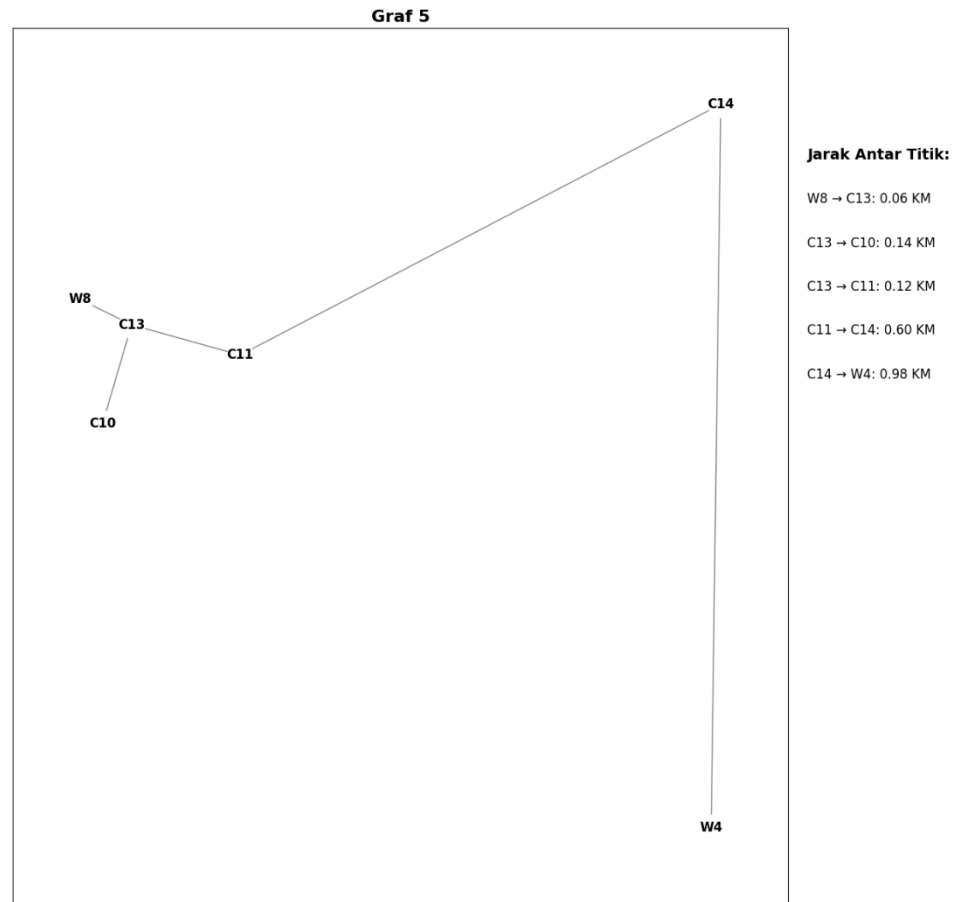
Karena hanya C36 yang sejalur dilewati, maka C36 dimasukkan ke dalam *List* tutup.

$$List\ Tutup = [['W8', 'C13', 'C18', 'C20', 'C34', 'C35', 'C36'], 0.8]$$

Karena hanya W5 yang sejalur dilewati dan merupakan jalur destinasi akhir yang menandakan telah sampainya dilokasi tujuan, maka W5 dimasukkan ke dalam *List* tutup sekaligus telah sampai dilokasi *cluster* tujuan:

$$List\ Tutup = [['W8', 'C13', 'C18', 'C20', 'C34', 'C35', 'C36', 'W5'], 1.42]$$

- **Penentuan rute W8→W4**



Gambar 4.12 Visualisasi Rute W8 Menuju W4

W8 hanya terhubung dengan C13, oleh karena itu C13 dimasukkan ke dalam *List* tutup.

$$List\ Tutup = [[W8, C13], 0.06]$$

Selanjutnya, akan ditentukan titik-titik mana saja yang memiliki jalur utama yang terhubung dengan C13, serta dilakukan perhitungan nilai heuristik untuk masing-masing titik tersebut. Setelah perhitungan, diperoleh hasil berupa panjang rute dan nilai heuristik sebagai berikut:

$$Terpilih = \{C10: (0.14, 0.83), C11: (0.12, 0.80)\}$$

Selanjutnya, kita melakukan seleksi melalui perhitungan untuk memilih nilai yang terkecil.

$$f(C10) = g(C10) + h(C10)$$

$$= (0.06 + 0.14) + 0.83$$

$$= 1.03$$

$$f(C11) = g(C11) + h(C11)$$

$$= (0.06 + 0.12) + 0.80$$

$$= 0.98$$

Hasil perhitungan yang didapat menunjukkan bahwa C11 memiliki nilai terendah, sehingga dapat dimasukkan ke dalam *List* tutup sebagai berikut:

$$List\ Tutup = [['W8', 'C13', 'C11'], 0.18]$$

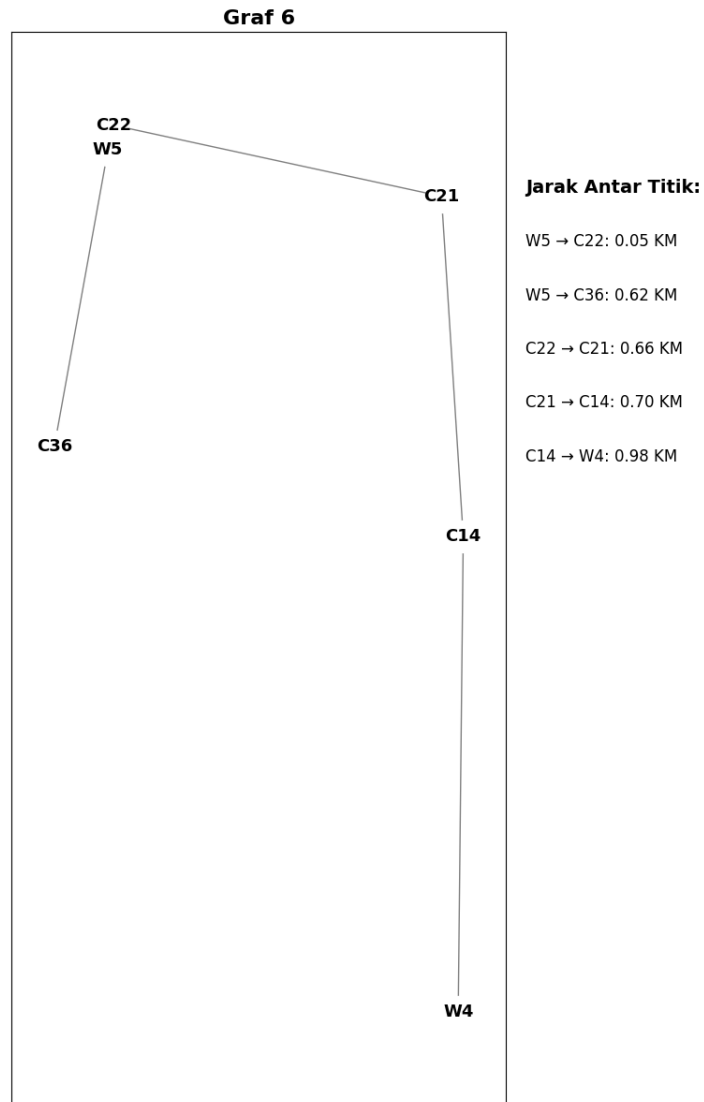
Karena hanya C14 yang sejalur dilewati, maka C14 dimasukkan ke dalam *List* tutup.

$$List\ Tutup = [['W8', 'C13', 'C11', 'C14'], 0.78]$$

Karena hanya W4 yang sejalur dilewati dan merupakan jalur destinasi akhir yang menandakan telah sampainya dilokasi tujuan, maka W4 dimasukkan ke dalam *List* tutup sekaligus telah sampai dilokasi *cluster* tujuan:

$$List\ Tutup = [['W8', 'C13', 'C11', 'C14', 'W4'], 1.76]$$

- **Penentuan rute W5→W4**



Gambar 4.13 Visualisasi Rute W5 Menuju W4

W5 terhubung dengan titik C22 dan C36, maka perhitungannya adalah sebagai berikut:

$$\text{Terpilih} = \{ 'C22': (0.05, 1.95), 'C36': (0.62, 1.41) \}$$

Selanjutnya, kita melakukan seleksi melalui perhitungan untuk memilih nilai yang terkecil.

$$f(C22) = g(C22) + h(C22)$$

$$= (0 + 0.05) + 1.95$$

$$= 2$$

$$f(C36) = g(C36) + h(C36)$$

$$= (0 + 0.62) + 1.41$$

$$= 2.03$$

Hasil perhitungan yang didapat menunjukkan bahwa C22 memiliki nilai terendah, sehingga dapat dimasukkan ke dalam *List* tutup sebagai berikut:

$$List\ Tutup = [['W5', 'C22'], 0.05]$$

Karena hanya C21 yang sejalur dilewati, maka C21 dimasukkan ke dalam *List* tutup, begitupun juga dengan C14.

$$List\ Tutup = [['W5', 'C22', 'C21'], 0.71]$$

$$List\ Tutup = [['W5', 'C22', 'C21', 'C14'], 1.41]$$

Karena hanya W4 yang sejalur dilewati dan merupakan jalur destinasi akhir yang menandakan telah sampainya dilokasi tujuan, maka W4 dimasukkan ke dalam *List* tutup sekaligus telah sampai dilokasi *cluster* tujuan:

$$List\ Tutup = [['W5', 'C22', 'C21', 'C14', 'W4'], 2.39]$$

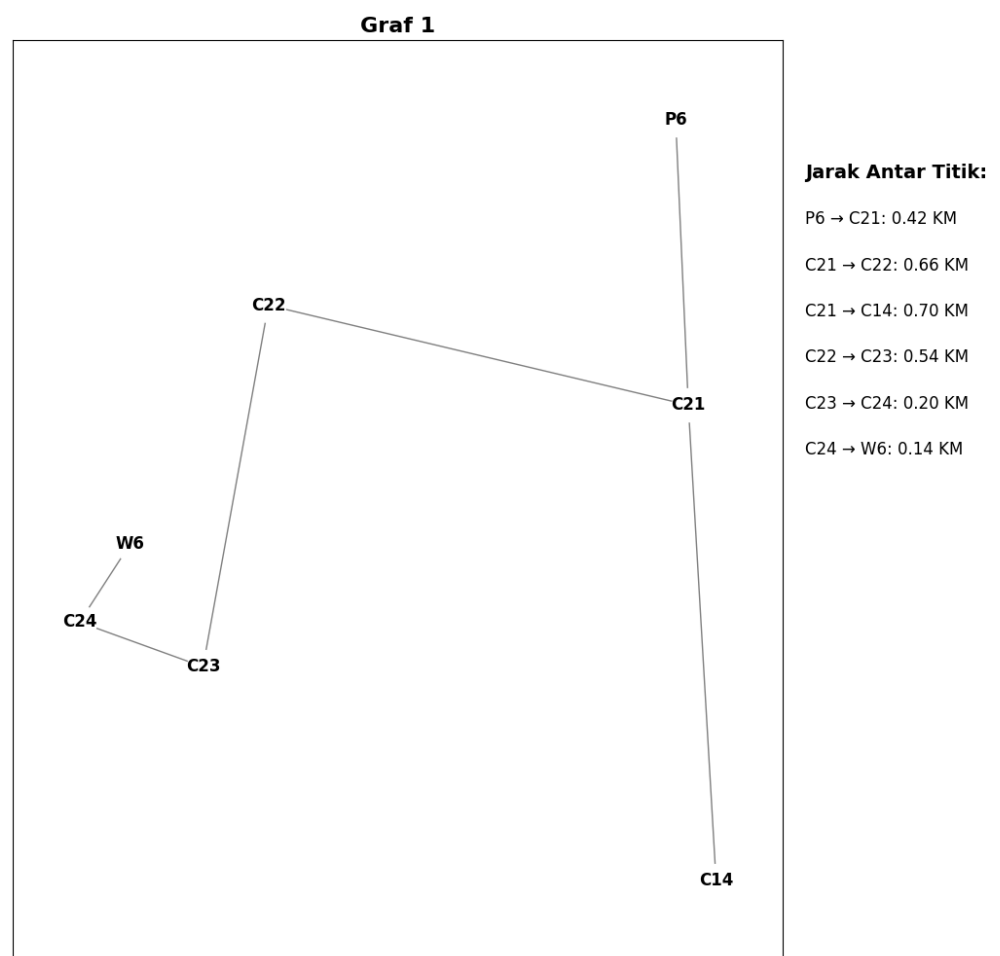
Rute terakhir yang tercantum dalam *List* tutup di atas dapat dimasukkan ke dalam list rute. Selanjutnya, total jarak minimum yang dapat ditempuh akan dijumlahkan.

$$List\ Rute = [['P2', 'C15', 'C9', 'C10', 'C13', 'W8'], ['W8', 'C13', 'C18', 'C20', 'C34', 'C35', 'C36', 'W5'], ['W5', 'C22', 'C21', 'C14', 'W4'], 4.78]$$

Berdasarkan *List* rute di atas, perjalanan dimulai dari titik P2 dengan tujuan pertama menuju W8. Selanjutnya, perjalanan dilanjutkan ke destinasi kedua, yaitu W5. Setelah melewati W5, rute diteruskan ke destinasi terakhir, yaitu W4.

3. Tittik Awal P6

- **Penentuan rute P6→W6**



Gambar 4.14 Visualisasi Rute P6 Menuju W6

Proses pencarian rute kali ini serupa dengan perhitungan rute sebelumnya. P6 berfungsi sebagai titik awal untuk mencapai destinasi, sehingga P6 dapat dimasukkan ke dalam *List* Tutup.

$$List\ Tutup = [['P6'], 0]$$

Karena hanya C21 yang sejalur dilewati, maka C21 dimasukkan ke dalam *List* tutup.

$$List\ Tutup = [['P6', 'C21'], 0.42]$$

Selanjutnya, akan ditentukan titik-titik mana saja yang memiliki jalur utama yang terhubung dengan C21, serta dilakukan perhitungan nilai heuristik untuk masing-masing titik tersebut. Setelah perhitungan, diperoleh hasil berupa panjang rute dan nilai heuristik sebagai berikut:

$$Terpilih = \{ 'C22': (0.66, 0.41), 'C14': (0.7, 1.03) \}$$

Nilai 0.42 didapatkan dari perhitungan *List* Tutup Sebelumnya, dan 0.66 dari perhitungan jarak antara C21 ke C22 yang kemudian dijumlahkan dan menghasilkan $g(C22)$ yang merupakan biaya jalur dari titik awal P6 ke C22, lalu untuk heuristik $h(C22)$ menghasilkan 0.41 yang merupakan perhitungan dari C22 ke tujuan W6. Selanjutnya nilai 0.42 sama saja didapatkan dari perhitungan *List* Tutup Sebelumnya, dan 0.7 dari perhitungan jarak antara C21 ke C14 yang kemudian dijumlahkan dan menghasilkan $g(C14)$ yang merupakan biaya jalur dari titik awal P6 ke C14, lalu untuk heuristik $h(C14)$ menghasilkan 1.03 yang merupakan perhitungan dari C14 ke tujuan W6. Kemudian, kita melakukan seleksi menggunakan perhitungan untuk menentukan nilai yang paling kecil di antara semua nilai yang ada.

$$f(C22) = g(C22) + h(C22)$$

$$= (0.42 + 0.66) + 0.41$$

$$= 1.49$$

$$\begin{aligned}
 f(C14) &= g(C14) + h(C14) \\
 &= (0.42 + 0.7) + 1.03 \\
 &= 2.15
 \end{aligned}$$

Hasil perhitungan yang didapat menunjukkan bahwa C22 memiliki nilai terendah, sehingga dapat dimasukkan ke dalam *List* tutup sebagai berikut:

$$List\ Tutup = [['P6', 'C21', 'C22'], 1.08]$$

Karena hanya C23 yang sejalur dilewati, maka C23 dimasukkan ke dalam *List* tutup, begitupun juga dengan C24.

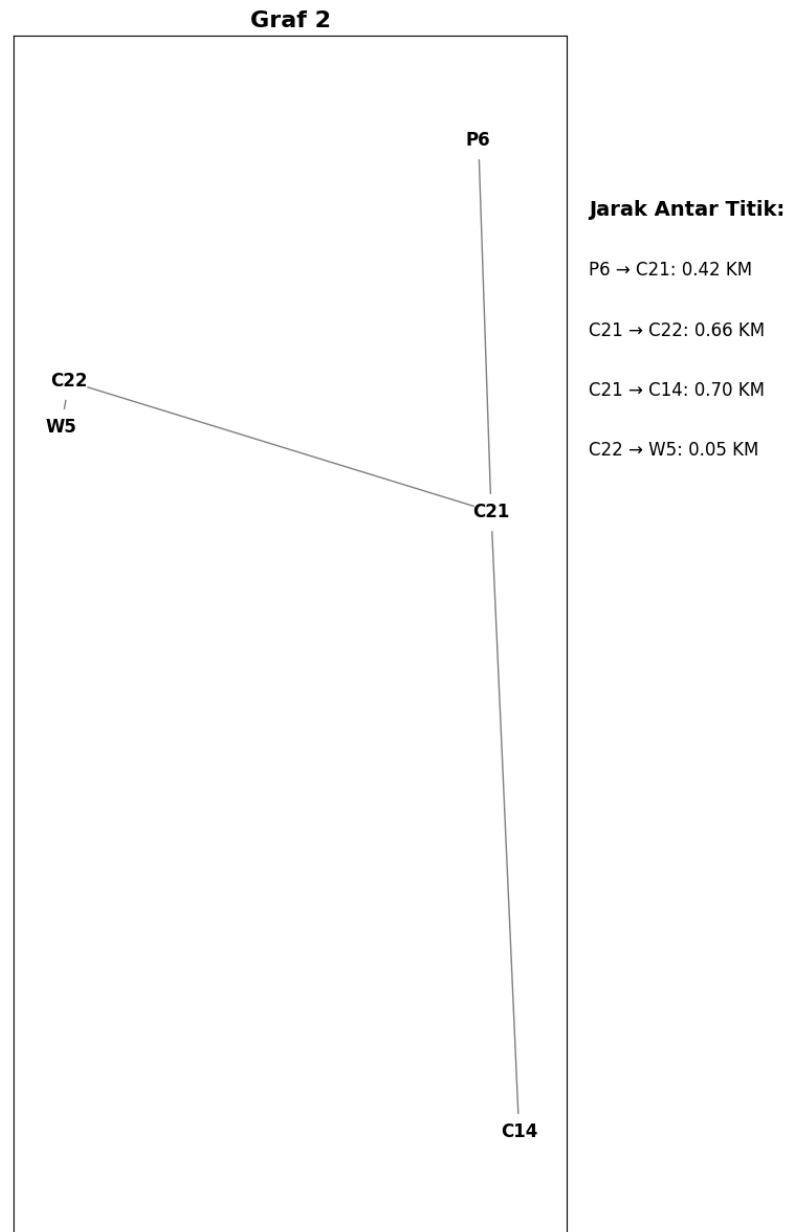
$$List\ Tutup = [['P6', 'C21', 'C22', 'C23'], 1.62]$$

$$List\ Tutup = [['P6', 'C21', 'C22', 'C23', 'C24'], 1.82]$$

Karena hanya W6 yang sejalur dilewati dan merupakan jalur destinasi akhir yang menandakan telah sampainya dilokasi tujuan, maka W6 dimasukkan ke dalam *List* tutup sekaligus telah sampai dilokasi *cluster* tujuan:

$$List\ Tutup = [['P6', 'C21', 'C22', 'C23', 'C24', 'W6'], 1.96]$$

- **Penentuan rute P6→W5**



Gambar 4.15 Visualisasi Rute P6 Menuju W5

Proses pencarian rute kali ini serupa dengan perhitungan rute sebelumnya. P6 berfungsi sebagai titik awal untuk mencapai destinasi, sehingga P6 dapat dimasukkan ke dalam *List Tutup*.

$$List\ Tutup = [['P6'], 0]$$

Karena hanya C21 yang sejalur dilewati, maka C21 dimasukkan ke dalam List tutup.

$$List\ Tutup = [[P6', C21'], 0.42]$$

Selanjutnya, akan ditentukan titik-titik mana saja yang memiliki jalur utama yang terhubung dengan C21, serta dilakukan perhitungan nilai heuristik untuk masing-masing titik tersebut. Setelah perhitungan, diperoleh hasil berupa panjang rute dan nilai heuristik sebagai berikut:

$$Terpilih = \{C22': (0.66, 0.05), C14': (0.7, 1.06)\}$$

Selanjutnya, kita melakukan seleksi melalui perhitungan untuk memilih nilai yang terkecil.

$$f(C22) = g(C22) + h(C22)$$

$$= (0.42 + 0.66) + 0.05$$

$$= 1.13$$

$$f(C14) = g(C14) + h(C14)$$

$$= (0.42 + 0.7) + 1.06$$

$$= 2.18$$

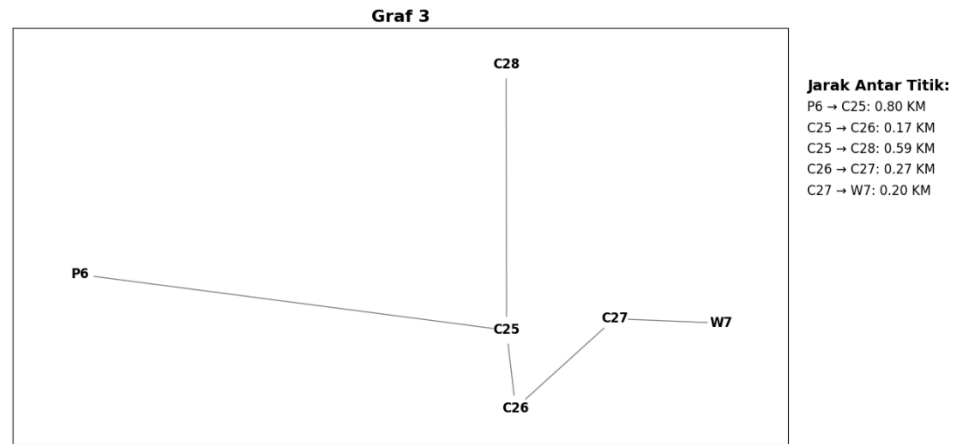
Hasil perhitungan yang didapat menunjukkan bahwa C22 memiliki nilai terendah, sehingga dapat dimasukkan ke dalam list tutup sebagai berikut:

$$List\ Tutup = [[P6', C21', C22'], 1.08]$$

Karena hanya W5 yang sejalur dilewati dan merupakan jalur destinasi akhir yang menandakan telah sampainya dilokasi tujuan, maka W5 dimasukkan ke dalam List tutup sekaligus telah sampai dilokasi *cluster* tujuan:

$$List\ Tutup = [['P6', 'C21', 'C22', 'W5'], 1.13]$$

- **Penentuan rute P6→W7**



Gambar 4.16 Visualisasi Rute P6 Menuju W7

Proses pencarian rute kali ini serupa dengan perhitungan rute sebelumnya. P6 berfungsi sebagai titik awal untuk mencapai destinasi, sehingga P6 dapat dimasukkan ke dalam *List Tutup*.

$$List\ Tutup = [['P6'], 0]$$

Karena hanya C25 yang sejalur dilewati, maka C25 dimasukkan ke dalam *List tutup*.

$$List\ Tutup = [['P6', 'C25'], 0.8]$$

Selanjutnya, akan ditentukan titik-titik mana saja yang memiliki jalur utama yang terhubung dengan C25, serta dilakukan perhitungan nilai heuristik untuk masing-masing titik tersebut. Setelah perhitungan, diperoleh hasil berupa panjang rute dan nilai heuristik sebagai berikut:

$$Terpilih = \{ 'C26': (0.17, 0.43), 'C28': (0.59, 0.7) \}$$

Selanjutnya, kita melakukan seleksi melalui perhitungan untuk memilih nilai yang terkecil.

$$f(C26) = g(C26) + h(C26)$$

$$= (0.8 + 0.17) + 0.43$$

$$= 1.4$$

$$f(C28) = g(C28) + h(C28)$$

$$= (0.8 + 0.59) + 0.7$$

$$= 2.09$$

Hasil perhitungan yang didapat menunjukkan bahwa C26 memiliki nilai terendah, sehingga dapat dimasukkan ke dalam *List* tutup sebagai berikut:

$$List\ Tutup = [['P6', 'C25', 'C26'], 0.97]$$

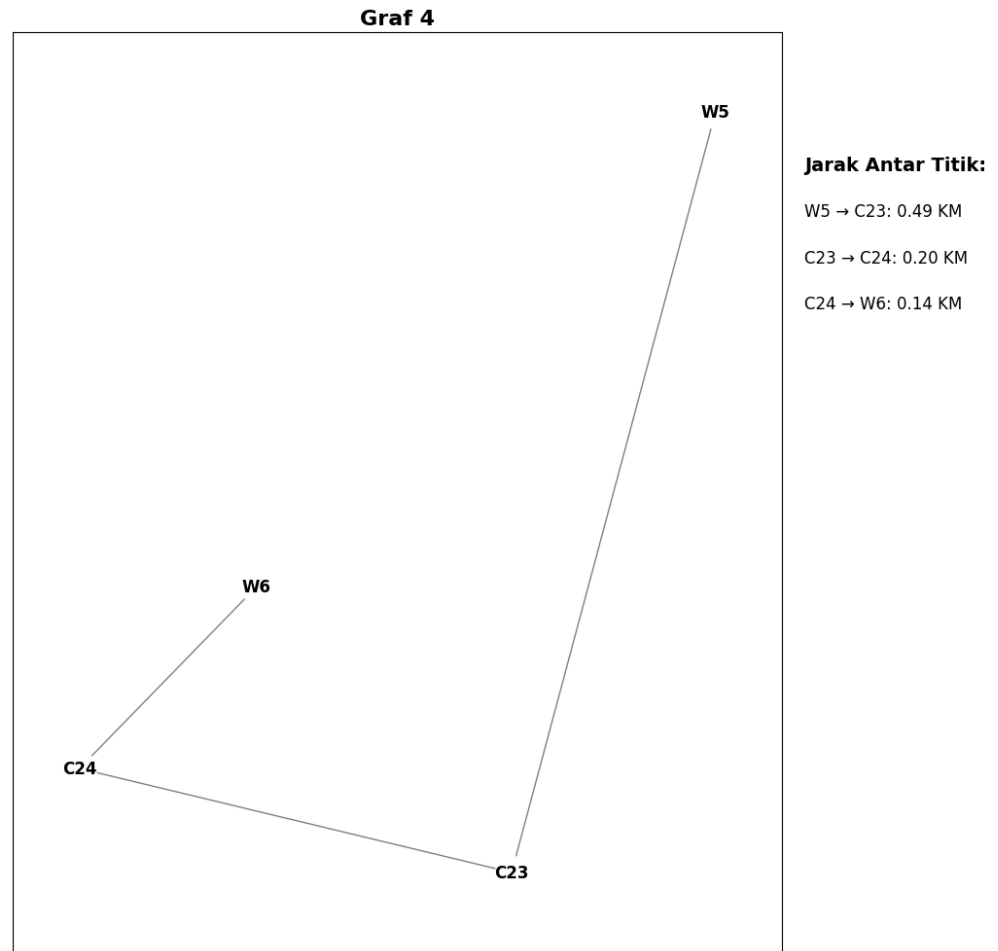
Karena hanya C27 yang sejalur dilewati, maka C27 dimasukkan ke dalam *List* tutup.

$$List\ Tutup = [['P6', 'C25', 'C26', 'C27'], 1.24]$$

Karena hanya W7 yang sejalur dilewati dan merupakan jalur destinasi akhir yang menandakan telah sampainya dilokasi tujuan, maka W7 dimasukkan ke dalam *List* tutup sekaligus telah sampai dilokasi *cluster* tujuan:

$$List\ Tutup = [['P6', 'C25', 'C26', 'C27', 'W7'], 1.44]$$

- **Penentuan rute W5→W6**



Gambar 4.17 Visualisasi Rute W5 Menuju W6

W5 hanya terhubung dengan C23, oleh karena itu C23 dimasukkan ke dalam *List* tutup.

$$List\ Tutup = [['W5', 'C23'], 0.49]$$

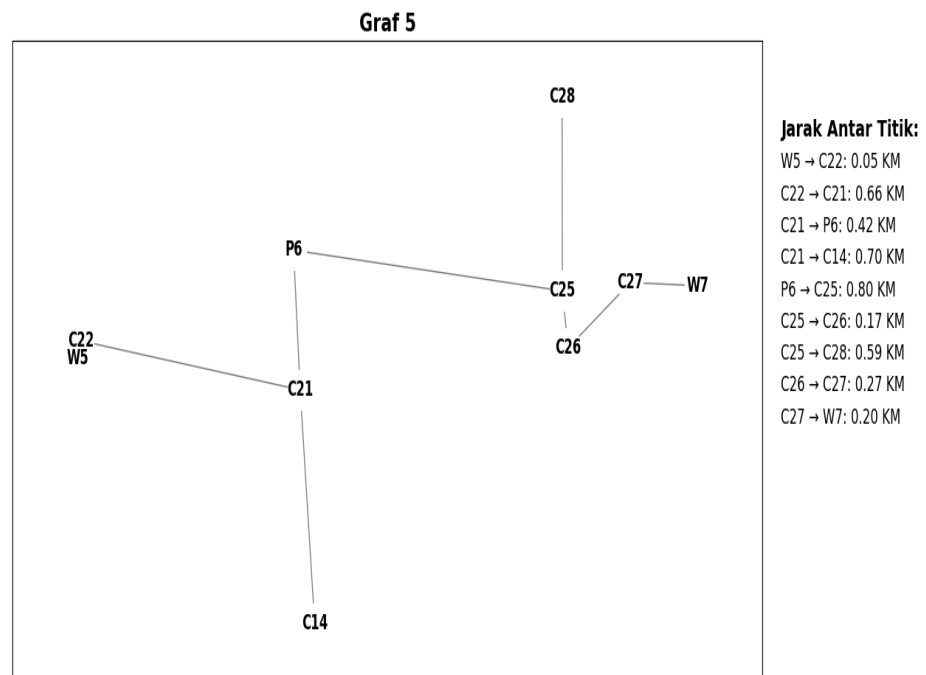
Karena hanya C24 yang sejalur dilewati, maka C24 dimasukkan ke dalam *List* tutup.

$$List\ Tutup = [['W5', 'C23', 'C24'], 0.69]$$

Hanya W8 yang sejalur dilewati dan merupakan jalur destinasi akhir yang menandakan telah sampainya dilokasi tujuan, maka W8 dimasukkan ke dalam *List* tutup sekaligus telah sampai dilokasi *cluster* tujuan:

$$List\ Tutup = [['W5', 'C23', 'C24', 'W6'], 0.83]$$

- **Penentuan rute W5→W7**



Gambar 4.18 Visualisasi Rute W5 Menuju W7

W5 hanya terhubung dengan C22, oleh karena itu C22 dimasukkan ke dalam *List* tutup.

$$List\ Tutup = [['W5', 'C22'], 0.05]$$

Karena hanya C21 yang sejalur dilewati, maka C21 dimasukkan ke dalam *List* tutup.

$$List\ Tutup = [['W5', 'C22', 'C21'], 0.71]$$

Selanjutnya, akan ditentukan titik-titik mana saja yang memiliki jalur utama yang terhubung dengan C21, serta dilakukan perhitungan nilai heuristik untuk masing-masing titik tersebut. Setelah perhitungan, diperoleh hasil berupa panjang rute dan nilai heuristik sebagai berikut:

Terpilih = {'P6': (0.42, 0.003781), 'C14': (0.7, 0.006338)}

Selanjutnya, kita melakukan seleksi melalui perhitungan untuk memilih nilai yang terkecil.

$$f(P6) = g(P6) + h(P6)$$

$$= (0.71 + 0.42) + 1.20$$

$$= 2.33$$

$$f(C14) = g(C14) + h(C14)$$

$$= (0.71 + 0.7) + 1.52$$

$$= 2.93$$

Hasil perhitungan yang didapat menunjukkan bahwa P6 memiliki nilai terendah, sehingga dapat dimasukkan ke dalam *List* tutup sebagai berikut:

$$List\ Tutup = [['W5', 'C22', 'C21', 'P6'], 1.13]$$

Karena hanya C25 yang sejalur dilewati, maka C25 dimasukkan ke dalam *List* tutup.

$$List\ Tutup = [['W5', 'C22', 'C21', 'P6', 'C25'], 1.93]$$

C25 terhubung dengan C26 dan C28 yang selanjutnya akan dilakukan seleksi melalui perhitungan untuk memilih nilai yang terkecil.

Terpilih = {'C26': (0.17, 0.43), 'C28': (0.59, 0.7)}

$$f(C26) = g(C26) + h(C26)$$

$$= (1.93 + 0.17) + 0.43$$

$$= 2.53$$

$$f(C28) = g(C28) + h(C28)$$

$$= (1.93 + 0.59) + 0.7$$

$$= 3.22$$

Hasil perhitungan yang didapat menunjukkan bahwa C26 memiliki nilai terendah, sehingga dapat dimasukkan ke dalam *List* tutup sebagai berikut:

$$List\ Tutup = [['W5', 'C22', 'C21', 'P6', 'C25', 'C26'], 2.1]$$

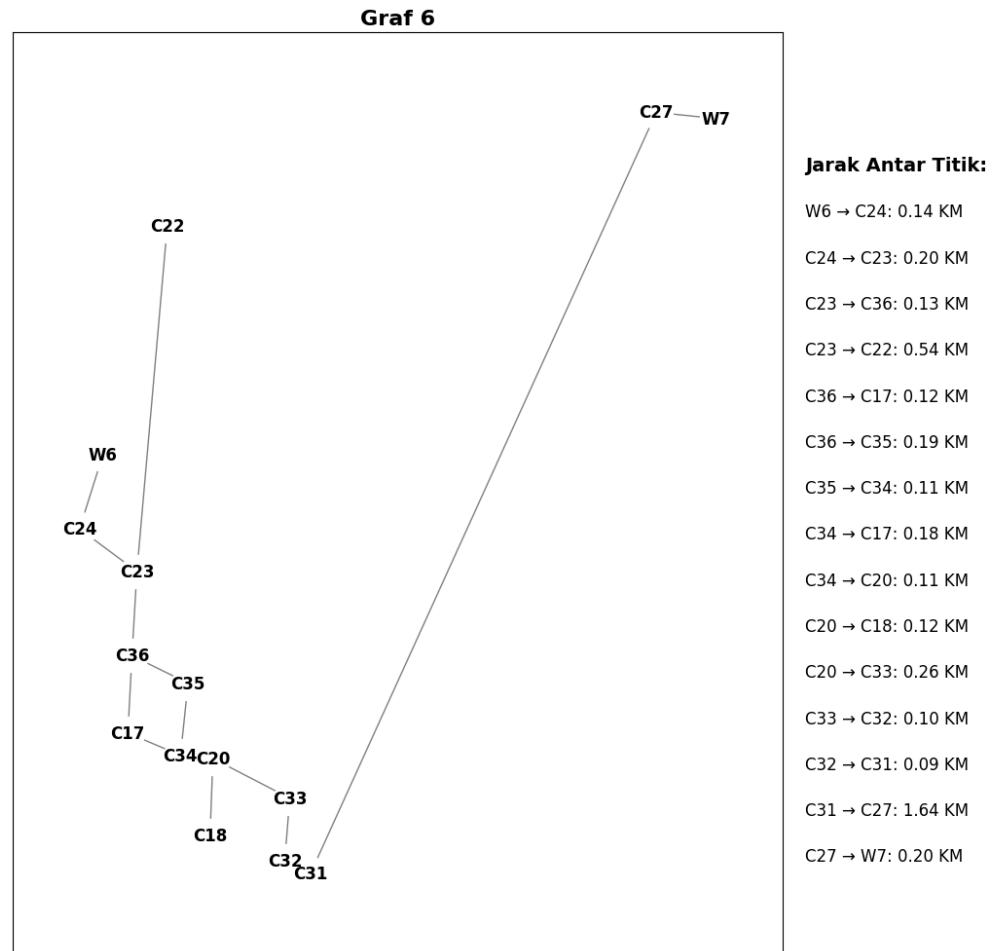
Karena hanya C27 yang sejalur dilewati, maka C27 dimasukkan ke dalam *List* tutup.

$$List\ Tutup = [['W5', 'C22', 'C21', 'P6', 'C25', 'C26', 'C27'], 2.37]$$

Karena hanya W7 yang sejalur dilewati dan merupakan jalur destinasi akhir yang menandakan telah sampainya dilokasi tujuan, maka W7 dimasukkan ke dalam *List* tutup sekaligus telah sampai dilokasi *cluster* tujuan:

$$List\ Tutup = [['W5', 'C22', 'C21', 'P6', 'C25', 'C26', 'C27', 'W7'], 2.57]$$

- **Penentuan rute W6→W7**



Gambar 4.19 Visualisasi Rute W6 Menuju W7

W6 hanya terhubung dengan C24, oleh karena itu C24 dimasukkan ke dalam *List* tutup.

$$List\ Tutup = [['W6', 'C24'], 0.14]$$

Karena hanya C23 yang sejalur dilewati, maka C23 dimasukkan ke dalam *List* tutup.

$$List\ Tutup = [['W6', 'C24', 'C23'], 0.34]$$

Selanjutnya, akan ditentukan titik-titik mana saja yang memiliki jalur utama yang terhubung dengan C23, serta dilakukan perhitungan nilai

heuristik untuk masing-masing titik tersebut. Setelah perhitungan, diperoleh hasil berupa panjang rute dan nilai heuristik sebagai berikut:

$$\text{Terpilih} = \{ 'C36': (0.13, 2.11), 'C22': (0.54, 1.83) \}$$

Selanjutnya, kita melakukan seleksi melalui perhitungan untuk memilih nilai yang terkecil.

$$f(C36) = g(C36) + h(C36)$$

$$= (0.34 + 0.13) + 2.11$$

$$= 2.58$$

$$f(C22) = g(C22) + h(C22)$$

$$= (0.34 + 0.54) + 1.83$$

$$= 2.71$$

Hasil perhitungan yang didapat menunjukkan bahwa C36 memiliki nilai terendah, sehingga dapat dimasukkan ke dalam *List* tutup sebagai berikut:

$$\text{List Tutup} = [['W6', 'C24', 'C23', 'C36'], 0.47]$$

C36 terhubung dengan C17 dan C35 yang selanjutnya akan dilakukan seleksi melalui perhitungan untuk memilih nilai yang terkecil.

$$\text{Terpilih} = \{ 'C17': (0.12, 2.17), 'C35': (0.19, 1.96) \}$$

$$f(C17) = g(C17) + h(C17)$$

$$= (0.47 + 0.12) + 2.17$$

$$= 2.76$$

$$f(C35) = g(C35) + h(C35)$$

$$= (0.47 + 0.19) + 1.96$$

$$= 2.62$$

Hasil perhitungan yang didapat menunjukkan bahwa C35 memiliki nilai terendah, sehingga dapat dimasukkan ke dalam *List* tutup sebagai berikut:

$$List\ Tutup = [['W6', 'C24', 'C23', 'C36', 'C35'], 0.66]$$

Karena hanya C34 yang sejalur dilewati, maka C34 dimasukkan ke dalam *List* tutup.

$$List\ Tutup = [['W6', 'C24', 'C23', 'C36', 'C35', 'C34'], 0.77]$$

C34 terhubung dengan C17 dan C20 yang selanjutnya akan dilakukan seleksi melalui perhitungan untuk memilih nilai yang terkecil.

$$Terpilih = \{ 'C17': (0.18, 2.17), 'C20': (0.11, 1.94) \}$$

$$f(C17) = g(C17) + h(C17)$$

$$= (0.77 + 0.18) + 2.17$$

$$= 3.12$$

$$f(C20) = g(C20) + h(C20)$$

$$= (0.77 + 0.11) + 1.94$$

$$= 2.82$$

Hasil perhitungan yang didapat menunjukkan bahwa C20 memiliki nilai terendah, sehingga dapat dimasukkan ke dalam *List* tutup sebagai berikut:

$$List\ Tutup = [['W6', 'C24', 'C23', 'C36', 'C35', 'C34', 'C20'], 0.88]$$

C20 terhubung dengan C18 dan C33 yang selanjutnya akan dilakukan seleksi melalui perhitungan untuk memilih nilai yang terkecil.

$$Terpilih = \{ 'C18': (0.12, 2.01), 'C33': (0.26, 1.76) \}$$

$$f(C18) = g(C18) + h(C18)$$

$$= (0.88 + 0.12) + 2.01$$

$$= 3.01$$

$$f(C33) = g(C33) + h(C33)$$

$$= (0.88 + 0.26) + 1.76$$

$$= 2.9$$

Hasil perhitungan yang didapat menunjukkan bahwa C33 memiliki nilai terendah, sehingga dapat dimasukkan ke dalam *List* tutup sebagai berikut:

$$List\ Tutup = [['W6', 'C24', 'C23', 'C36', 'C35', 'C34', 'C20', 'C33'], 1.14]$$

Karena hanya C32 yang sejalur dilewati, maka C32 dimasukkan ke dalam *List* tutup, begitupun juga C31, dan C27.

$$List\ Tutup = [['W6', 'C24', 'C23', 'C36', 'C35', 'C34', 'C20', 'C33', \\ 'C32'], 1.24]$$

$$List\ Tutup = [['W6', 'C24', 'C23', 'C36', 'C35', 'C34', 'C20', 'C33', \\ 'C32', 'C31'], 1.33]$$

$$List\ Tutup = [['W6', 'C24', 'C23', 'C36', 'C35', 'C34', 'C20', 'C33', \\ 'C32', 'C31', 'C27'], 2.97]$$

Karena hanya W7 yang sejalur dilewati dan merupakan jalur destinasi akhir yang menandakan telah sampainya dilokasi tujuan, maka W7 dimasukkan ke dalam *List* tutup sekaligus telah sampai dilokasi *cluster* tujuan:

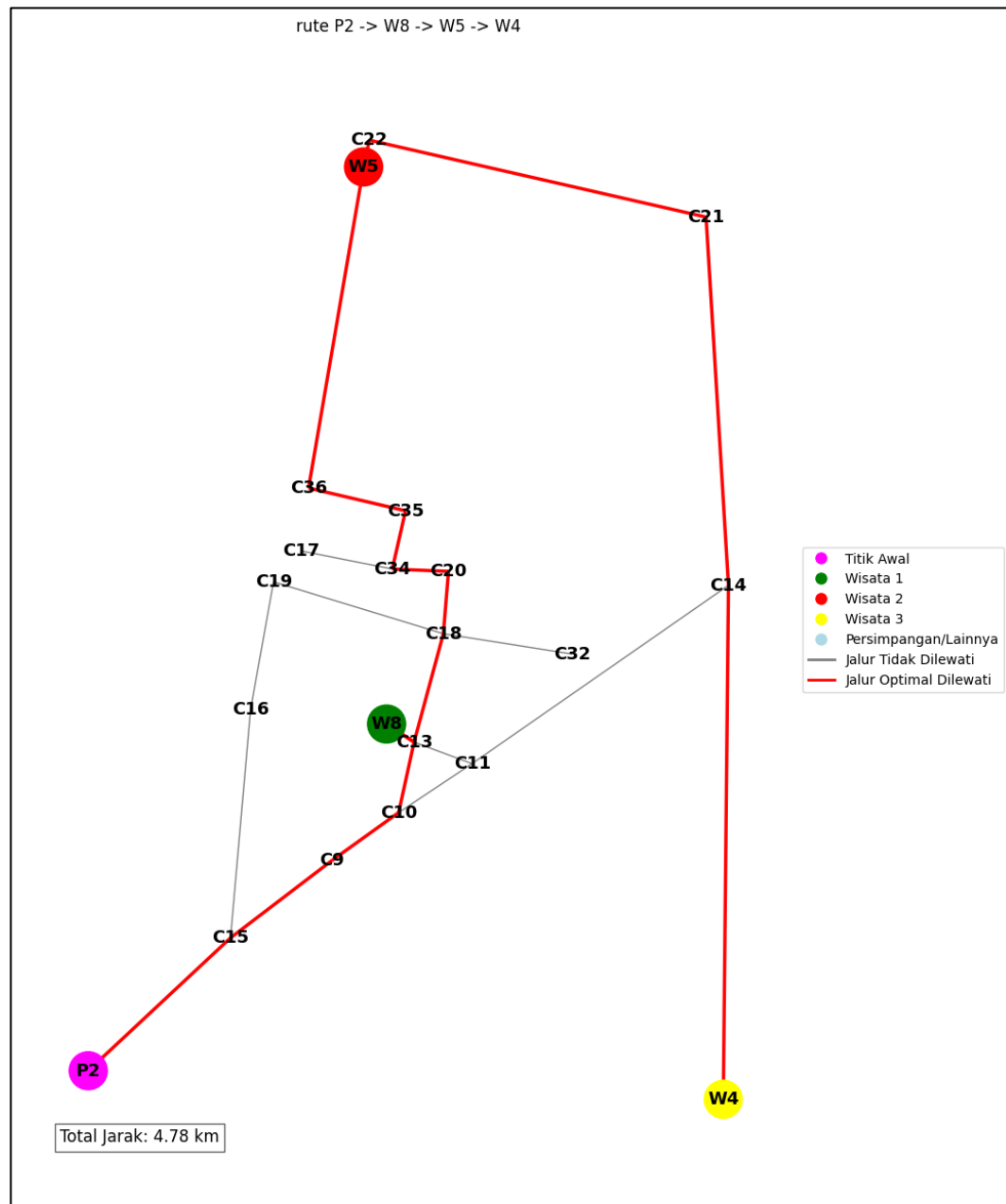
$$List\ Tutup = [['W6', 'C24', 'C23', 'C36', 'C35', 'C34', 'C20', 'C33', \\ 'C32', 'C31', 'C27', 'W7'], 3.17]$$

Rute terakhir yang tercantum dalam *List* tutup di atas dapat dimasukkan ke dalam list rute. Selanjutnya, total jarak minimum yang dapat ditempuh akan dijumlahkan.

$$\begin{aligned} \text{List Rute} = & [[P6, C21, C22, W5], [W5, C23, C24, W6], \\ & [W6, C24, C23, C36, C35, C34, C20, C33, \\ & C32, C31, C27, W7], 5.13] \end{aligned}$$

Berdasarkan *List* rute di atas, perjalanan dimulai dari titik P6 dengan tujuan pertama menuju W5. Selanjutnya, perjalanan dilanjutkan ke destinasi kedua, yaitu W6. Setelah melewati W6, rute diteruskan ke destinasi terakhir, yaitu W7.

4. Visualisasi Route



Gambar 4.20 Hasil Visualisasi Route Terpendek

Gambar 4.20 menampilkan visualisasi dari sampel titik awal P2. Pada visualisasi ini, semua titik persimpangan ditandai dengan simbol C, sementara titik-titik destinasi wisata budaya yang dipilih diwarnai hijau. Garis-garis dengan panah menunjukkan bahwa graf tersebut adalah graf berarah. Output hasil visualisasi ini diberikan warna-warna khusus agar mudah dibedakan, di mana

warna ungu menyimbolkan titik awal perjalanan yang telah dimasukkan pada tahap awal, dan warna hijau, merah, kuning menandakan titik-titik yang akan menjadi tujuan destinasi wisata. Garis penghubung diberi warna merah untuk menunjukkan jalur yang efisien dan dipilih untuk dilalui, sementara garis abu menunjukkan jalur yang tidak efisien. Visualisasi ini tidak hanya menunjukkan rute-rute yang mungkin ditempuh, tetapi juga membedakan antara jalur-jalur yang efisien dan tidak efisien melalui penggunaan warna pada garis penghubung.

4.3 Hasil Pengujian

Penelitian mengenai rute terpendek ini akan diuji dan divalidasi menggunakan Google Maps untuk beberapa sampel destinasi wisata di Kabupaten Mojokerto. Terdapat beberapa alasan mengapa penelitian ini lebih memilih menggunakan algoritma *A-Star*. Sebelumnya, metode lain seperti algoritma *Greedy* dan *Dijkstra* telah dicoba dihitung dengan bantuan Google Colab, namun setelah dilakukan perbandingan hasil, *A-Star* menunjukkan performa yang lebih baik. Hal ini dapat dilihat pada tabel di bawah yang memperlihatkan keunggulan hasil dari *A-Star* dibandingkan dengan metode lainnya.

Tabel 4.4 Tabel Hasil Perbandingan dengan Algoritma Lainnya

Simbol	Percobaan <i>A-Star</i> (KM)	Simbol	Percobaan <i>Dijkstra</i> (KM)	Simbol	Percobaan <i>Greedy</i> (KM)
P2 → W8	0.97	P2 → W8	0.96	P2 → W8	0.98
P2 → W4	2.52	P2 → W4	2.52	P2 → W4	2.61
P2 → W5	1.88	P2 → W5	1.88	P2 → W5	2.30
W8 → W5	1.42	W8 → W5	1.43	W8 → W5	2.00
W8 → W4	1.76	W8 → W4	1.76	W8 → W4	1.76
W5 → W4	2.39	W5 → W4	2.40	W4 → W5	2.40
Total Jarak:	4.78	Total Jarak:	4.79	Total Jarak:	5.14
P6 → W6	1.96	P6 → W6	1.97	P6 → W6	1.97

Simbol	Percobaan <i>A-Star</i> (KM)	Simbol	Percobaan <i>Dijkstra</i> (KM)	Simbol	Percobaan <i>Greedy</i> (KM)
P6 → W5	1.13	P6 → W5	1.14	P6 → W5	1.14
P6 → W7	1.44	P6 → W7	1.45	P6 → W7	1.45
W5 → W6	0.83	W5 → W6	0.83	W5 → W6	0.83
W5 → W7	2.57	W5 → W7	2.58	W5 → W7	2.58
W6 → W7	3.17	W6 → W7	3.54	W6 → W7	4.53
Total Jarak:	5.13	Total Jarak:	5.51	Total Jarak:	6.5
P1 → W1	1.72	P1 → W1	1.72	P1 → W1	1.72
P1 → W2	5.37	P1 → W2	5.38	P1 → W2	6.10
P1 → W3	4.63	P1 → W3	4.91	P1 → W3	4.91
W1 → W2	4.41	W1 → W2	4.42	W1 → W2	5.13
W1 → W3	3.94	W1 → W3	3.95	W1 → W3	4.50
W3 → W2	4.13	W3 → W2	4.14	W3 → W2	4.42
Total Jarak:	9.79	Total Jarak:	9.81	Total Jarak:	10.64

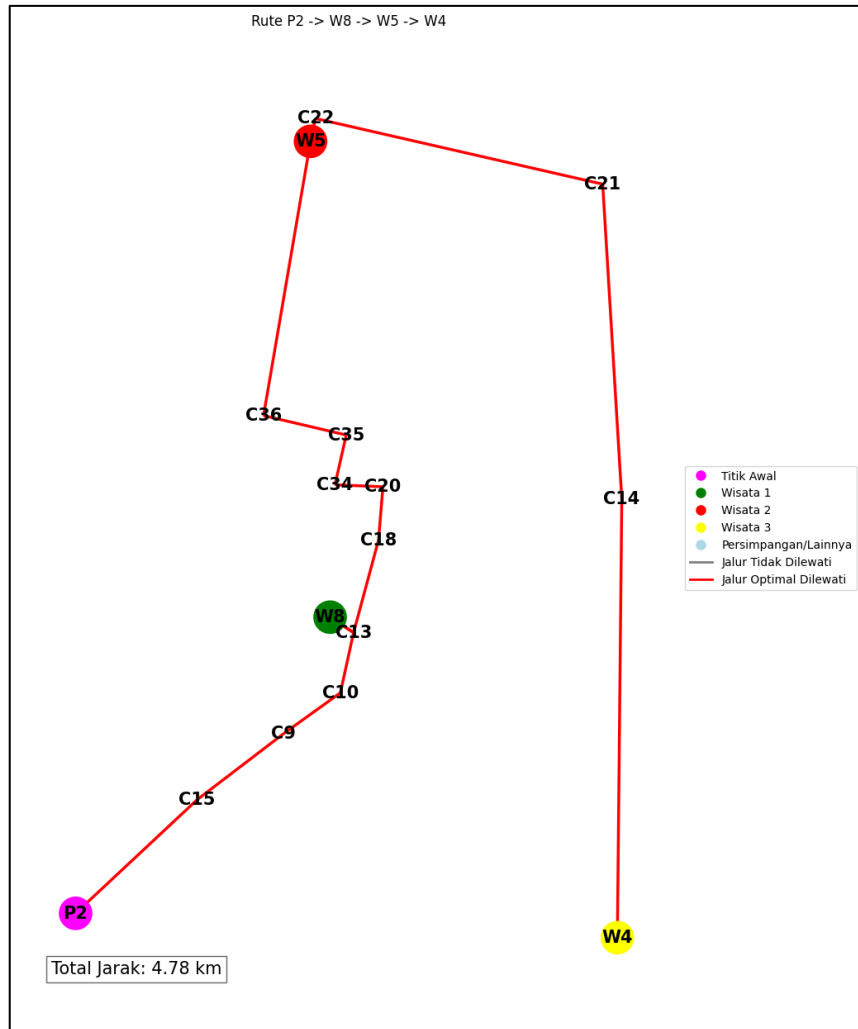
Terdapat perbedaan hasil rute antara algoritma *A-Star*, *Dijkstra*, dan *Greedy* terlihat dari prioritas lokasi wisata yang dikunjungi dan urutan persimpangan yang dilalui. Penggunaan warna merah bertujuan untuk menandai jalur yang optimal, sedangkan penggunaan warna hitam bertujuan untuk menandai jalur yang lebih panjang atau kurang optimal. Algoritma *A-Star* dan *Dijkstra* menunjukkan kesamaan dalam prioritas rute, di mana setelah mengunjungi W8, kedua algoritma ini melanjutkan ke W5 sebelum menuju W4. Hal ini menunjukkan bahwa keduanya berusaha mencari jalur terpendek dengan eksplorasi yang optimal. Karena pendekatan ini, *A-Star* dan *Dijkstra* menghasilkan urutan persimpangan yang lebih teratur dan konsisten, sehingga total jarak yang ditempuh hampir sama, seperti terlihat pada percobaan pertama (*A-Star* = 4.78 km, *Dijkstra* = 4.79 km).

Sebaliknya, algoritma *Greedy* menunjukkan perbedaan yang mencolok dalam menentukan prioritas rute. Setelah mengunjungi W8, *Greedy* justru memilih untuk melanjutkan ke W4 sebelum ke W5. Ini terjadi karena algoritma *Greedy*

hanya mempertimbangkan keuntungan lokal di setiap langkah, tanpa memperhitungkan keseluruhan rute yang optimal. Akibatnya, jalur yang dipilih sering kali tidak efisien, seperti yang terlihat dari jarak yang lebih panjang, misalnya 5.14 km pada percobaan pertama, lebih besar dibandingkan *A-Star* dan *Dijkstra*.

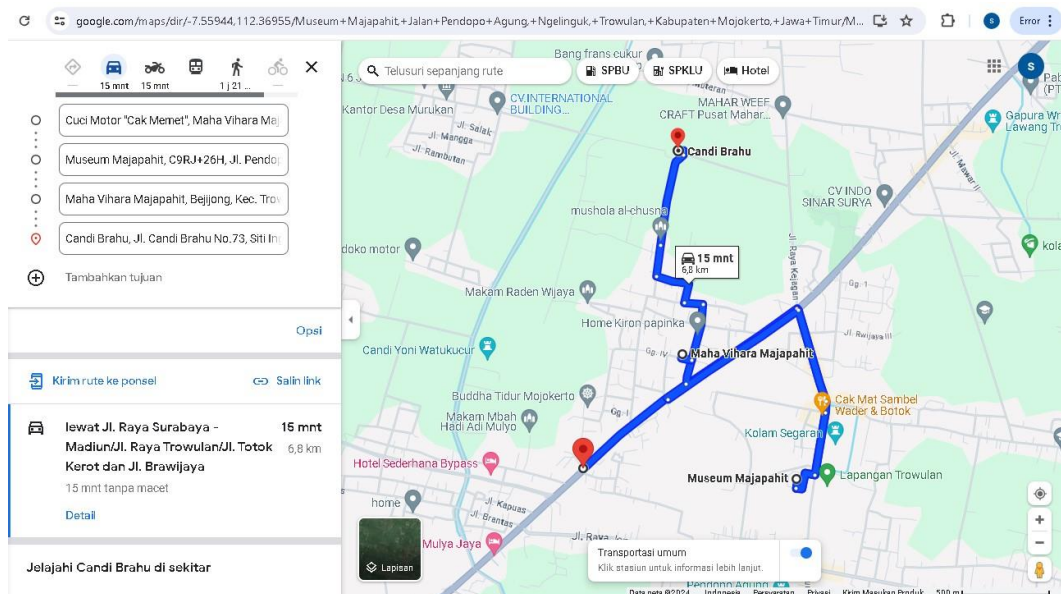
Secara keseluruhan, *A-Star* dan *Dijkstra* memberikan hasil rute yang lebih optimal dan konsisten, dengan urutan persimpangan yang teratur. Sementara itu, algoritma *Greedy* cenderung menghasilkan rute yang kurang efisien karena mengutamakan keuntungan lokal tanpa mempertimbangkan solusi global, sehingga total jarak yang ditempuh lebih panjang.

Sebagai contoh, perhitungan yang telah dijelaskan menggunakan algoritma *A-Star* adalah dari P2 ke W8, W5, dan W4. Hasilnya disajikan pada gambar berikut melalui Google Maps dengan jarak 4.78 km.



Gambar 4.21 Hasil Perhitungan Rute Penelitian di Google Maps

Kemudian, pembuktian akan dilakukan dengan melacak rute pada Google Maps dari P2 menuju W4, W8, dan W5. Penelitian ini menggunakan satuan kilometer (KM) untuk mengukur jarak. Salah satu hasil yang ditampilkan oleh Google Maps berdasarkan sampel di atas ditunjukkan pada gambar berikut.



Gambar 4.22 Hasil Perhitungan Rute Pembanding di Google Maps

Gambar di atas menunjukkan bahwa rute yang dihasilkan mengikuti urutan tujuan yang telah dimasukkan tanpa mempertimbangkan destinasi terdekat dari titik awal untuk dikunjungi terlebih dahulu. Jarak yang ditempuh adalah 6,8 km, yang lebih panjang dibandingkan dengan jarak yang dihasilkan oleh algoritma *A-Star*. Pengujian akan dilakukan sebanyak 10 kali secara acak, tetapi tetap menyeluruh, yang berarti semua destinasi wisata akan diuji sesuai dengan program yang telah dirancang sebelumnya. Meskipun pengujian dilakukan secara acak, setiap destinasi wisata akan mendapat kesempatan untuk diuji. Hasil dari pengujian ini akan ditampilkan dalam tabel berikut.

Tabel 4.5 Tabel Hasil Proses Pengujian Rute Terpendek

No.	Titik Awal	Tujuan	Rute Hasil Penelitian	Jarak pada Penelitian (KM)	Rute pada Maps	Jarak pada Maps (KM)
1.	P1	W1, W2, W3	W1, W3, W2	9.79	W3, W2, W1	16.0
2.	P7	W4, W14, W15	W15, W14, W4	3.78	W14, W15, W4	4.8
3.	P5	W11, W12, W13	W13, W12, W11	20.28	W13, W11, W12	31.3
4.	P2	W4, W5, W8	W8, W5, W4	4.78	W4, W8, W5	6.8
5.	P6	W6, W7, W5	W5, W6, W7	5.13	W7, W6, W5	6.3

No.	Titik Awal	Tujuan	Rute Hasil Penelitian	Jarak pada Penelitian (KM)	Rute pada Maps	Jarak pada Maps (KM)
6.	P3	W7, W5, W6	W5, W6, W7	9.58	W5, W7, W6	13.7
7.	P4	W21, W22, W3	W3, W21, W22	12.04	W3, W21, W22	14.3
8.	P9	W17, W18, W26	W17, W26, W18	0.93	W18, W26, W17	2.3
9.	P8	W4, W6, W8	W4, W8, W6	4.47	W8, W4, W6	7
10.	P10	W24, W19, W18	W19, W18, W24	8.29	W24, W18, W19	12.9

4.4 Evaluasi Pengujian

Setelah melakukan pengujian berulang kali, hasil menunjukkan bahwa rute yang dihitung menggunakan algoritma *A-Star* memiliki jarak yang relatif lebih pendek dibandingkan dengan rute yang dihasilkan oleh Google Maps. Setiap kali pengujian dilakukan, algoritma *A-Star* consistently menghasilkan rute yang lebih efisien dalam hal jarak tempuh. Hal ini menegaskan keunggulan algoritma *A-Star* dalam menemukan rute terpendek dibandingkan dengan metode penelusuran jalur di Google Maps.

Pengujian dilakukan dengan berbagai kombinasi tujuan wisata untuk memastikan keandalan hasil. Data yang diperoleh menunjukkan konsistensi bahwa algoritma *A-Star* selalu menemukan rute yang lebih singkat. Perhitungan efektivitas jarak tempuh dari hasil pengujian ini dilakukan dengan membandingkan jarak yang dihasilkan oleh algoritma *A-Star* dan Google Maps, dan kemudian menghitung persentase penghematan jarak. Hasil dari proses pengujian dan perhitungan ini akan ditampilkan dalam tabel berikut, yang menunjukkan efektivitas algoritma *A-Star* dalam mengurangi jarak tempuh dibandingkan dengan Google Maps:

Tabel 4.6 Tabel Hasil Nilai Efektifitas

Pengujian Ke-	Presentase (%)
1	38.81%
2	21.25%
3	35.21%
4	29.71%
5	18.57%
6	30.07%
7	15.80%
8	59.57%
9	36.14%
10	35.74%
Rata-rata	32.1%

Tabel di atas menunjukkan bahwa pengujian yang dilakukan menggunakan algoritma *A-Star* memiliki efisiensi jarak tempuh dengan persentase rata-rata sebesar 32.1%. Jarak tempuh yang lebih rendah yang dihasilkan oleh algoritma *A-Star* menjadikannya lebih efektif. Terdapat beberapa alasan yang menjelaskan mengapa hasil pengujian rute terpendek menuju beberapa destinasi wisata jauh lebih efisien. Berikut adalah penjelasannya:

1. Perhitungan nilai heuristik, keunikan yang dimiliki oleh algoritma *A-Star* terletak pada perhitungan nilai heuristiknya. Nilai heuristik ini adalah jarak antara setiap titik yang dihubungkan dengan titik tujuan, dihitung melalui perhitungan formula *haversine* yang diperoleh dari koordinat masing-masing titik yang tersebar pada gambar. Keefektifan nilai heuristik ini dapat terlihat pada pengujian ke-7. Pengujian ini dilakukan dengan titik awal P4 dan destinasi tujuan W21 (Alas Veenuz Trawas), W3 (Komunitas Pertanian Organik Brenjonk), dan W22 (Air Terjun Dlundung).

Hasil yang ditunjukkan oleh algoritma *A-Star* berbeda dengan Google Maps dalam hal urutan destinasi yang dikunjungi. Algoritma *A-Star* menunjukkan urutan pertama menuju W3 (Komunitas Pertanian Organik Brenjonk),

kemudian W21 (Alas Veenuz Trawas), dan terakhir W22 (Air Terjun Dlundung). Selain itu, terdapat perbedaan pada jarak yang ditempuh. Algoritma *A-Star* menghasilkan jarak rute sebesar 12.04 km, yang lebih pendek dibandingkan dengan jarak yang dihasilkan oleh Google Maps, yaitu 14.3 km, dengan prioritas jalur yang berbeda.

Meskipun prioritas destinasi wisata yang dituju sama, rute yang diambil oleh kedua metode berbeda. Hal ini merupakan hasil dari perhitungan nilai heuristik dalam penentuan rute yang lebih efisien. Penggunaan nilai heuristik yang efektif oleh algoritma *A-Star* memungkinkan penghitungan rute yang lebih pendek dan efisien dibandingkan dengan Google Maps. Efektivitas ini terlihat jelas pada perbedaan jarak yang dihasilkan, yang menunjukkan bahwa algoritma *A-Star* lebih optimal dalam menentukan rute terpendek.

2. Penyeleksian prioritas rute destinasi, proses penelitian ini tidak hanya memusatkan perhatian pada satu tujuan destinasi, melainkan dilakukan pada beberapa titik tujuan destinasi wisata sekaligus. Ketika ada tiga tujuan yang dipilih, masing-masing dari ketiga tujuan tersebut akan disaring menggunakan algoritma *A-Star*. Selanjutnya, destinasi yang akan menjadi tujuan pertama akan dipilih. Proses ini akan diulang untuk menentukan tujuan destinasi kedua, dan seterusnya, hingga mencapai tujuan destinasi terakhir. Pendekatan ini memungkinkan pengujian rute menjadi lebih efisien daripada pengujian rute yang dilakukan melalui Google Maps.

Perbedaan utama antara pendekatan dengan algoritma *A-Star* dan penggunaan Google Maps terletak pada proses penyeleksian destinasi wisata. Di Google Maps, tidak ada penyesuaian terhadap urutan destinasi wisata yang dipilih. Misalnya, jika seseorang memilih W4, W5, dan W8, Google Maps akan menampilkan destinasi pertama sebagai W4, kemudian W5, dan terakhir W8. Namun, pendekatan dengan algoritma *A-Star* berbeda. Meskipun destinasi yang dipilih adalah W4, W5, dan W8, belum tentu W4 akan menjadi tujuan destinasi pertama. Destinasi yang pertama akan dikunjungi adalah destinasi yang memiliki rute terpendek dari titik asal. Oleh karena itu, desain algoritma ini juga memiliki peran penting dalam menentukan pemilihan rute yang paling optimal dan efisien. Dengan demikian, pengujian dengan algoritma *A-Star* dapat memberikan hasil yang lebih efektif dan sesuai dengan kebutuhan perjalanan.

4.5 Implementasi Algoritma *A-Star* dalam Pandangan Islam

Implementasi Algoritma *A-Star* dalam pandangan Islam dapat dilihat sebagai upaya untuk mencapai tujuan dengan cara yang paling efisien dan efektif, sesuai dengan prinsip-prinsip Islam yang menekankan pentingnya kebijaksanaan, perencanaan yang baik, dan pengambilan keputusan yang tepat. Algoritma *A-Star* adalah salah satu algoritma pencarian yang paling populer dan efisien, digunakan untuk menemukan rute terpendek antara dua titik. Algoritma ini menggunakan heuristik untuk memprioritaskan jalur yang paling menjanjikan menuju tujuan, serupa dengan bagaimana seorang Muslim diarahkan untuk memilih jalan yang

paling tepat dan baik dalam hidup mereka, berdasarkan petunjuk dari Al-Quran dan Hadis. Dalam Surah Al-Baqarah ayat 286, Allah berfirman (Kemenag, 2024):

لَا يُكَلِّفُ اللَّهُ نَفْسًا إِلَّا وُسْعَهَا

Artinya:

"Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya."

Ayat ini menekankan bahwa setiap individu harus bertindak sesuai dengan kemampuan mereka dan tidak melebihi batas kemampuan tersebut. Dalam konteks Algoritma *A-Star*, konsep ini diterjemahkan menjadi penggunaan sumber daya yang efisien dan optimal. Algoritma ini memastikan bahwa setiap langkah yang diambil adalah yang paling efisien, tanpa membebani sistem dengan tugas yang melebihi kemampuannya, seperti waktu komputasi yang berlebihan atau penggunaan memori yang tidak perlu. Hal ini sejalan dengan ajaran Islam untuk berbuat sesuai dengan kemampuan dan tidak berlebihan dalam segala hal.

Selain itu, hadis dari Sahih Muslim menyebutkan bahwa Rasulullah SAW bersabda (Rohim & dkk, 2020):

إِنَّ اللَّهَ عَزَّ وَجَلَّ يُحِبُّ إِذَا عَمِلَ أَحَدُكُمْ عَمَلًا أَنْ يُتَّقِنَهُ

Artinya:

"Sesungguhnya Allah menyukai jika seseorang di antara kalian melakukan pekerjaan, maka dia melakukannya dengan itqan (dengan tekun dan sempurna)."

Konsep itqan, atau bekerja dengan tekun dan sempurna, juga sangat relevan dengan Algoritma *A-Star*. Algoritma ini dirancang untuk menemukan solusi yang tidak hanya benar tetapi juga yang paling optimal, dengan memprioritaskan langkah-langkah yang membawa kita lebih dekat ke tujuan dengan cara yang paling

efisien. Dengan kata lain, *A-Star* tidak hanya memastikan pencarian yang benar tetapi juga pencarian yang dilakukan dengan cara yang terbaik dan paling hati-hati.

Dalam Islam, efisiensi, ketekunan, dan perencanaan yang baik adalah sifat-sifat yang sangat dihargai. Prinsip-prinsip ini dapat dilihat dalam cara kerja Algoritma *A-Star*, yang selalu berusaha untuk mencapai tujuan akhir dengan menggunakan jalur yang paling optimal. Kata perencanaan berasal dari kata rencana, yang menurut Kamus Besar Bahasa Indonesia (Pusat Bahasa & Tim Penyusun Kamus, 2002) diartikan sebagai proses penyusunan sesuatu yang akan dilakukan. Sementara itu, kata 'jalan' dalam kamus yang sama dijelaskan sebagai cara, metode, syarat, atau usaha untuk mencapai sesuatu. Ini menunjukkan bahwa terdapat keterkaitan antara rencana dan jalan, terutama dari segi tujuan yang ingin dicapai, karena keduanya merujuk pada proses atau metode untuk mewujudkan tujuan di masa depan (Tajang & D, 2020).

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa penggunaan algoritma *A-Star* untuk menentukan rute terpendek terbukti lebih efektif dalam aspek jarak. Hal ini dibuktikan melalui pengujian rute terpendek yang dilakukan secara acak terhadap seluruh destinasi wisata sebanyak. Dari pengujian tersebut, ditemukan bahwa rata-rata jarak yang dihasilkan oleh algoritma *A-Star* lebih pendek sebesar 32.1%. Selain itu, perbandingan dengan algoritma *Greedy* dan *Dijkstra* menunjukkan bahwa algoritma *A-Star* memiliki kinerja yang lebih unggul dalam menemukan rute terpendek.

Keefektifan ini dapat dijelaskan melalui dua faktor utama. Pertama, adanya nilai heuristik yang diterapkan pada setiap titik terhadap titik tujuan memungkinkan algoritma *A-Star* untuk melakukan pemilihan rute dengan lebih tepat. Dengan demikian, jarak yang ditempuh menjadi lebih singkat. Faktor keefektifan kedua adalah desain penelitian yang melibatkan pembuatan prioritas destinasi. Dalam proses ini, beberapa destinasi tujuan dipilih dan diurutkan berdasarkan jarak terdekat dari titik awal. Algoritma *A-Star* digunakan untuk menentukan urutan kunjungan yang optimal. Inovasi dalam desain penelitian ini berkontribusi signifikan terhadap efektivitas pemilihan rute. Dengan demikian, penelitian ini menunjukkan bahwa pendekatan yang sistematis dan berbasis algoritma dapat meningkatkan efisiensi dalam perencanaan rute wisata.

5.2 Saran Untuk Penelitian Lanjutan

Penelitian ini telah menghasilkan modifikasi baru terhadap penentuan beberapa rute destinasi menggunakan algoritma *A-Star*. Dengan hasil yang diperoleh, diharapkan penelitian ini dapat menjadi pondasi yang kuat untuk penelitian selanjutnya, terutama dalam konteks menentukan rute destinasi di luar wilayah Kabupaten Mojokerto. Selain itu, ada harapan agar penelitian ini dapat terus dikembangkan, khususnya dalam aspek visualisasi, sehingga tampilan peta dapat lebih rapi dan sesuai dengan format yang digunakan pada Google Maps. Pengembangan ini tidak hanya akan meningkatkan estetika, tetapi juga memudahkan pengguna dalam memahami rute yang dihasilkan.

Selain itu, pembuatan program untuk mengotomatisasi proses perhitungan juga sangat penting untuk penelitian di masa mendatang. Mengingat perhitungan manual yang cukup panjang dan kompleks, pembuatan program yang mampu menghitung algoritma *A-Star* secara otomatis akan sangat membantu dalam implementasi algoritma ini dalam kehidupan sehari-hari. Dengan adanya program tersebut, pengguna dapat dengan mudah mendapatkan rute terpendek tanpa harus melalui proses manual yang memakan waktu, sehingga dapat meningkatkan efisiensi dan kenyamanan dalam merencanakan perjalanan.

Selanjutnya, penelitian ini juga dapat menjajaki integrasi dengan data real-time, seperti kondisi lalu lintas dan informasi cuaca, yang akan semakin memperkaya akurasi rute yang dihasilkan. Dengan demikian, penelitian ini tidak hanya memberikan kontribusi dalam pengembangan teori, tetapi juga dalam praktik nyata yang dapat diaplikasikan dalam dunia pariwisata dan navigasi sehari-hari.

DAFTAR PUSTAKA

- Ali, M., Izzuddin, A., & dkk. (2013). *Bimbingan Musafir*. Jakarta: Kementerian Agama R.I.
- Arsyad, M. A., Supriyadi, D., & dll. (2019). Penerapan Algoritma A Star Untuk Pencarian Rute Terpendek Pukesmas Rawat Inap Di Banyumas. *CENTIVE*, 74-82.
- Avini, T., Alamin, Z., & dkk. (2024). *Fundamental Algorithma*. Serang-Banten: Sada Kurnia Pustaka.
- Betz, J. W. (2015). *Engineering Satellite-Based Navigation and Timing: Global Navigation Satellite Systems, Signals, and Receivers*. Hoboken: IEEE PRESS WILEY.
- Bunaen, M. C., Pratiwi, H., & Riti, Y. F. (2022). Penerapan Algoritma *Dijkstra* untuk Menentukan Rute Terpendek Dari Pusat Kota Surabaya Ke Tempat Bersejarah. *Jurnal Teknologi Dan Sistem Informasi Bisnis* , 213-223.
- Dalem, I. A. (2018). Penerapan Algoritma A* (Star) Menggunakan Graph Untuk Menghitung Jarak Terpendek. *Jurnal RESISTOR*, 41-47.
- Dinas Komunikasi dan Informatika. (2024, Maret 18). *Gambaran Umum Wilayah Kabupaten Mojokerto*. Retrieved from Pemerintah Kabupaten Mojokerto: <https://mojokertokab.go.id/gambaran-umum>
- Fauzi, A., Supriyadi, & Yanitasari, Y. (2018). Implementasi Algoritma *A-Star* untuk Pemetaan Koordinat Tumbuhan Langka Berbasis Web. *Informatika Pertanian*, 87-100.
- Fernando, Y., Mutsaqof, M. A., & Megawaty, D. A. (2020). Penerapan Algoritma *A-Star* Pada Aplikasi Pencarian Lokasi Fotografi Di Bandar Lampung Berbasis Android. *Jurnal TEKNOINFO*, 27-34.
- Hartatik, Rahim, A. A., & dkk. (2023). Implementasi Algoritma *Greedy* untuk Mencari Jalur Terpendek Pada Sebuah Lokasi Wisata Kota Ternate. *Jurnal Informatika Komputer Bisnis dan Manajemen*, 13-22.
- Hutabalian, M., Sunanto, & Amien, J. A. (2021). Sistem Informasi Geografis Pemetaan Tempat Pembungan Sampah Sementara di Kota Pekanbaru Dengan Mencari Rute Terdekat Menggunakan Algoritma A Star (A*). *Jurnal Computer Science and Information Technology (CoSciTech)*, 98-107.
- Indriyono, B. V., & Widyatmoko. (2021). Optimization of Breadth-First Search Algorithm for Path Solutions in Mazyin Games. *International Journal of Artificial Intelligence & Robotics (IJAIR)*, 58-66.
- Juhari. (2021). *Matematika Diskrits*. Jawa Tengah: CV. Pena Persada.
- Katsir, I. (2015, Oktober 25). *Tafsir Surat Al-Mulk, ayat 12-15*. Retrieved from Tafsir Ibnu Katsir: <http://www.ibnukatsironline.com/2015/10/tafsir-surat-al-mulk-ayat-12-15.html>
- Kemenag. (2024, Maret 12). *Qur'an Kemenag*. Retrieved from Lajnah Pentashihan mushaf Al-Qur'an: <https://quran.kemenag.go.id/>
- Kementerian Agama RI. (2011). Al-Qur'an Dan Tafsirnya (Edisi Yang Disempurnakan). In *Al-Qur'an Dan Tafsirnya (Edisi Yang Disempurnakan)*. Jakarta: Widya Cahaya.
- Lubis, M. A., Triase, & Alda, M. (2024). Implementasi Algoritma A* (*A-Star*) Pada Sistem Informasi Geografis Berbasis Web untuk Menentukan Jalur

- Terpendek Halte Bus Transmetro Deli. *Journal of Science and Sosial Research*, 758-764.
- Marcelina, D., & Yulianti, E. (2020). Aplikasi Pencarian Rute Terpendek Lokasi Kuliner Khas Palembang Menggunakan Algoritma Euclidean Distance dan A*(Star). *Jurnal SISFOKOM*, 195-202.
- Martins, O. O., & dkk. (2022). An Improved multi-objective *A-Star* algorithm for path planning in a large workspace: Design, Implementation, and Evaluation. *Scientific African*, 1-11.
- Miftahuddin, Y., Umaroh, S., & Karim, F. R. (2020). Perbandingan Metode Perhitungan Jarak Euclidean, Haversine, dan Manhattan dalam Penentuan Posisi Karyawan (Studi Kasus: Institut Teknologi Nasional Bandung). *Jurnal Tekno Insentif*, 69-77.
- Mukti, M. R., & Mulyono. (2018). Menentukan Rute Terpendek Dengan Menggunakan Algoritma Floyd-Warshall Dalam Pendistribusian Barang Pada PT. Rapy Ray Putratama. *KARISMATIKA*, 39-53.
- Mulya, M. F., Trisanto, D., & dkk. (2022). Analisis Sistem Pencarian Jalur Terdekat Bengkel UMKM Kendaraan Roda Dua Dengan Algoritma A* (*A-Star*) dan Formula Haversine. *Jurnal Sistem Komputer dan Kecerdasan Buatan*, 61-68.
- Munir, R. (2016). *Matematika Diskrit*. Bandung: Informatika Bandung.
- Pusat Bahasa, & Tim Penyusun Kamus. (2002). *Kamus Besar Bahasa Indonesia*. Jakarta: Balai Pustaka.
- Rahayuningsih, S. (2018). *Teori Graph dan Penerapannya*. Malang: Universitas Wisnuwardhana Press Malang (Unidha Press).
- Rohim, A. N., & dkk. (2020). *Kumpulan Kultum Ekonomi Syariah Seri 3*. Jakarta: Komite Nasional Ekonomi dan Keuangan Syariah.
- Russel, S. J., & Norvig, P. (2010). *Artificial Intelligence A Modern Approach Third Edition*. London: Pearson.
- Sayono, J. (2021). Langkah-Langkah Heuristik dalam Metode Sejarah di Era Digital. *Jurnal Sejarah dan Budaya*, 369-376.
- Setyawan, A. R., Irawan, J. D., & Faisol, A. (2023). Rancang Bangun Aplikasi Penentuan Rute Terpendek Distribusi Jamu Tradisional Menggunakan Metode A*. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 2218-2225.
- Shishkov, B. (2024). *Business Modeling and Software Design*. Cham: Springer.
- Sinlae, A. J., Nuraini, R., & dkk. (2023). Implementasi Algoritma A* (*A-Star*) dan Greedy Dalam Penentuan Routing Pada Wide Area Network (WAN). *Journal of Computer System and Informatics (JoSYC)*, 551-557.
- Suwandi, A. K., Haryati, E., & Suroso. (2023). Evaluasi Kebijakan Pengembangan Destinasi Wisata Peninggalan Kerajaan Majapahit di Kabupaten Mojokerto Untuk Meningkatkan Kunjungan Wisatawan. *Soetomo Administrasi Publik*, 615-624.
- Syihabuddin, R. F., Jauhari, M. N., & dkk. (2022). Implementasi Algoritma *A-Star* dalam Menentukan Rute Terpendek Destinasi Wisata Kota Malang. *Jurnal Riset Mahasiswa Matematika*, 236-245.
- Tafsir, L. Q. (2024). *Tafsir Surat Al-Kahfi ayat 19*. Retrieved Juni 24, 2024, from Learn Quran Tafsir: <https://tafsir.learn-quran.co/id/surat-18-al-kahfi/ayat-19>

- Tajang, A. D., & D, A. Z. (2020). Konsep Perencanaan dalam Islam: Suatu Pengantar. *Study of Scientific and Behavioral Management (SSBM)*, 103-115.
- Talan, K., & Bamnote, G. R. (2015). Shortest Path Finding Using a Star Algorithm and Minimum weight Node First Principle. *International Journal of Innovation Research in Computer and Communication Engineering*, 4290-4294.
- Taufik, M., & Wandini, A. I. (2012). Pengembangan WEBSIG Obyek Wisata dan Budaya di Kabupaten Mojokerto. *GEOID*, 14-22.
- Wayahdi, M. R., Ginting, S. N., & dkk. (2021). Greedy, A-Star, and Dijkstra's Algorithms in Finding Shortest Path. *International Journal of Advances in Data and Information Systems*, 45-52.

LAMPIRAN

Lampiran 1. Data Jarak Antar Titik

No.	Titik 1	Titik 2	Jarak (KM)	No.	Titik 1	Titik 2	Jarak (KM)	No.	Titik 1	Titik 2	Jarak (KM)
1.	C15	C9	0.24	81.	C60	C62	0.16	161.	C22	C23	0.54
2.	C15	C16	0.44	82.	C62	C63	0.76	162.	C23	C24	0.2
3.	C9	C10	0,16	83.	C62	C64	0.21	163.	C24	W6	0.14
4.	C9	C12	0.26	84.	C64	C65	0.76	164.	W4	C44	0.12
5.	C10	C13	0.14	85.	C64	C66	0.88	165.	C44	C43	0.43
6.	C10	C11	0.17	86.	C67	C66	0.19	166.	C44	C40	0.53
7.	C13	W8	0.06	87.	C67	C68	0.23	167.	C43	P8	0.56
8.	C11	C14	0.6	88.	C71	C72	0.84	168.	P8	C21	0.7
9.	C19	C17	0.08	89.	C71	C73	0.55	169.	W8	C13	0.06
10.	C20	C34	0.11	90.	C73	C74	0.2	170.	C13	C18	0.21
11.	C19	C18	0.34	91.	C73	C75	0.37	171.	C13	C31	0.46
12.	C17	C36	0.12	92.	C78	C79	0.35	172.	C18	C20	0.12
13.	C17	C34	0.18	93.	C78	C80	0.34	173.	C18	C19	0.34
14.	C18	C32	0.25	94.	C56	C57	0.41	174.	C19	C17	0.08
15.	C18	C19	0.34	95.	C56	C54	0.36	175.	C36	C17	0.12
16.	C13	C18	0.21	96.	C53	C52	0.7	176.	C34	C35	0.11
17.	C18	C20	0.12	97.	C48	C47	0.07	177.	C34	C17	0.18
18.	C34	C17	0.18	98.	C48	C50	0.26	178.	C35	C36	0.19
19.	C34	C35	0.11	99.	C47	C46	0.23	179.	C36	C23	0.13
20.	C13	C10	0.14	100.	P5	C82	2.95	180.	P10	C86	0.06
21.	C13	C11	0.12	101.	C51	C52	1.14	181.	P10	C87	0.06
22.	W5	C22	0.05	102.	C58	C59	0.64	182.	C86	C88	0.56
23.	C35	C36	0.19	103.	C59	C60	0.27	183.	C88	C89	0.1
24.	W5	C36	0.62	104.	C65	C67	0.1	184.	C89	C90	0.1
25.	P2	C15	0.37	105.	C66	C69	0.19	185.	C90	W19	0.07
26.	C13	W8	0.06	106.	C69	C70	0.63	186.	C88	W18	1.48
27.	C11	C14	0.6	107.	C70	C71	1.35	187.	P10	W24	2.5
28.	C14	W4	0.98	108.	C74	C76	0.24	188.	C90	C91	0.05
29.	C12	C16	0.18	109.	C76	C77	0.33	189.	C91	C87	0.59
30.	C16	C19	0.25	110.	C77	C78	1.74	190.	C87	C86	0.09
31.	C36	W5	0.62	111.	C80	W11	0.99	191.	C87	W24	2.49
32.	W8	C13	0.06	112.	C56	W12	1.89	192.	W18	C88	1.48
33.	C35	C36	0.19	113.	C48	C49	0.2	193.	C88	P10	0.53
34.	C22	C21	0.66	114.	C49	W13	0.11	194.	P4	C6	2.39
35.	C21	C14	0.7	115.	C83	C84	0.43	195.	C6	C5	0.44
36.	C14	W4	0.98	116.	C83	C48	0.26	196.	C6	C2	0.62
37.	C1	W1	0.38	117.	C55	C57	0.5	197.	C5	C8	0.89
38.	C1	C2	0.97	118.	C21	C22	0.66	198.	C5	C4	1.01
39.	C1	C8	1.36	119.	C21	C14	0.7	199.	C8	C3	0.83
40.	C6	C5	0.5	120.	C25	C26	0.17	200.	C3	W21	1.25
41.	C6	C7	1.96	121.	C25	C28	0.59	201.	P4	C2	2.83
42.	C1	C3	2.14	122.	C21	P6	0.42	202.	C2	W22	2.09
43.	C4	C5	1.01	123.	C23	C36	0.13	203.	C4	W3	0.46
44.	C4	C3	1.45	124.	C23	C22	0.54	204.	C4	C3	1.45

No.	Titik 1	Titik 2	Jarak (KM)	No.	Titik 1	Titik 2	Jarak (KM)	No.	Titik 1	Titik 2	Jarak (KM)
45.	C5	C6	0.5	125.	C36	C17	0.12	205.	C8	C2	0.43
46.	C5	C7	2.18	126.	C36	C35	0.19	206.	P3	C93	1.46
47.	P1	C1	1.34	127.	C17	C19	0.1	207.	C93	C92	1.93
48.	C2	C6	0.62	128.	C17	C34	0.18	208.	C93	C27	4.09
49.	C4	W3	0.19	129.	C18	C20	0.12	209.	C27	C14	1.39
50.	W1	C1	0.38	130.	C18	C32	0.25	210.	C92	C28	2.3
51.	C7	W2	0.48	131.	C23	C24	0.2	211.	P6	C25	0.8
52.	C4	W3	0.19	132.	C24	W6	0.14	212.	C25	C26	0.17
53.	W3	C4	0.19	133.	C22	W5	0.05	213.	C26	C27	0.31
54.	P7	C37	0.48	134.	P6	C25	0.8	214.	C27	W7	0.2
55.	P7	W15	0.35	135.	C26	C27	0.27	215.	P6	C21	0.42
56.	C38	C39	0.62	136.	C27	W7	0.2	216.	C21	C22	0.66
57.	C38	C41	0.92	137.	W5	C23	0.49	217.	C22	W5	0.05
58.	C39	C40	0.28	138.	C24	W6	0.14	218.	C22	C23	0.55
59.	C39	C42	0.5	139.	C19	C18	0.34	219.	C23	C24	0.2
60.	W15	C45	0.46	140.	C20	C33	0.26	220.	C24	W6	0.14
61.	C45	C37	0.22	141.	C33	C32	0.1	221.	W5	C22	0.05
62.	C37	W14	0.14	142.	C32	C31	0.09	222.	C21	C14	0.7
63.	C37	C38	1.13	143.	C31	C27	1.64	223.	C25	C28	0.59
64.	C40	C44	0.53	144.	P9	W26	0.3	224.	W5	C23	0.49
65.	C44	W4	0.12	145.	W26	W17	0.13	225.	W6	C24	0.12
66.	W14	C38	1.06	146.	W17	W18	0.48	226.	C23	C36	0.13
67.	C82	C46	0.22	147.	P9	W17	0.2	227.	C36	C17	0.12
68.	C82	C83	0.26	148.	P9	W18	0.42	228.	C36	C35	0.19
69.	C46	C47	0.23	149.	W26	W18	0.6	229.	C17	C19	0.1
70.	C51	C52	1.14	150.	P8	C43	0.56	230.	C17	C34	0.18
71.	C46	C51	0.33	151.	C43	C44	0.43	231.	C19	C18	0.34
72.	C47	C48	0.07	152.	C43	C41	0.9	232.	C18	C20	0.12
73.	C47	C83	0.19	153.	C44	W4	0.12	233.	C18	C32	0.25
74.	C52	C53	0.7	154.	P8	C10	0.97	234.	C20	C33	0.26
75.	C52	C55	0.75	155.	C10	C13	0.14	235.	C33	C32	0.1
76.	C53	C54	0.16	156.	C10	C11	0.17	236.	C32	C31	0.09
77.	C53	C55	0.33	157.	C13	W8	0.06	237.	C31	C14	0.26
78.	C57	C58	0.67	158.	C10	C13	0.14	238.	C14	C27	1.43
79.	C57	C59	0.93	159.	P8	C10	0.77				
80.	C60	C61	0.76	160.	C21	C22	0.66				

Lampiran 2. Data Heuristik Menggunakan Formula Haversine

No.	Titik 1	Titik 2	Jarak (KM)	No.	Titik 1	Titik 2	Jarak (KM)	No.	Titik 1	Titik 2	Jarak (KM)
1.	P1	W1	1.14	155.	P4	W22	4.33	309.	C26	W7	0.43
2.	P1	W2	3.22	156.	P4	W3	3.74	310.	C25	W5	1.45
3.	P1	W3	4.07	157.	C6	W21	2.58	311.	C25	W6	1.71
4.	C1	W1	0.38	158.	C6	W22	2.68	312.	C25	W7	0.4
5.	C1	W2	2.79	159.	C6	W3	1.77	313.	C27	W5	1.65
6.	C1	W3	2.77	160.	C5	W21	2.14	314.	C27	W6	1.91
7.	C2	W1	1.29	161.	C5	W22	2.79	315.	C27	W7	0.2
8.	C2	W2	3.05	162.	C5	W3	1.3	316.	C28	W5	1.64
9.	C2	W3	1.97	163.	C2	W21	2.3	317.	C28	W6	1.97
10.	C3	W1	2.49	164.	C2	W22	2.09	318.	C28	W7	0.7
11.	C3	W2	4.09	165.	C2	W3	1.97	319.	C36	W5	0.62
12.	C3	W3	1.92	166.	C8	W21	1.92	320.	C36	W6	0.33
13.	C4	W1	2.67	167.	C8	W22	1.93	321.	C36	W7	2.11
14.	C4	W2	3.23	168.	C8	W3	1.88	322.	C17	W5	0.74
15.	C4	W3	0.47	169.	C4	W21	1.44	323.	C17	W6	0.44
16.	C5	W1	1.66	170.	C4	W22	3.29	324.	C17	W7	2.17
17.	C5	W2	2.65	171.	C4	W3	0.47	325.	C35	W5	0.66
18.	C5	W3	1.3	172.	C3	W21	1.25	326.	C35	W6	0.45
19.	C6	W1	1.17	173..	C3	W22	1.95	327.	C35	W7	1.96
20.	C6	W2	2.44	174.	C3	W3	1.92	328.	C19	W5	0.81
21.	C6	W3	1.77	175..	P5	W11	8.39	329.	C19	W6	0.49
22.	C7	W1	2.12	176.	P5	W12	5.18	330.	C19	W7	2.25
23.	C7	W2	0.48	177.	P5	W13	3.13	331.	C34	W5	0.77
24.	C7	W3	2.72	178.	C82	W11	6.91	332.	C34	W6	0.53
25.	C8	W1	1.7	179.	C82	W12	3.13	333.	C34	W7	2.03
26.	C8	W2	3.41	180.	C82	W13	0.49	334.	C18	W5	0.9
27.	C8	W3	1.88	181.	C46	W11	6.88	335.	C18	W6	0.69
28.	P2	W8	0.88	182.	C46	W12	3.08	336.	C18	W7	2.01
29.	P2	W4	1.22	183.	C46	W13	0.43	337.	C20	W5	0.79
30.	P2	W5	1.8	184.	C51	W11	6.85	338.	C20	W6	0.6
31.	C15	W8	0.51	185.	C51	W12	3.04	339.	C20	W7	1.94
32.	C15	W4	1	186.	C51	W13	0.53	340.	C32	W5	1.01
33.	C15	W5	1.49	187.	C52	W11	6.83	341.	C32	W6	0.87
34.	C9	W8	0.28	188.	C52	W12	3.11	342.	C32	W7	1.84
35.	C9	W4	0.88	189.	C52	W13	1.53	343.	C33	W5	0.93
36.	C9	W5	1.32	190.	C53	W11	6.14	344.	C33	W6	0.82
37.	C16	W8	0.26	191.	C53	W12	2.4	345.	C33	W7	1.76
38.	C16	W4	1.17	192.	C53	W13	1.69	346.	C31	W5	1.07
39.	C16	W5	1.06	193.	C54	W11	6.01	347.	C31	W6	0.95
40.	C10	W8	0.17	194.	C54	W12	2.25	348.	C31	W7	1.78
41.	C10	W4	0.83	195.	C54	W13	1.71	349.	P7	W15	0.35
42.	C10	W5	1.23	196.	C55	W11	6.1	350.	P7	W14	0.53
43.	C12	W8	0.08	197.	C55	W12	2.45	351.	P7	W4	2.7
44.	C12	W4	1.02	198.	C55	W13	1.99	352.	C37	W15	0.58
45.	C12	W5	1.07	199.	C56	W11	5.65	353.	C37	W14	0.14
46.	C13	W8	0.06	200.	C56	W12	1.89	354.	C37	W4	2.27
47.	C13	W4	0.9	201.	C56	W13	1.94	355.	C45	W15	0.46
48.	C13	W5	1.1	202.	C57	W11	5.6	356.	C45	W14	0.36

No.	Titik 1	Titik 2	Jarak (KM)	No.	Titik 1	Titik 2	Jarak (KM)	No.	Titik 1	Titik 2	Jarak (KM)
49.	C11	W8	0.18	203.	C57	W12	1.95	357.	C45	W4	2.41
50.	C11	W4	0.8	204.	C57	W13	2.25	358.	C38	W15	1.69
51.	C11	W5	1.16	205.	C58	W11	5.57	359.	C38	W14	1.06
52.	C14	W8	0.71	206.	C58	W12	2.21	360.	C38	W4	1.17
53.	C14	W4	0.98	207.	C58	W13	2.82	361.	C39	W15	2.3
54.	C14	W5	1.06	208.	C59	W11	4.93	362.	C39	W14	1.68
55.	C18	W8	0.2	209.	C59	W12	1.7	363.	C39	W4	0.66
56.	C18	W4	1.04	210.	C59	W13	3.18	364.	C40	W15	2.58
57.	C18	W5	0.9	211.	C60	W11	4.72	365.	C40	W14	1.96
58.	C19	W8	0.35	212.	C60	W12	1.66	366.	C40	W4	0.51
59.	C19	W4	1.31	213.	C60	W13	3.44	367.	C41	W15	2.14
60.	C19	W5	0.81	214.	C61	W11	4.82	368.	C41	W14	1.45
61.	C17	W8	0.37	215.	C61	W12	2.3	369.	C41	W4	1.09
62.	C17	W4	1.32	216.	C61	W13	4.05	370.	C42	W15	2.41
63.	C17	W5	0.74	217.	C62	W11	4.56	371.	C42	W14	1.73
64.	C36	W8	0.47	218.	C62	W12	1.57	372.	C42	W4	0.47
65.	C36	W4	1.41	219.	C62	W13	3.54	373.	C44	W15	2.76
66.	C36	W5	0.62	220.	C63	W11	4.67	374.	C44	W14	2.08
67.	C34	W8	0.3	221.	C63	W12	2.24	375.	C44	W4	0.12
68.	C34	W4	1.2	222.	C63	W13	4.13	376.	P8	W4	0.98
69.	C34	W5	0.76	223.	C64	W11	4.35	377.	P8	W8	0.71
70.	C32	W8	0.38	224.	C64	W12	1.46	378.	P8	W6	1.03
71.	C32	W4	0.9	225.	C64	W13	3.69	379.	C43	W4	0.48
72.	C32	W5	1.01	226.	C65	W11	4.46	380.	C43	W8	0.87
73.	C20	W8	0.31	227.	C65	W12	2.17	381.	C43	W6	1.49
74.	C20	W4	1.14	228.	C65	W13	4.26	382.	C44	W4	0.12
75.	C20	W5	0.79	229.	C66	W11	4.26	383.	C44	W8	1.03
76.	C35	W8	0.41	230.	C66	W12	2.19	384.	C44	W6	1.77
77.	C35	W4	1.28	231.	C66	W13	4.47	385.	C41	W4	1.09
78.	C35	W5	0.66	232.	C67	W11	4.45	386.	C41	W8	1.78
79.	C22	W8	1.11	233.	C67	W12	2.24	387.	C41	W6	2.32
80.	C22	W4	1.95	234.	C67	W13	4.35	388.	C10	W4	0.83
81.	C22	W5	0.05	235.	C68	W11	4.58	389.	C10	W8	0.17
82.	C21	W8	1.15	236.	C68	W12	2.45	390.	C10	W6	0.97
83.	C21	W4	1.68	237.	C68	W13	4.46	391.	C13	W4	0.9
84.	C21	W5	0.66	238.	C69	W11	4.37	392.	C13	W8	0.06
85.	P3	W7	5.49	239.	C69	W12	2.38	393.	C13	W6	0.85
86.	P3	W6	5.78	240.	C69	W13	4.58	394.	C11	W4	0.8
87.	P3	W5	5.47	241.	C70	W11	4.51	395.	C11	W8	0.18
88.	C93	W7	4.13	242.	C70	W12	2.98	396.	C11	W6	0.93
89.	C93	W6	4.75	243.	C70	W13	5.14	397.	C21	W4	1.68
90.	C93	W5	4.41	244.	C71	W11	3.33	398.	C21	W8	1.15
91.	C92	W7	2.38	245.	C71	W12	3.1	399.	C21	W6	0.88
92.	C92	W6	2.85	246.	C71	W13	6	400.	C22	W4	1.95
93.	C92	W5	2.5	247.	C72	W11	3.84	401.	C22	W8	1.11
94.	C27	W7	0.2	248.	C72	W12	3.91	402.	C22	W6	0.41
95.	C27	W6	1.91	249.	C72	W13	6.65	403.	C23	W4	1.51
96.	C27	W5	1.65	250.	C73	W11	2.98	404.	C23	W8	0.6
97.	P6	W7	1.2	251.	C73	W12	3.41	405.	C23	W6	0.21

No.	Titik 1	Titik 2	Jarak (KM)	No.	Titik 1	Titik 2	Jarak (KM)	No.	Titik 1	Titik 2	Jarak (KM)
98.	P6	W6	1.05	252.	C73	W13	6.46	406.	C24	W4	1.67
99.	P6	W5	0.72	253.	C74	W11	2.81	407.	C24	W8	0.72
100.	C28	W7	0.7	254.	C74	W12	3.45	408.	C24	W6	0.14
101.	C28	W6	1.97	255.	C74	W13	6.57	409.	C40	W4	0.51
102.	C28	W5	1.64	256.	C75	W11	2.8	410.	C40	W8	1.42
103.	C25	W7	0.4	257.	C75	W12	3.06	411.	C40	W6	2.21
104.	C25	W6	1.71	258.	C75	W13	6.17	412.	C18	W4	1.04
105.	C25	W5	1.45	259.	C76	W11	2.59	413.	C18	W8	0.2
106.	C26	W7	0.43	260.	C76	W12	3.51	414.	C18	W6	0.69
107.	C26	W6	1.69	261.	C76	W13	6.7	415.	C31	W4	0.85
108.	C26	W5	1.45	262.	C77	W11	2.77	416.	C31	W8	0.46
109.	C21	W7	1.22	263.	C77	W12	3.83	417.	C31	W6	0.95
110.	C21	W6	0.88	264.	C77	W13	6.98	418.	C20	W4	1.14
111.	C21	W5	0.67	265.	C78	W11	1.16	419.	C20	W8	0.31
112.	C22	W7	1.83	266.	C78	W12	3.98	420.	C20	W6	0.6
113.	C22	W6	0.41	267.	C78	W13	7.46	421.	C19	W4	1.31
114.	C22	W5	0.05	268.	C79	W11	1.28	422.	C19	W8	0.35
115.	C23	W7	2.05	269.	C79	W12	4.33	423.	C19	W6	0.49
116.	C23	W6	0.21	270.	C79	W13	7.8	424.	C34	W4	1.19
117.	C23	W5	0.49	271.	C80	W11	0.99	425.	C34	W8	0.3
118.	C24	W7	2.21	272.	C80	W12	4.2	426.	C34	W6	0.53
119.	C24	W6	0.14	273.	C80	W13	7.69	427.	C17	W4	1.32
120.	C24	W5	0.5	274.	C83	W11	7.15	428.	C17	W8	0.37
121.	C14	W7	1.52	275.	C83	W12	3.36	429.	C17	W6	0.44
122.	C14	W6	1.03	276.	C83	W13	0.25	430.	C35	W4	1.28
123.	C14	W5	1.06	277.	C47	W11	7.11	431.	C35	W8	0.41
124.	C36	W7	2.11	278.	C47	W12	3.3	432.	C35	W6	0.45
125.	C36	W6	0.33	279.	C47	W13	0.2	433.	C36	W4	1.41
126.	C36	W5	0.62	280.	C48	W11	7.09	434.	C36	W8	0.47
127.	C17	W7	2.17	281.	C48	W12	3.28	435.	C36	W6	0.33
128.	C17	W6	0.44	282.	C48	W13	0.23	436.	P9	W26	0.3
129.	C17	W5	0.74	283.	C49	W11	7.28	437.	P9	W17	0.2
130.	C35	W7	1.96	284.	C49	W12	3.47	438.	P9	W18	0.42
131.	C35	W6	0.45	285.	C49	W13	0.11	439.	P10	W19	0.6
132.	C35	W5	0.66	286.	C50	W11	7.02	440.	P10	W18	1.96
133.	C19	W7	2.25	287.	C50	W12	3.21	441.	P10	W24	2.5
134.	C19	W6	0.49	288.	C50	W13	0.43	442.	C86	W19	0.64
135.	C19	W5	0.81	289.	P6	W5	0.72	443.	C86	W18	2
136.	C34	W7	2.03	290.	P6	W6	1.05	444.	C86	W24	2.56
137.	C34	W6	0.53	291.	P6	W7	1.2	445.	C87	W19	0.64
138.	C34	W5	0.77	292.	C21	W5	0.67	446.	C87	W18	2
139.	C18	W7	2.01	293.	C21	W6	0.88	447.	C87	W24	2.49
140.	C18	W6	0.69	294.	C21	W7	1.22	448.	C88	W19	0.27
141.	C18	W5	0.9	295.	C22	W5	0.05	449.	C88	W18	1.48
142.	C20	W7	1.94	296.	C22	W6	0.41	450.	C88	W24	2.35
143.	C20	W6	0.6	297.	C22	W7	1.83	451.	C89	W19	0.16
144.	C20	W5	0.79	298.	C14	W5	1.06	452.	C89	W18	1.44
145.	C32	W7	1.84	299.	C14	W6	1.03	453.	C89	W24	2.25
146.	C32	W6	0.87	300.	C14	W7	1.52	454.	C90	W19	0.07

No.	Titik 1	Titik 2	Jarak (KM)	No.	Titik 1	Titik 2	Jarak (KM)	No.	Titik 1	Titik 2	Jarak (KM)
147.	C32	W5	1.01	301.	C23	W5	0.49	455.	C90	W18	1.37
148.	C33	W7	1.76	302.	C23	W6	0.21	456.	C90	W24	2.16
149.	C33	W6	0.82	303.	C23	W7	2.05	457.	C91	W19	0.07
150.	C33	W5	0.93	304.	C24	W5	0.5	458.	C91	W18	1.41
151.	C31	W7	1.78	305.	C24	W6	0.14	459.	C91	W24	2.16
152.	C31	W6	0.95	306.	C24	W7	2.21	460.	W15	W14	0.69
153.	C31	W5	1.07	307.	C26	W5	1.45	461.	W15	W4	2.85
154.	P4	W21	4.93	308.	C26	W6	1.69				

Lampiran 3. Data Titik Awal

No.	Nama Jalan	Simbol	Koordinat	
			<i>Latitude</i>	<i>Longitude</i>
1.	Jl. Raya Slepi	P1	-7.67475	112.61813
2.	Pers. Kecamatan Trowulan	P7	-7.568656	112.403497
3.	Jl. Gempol - Mojokerto	P5	-7.56672	112.65819
4.	Jl. Raya Totok Kerot	P2	-7.559440	112.369550
5.	Jl. Raya Kejagan	P6	-7.541012	112.380153
6.	Jl. Raya Balong Wono	P3	-7.4948	112.37641
7.	Jl. Raya Trawas	P4	-7.6583	112.62511
8.	Pers. Trowulan	P8	-7.55111	112.38071
9.	Jl. RayaTrawas	P9	-7.68384	112.5498
10.	Jl. Raya Pacet	P10	-7.66698	112.53701

Lampiran 4. Script Code

```

from math import radians, sin, cos, sqrt, atan2
from heapq import heappush, heappop
from itertools import count
# -----fungsi Haversine----- #
def haversine(lat1, lon1, lat2, lon2):
    R = 6371 # Radius bumi dalam kilometer
    lat1, lon1, lat2, lon2 = map(radians, [lat1, lon1, lat2, lon2])
    dlat = lat2 - lat1
    dlon = lon2 - lon1
    a = sin(dlat / 2)**2 + cos(lat1) * cos(lat2) * sin(dlon / 2)**2
    c = 2 * atan2(sqrt(a), sqrt(1 - a))
    distance = R * c
    return distance
# -----pendefinisian nilai heuristik----- #
# Posisi tiap titik node
posisi_node = {
    'TitikNode1': [-7.0, 112.0],
    'TitikNode2': [-7.1, 112.1],
    # Tambahkan titik node lainnya sesuai kebutuhan
}
def AmbilAbsis(node):
    return posisi_node[node][0] # Ambil absis dari dictionary posisi_node
def AmbilOrdinat(node):
    return posisi_node[node][1] # Ambil ordinat dari dictionary posisi_node
def AmbilJarak(awal, akhir):
    # Menghitung jarak Haversine sebagai heuristik
    return haversine(AmbilAbsis(awal), AmbilOrdinat(awal), AmbilAbsis(akhir), AmbilOrdinat(akhir))

```

```

# -----algoritma A-star----- #
def astar_wisatamojokerto(G, asal, target, heuristik=None, bobot=None):
    c = count() # Counter untuk menangani prioritas
    ListBuka = [(0, next(c), asal, 0, None)] # Daftar titik terbuka
    Terseleksi = {} # Dictionary titik yang terseleksi
    ListTutup = {} # Dictionary titik yang sudah tertutup
    while ListBuka:
        _, __, node_terkini, dist, pangkal = heappop(ListBuka)
        if node_terkini == target:
            jalur_terpilih = [node_terkini]
            node = pangkal
            while node is not None:
                jalur_terpilih.append(node)
                node = ListTutup.get(node, None)
            jalur_terpilih.reverse()
            return jalur_terpilih
        if node_terkini in ListTutup:
            continue
        ListTutup[node_terkini] = pangkal
        if node_terkini not in G or not G[node_terkini]:
            continue
        for tetangga, w in G[node_terkini]:
            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))
    return []

# -----fungsi jarak antar node----- #
def fungsi_jarak(G, asal, target, heuristik=None, bobot=None):
    lintasan = astar_wisatamojokerto(G, asal, target, heuristik=heuristik, bobot=bobot)
    if not lintasan:
        return float('inf')
    total_jarak = 0
    for i in range(len(lintasan) - 1):
        node_awal = lintasan[i]
        node_tujuan = lintasan[i + 1]
        for tetangga, jarak in G[node_awal]:
            if tetangga == node_tujuan:
                total_jarak += jarak
                break
    return total_jarak

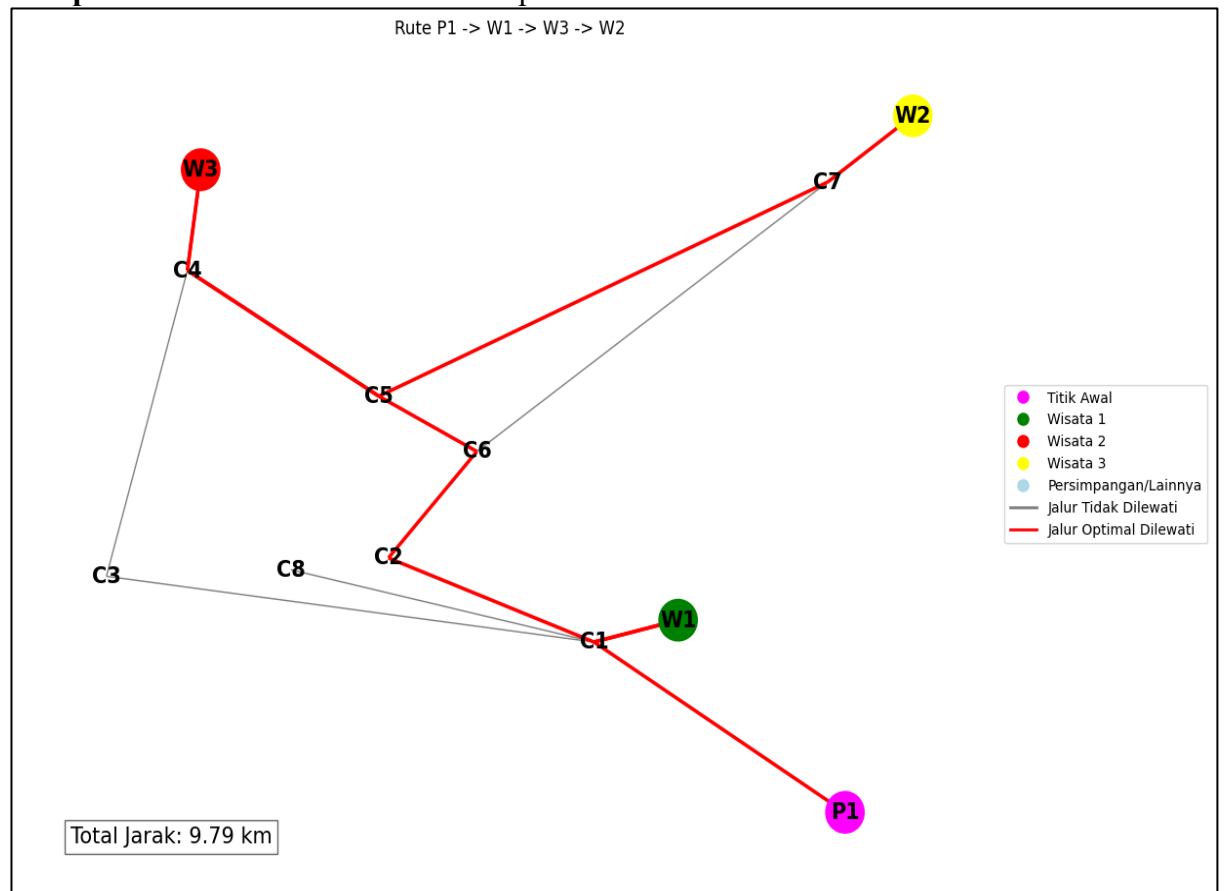
```

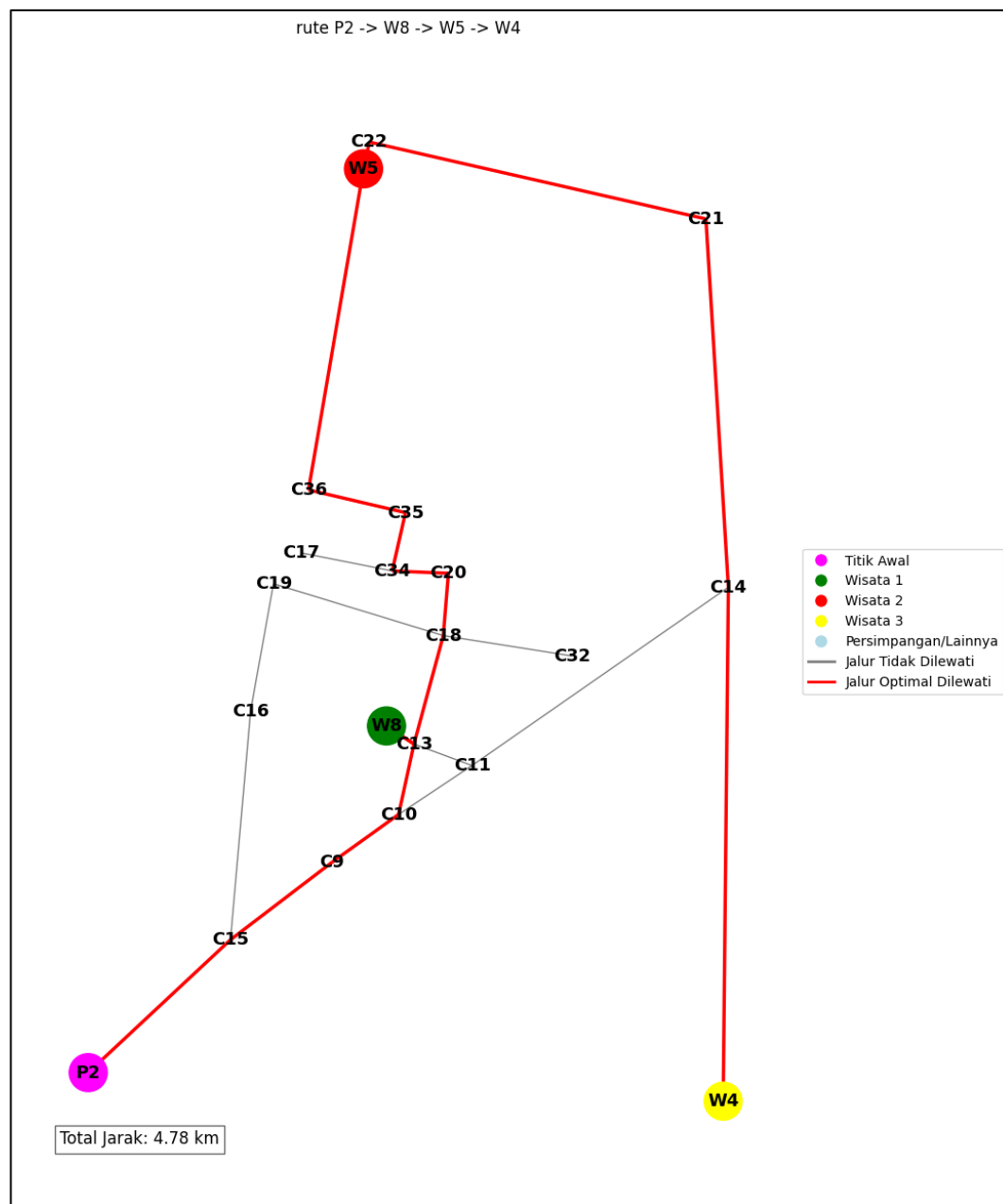
```

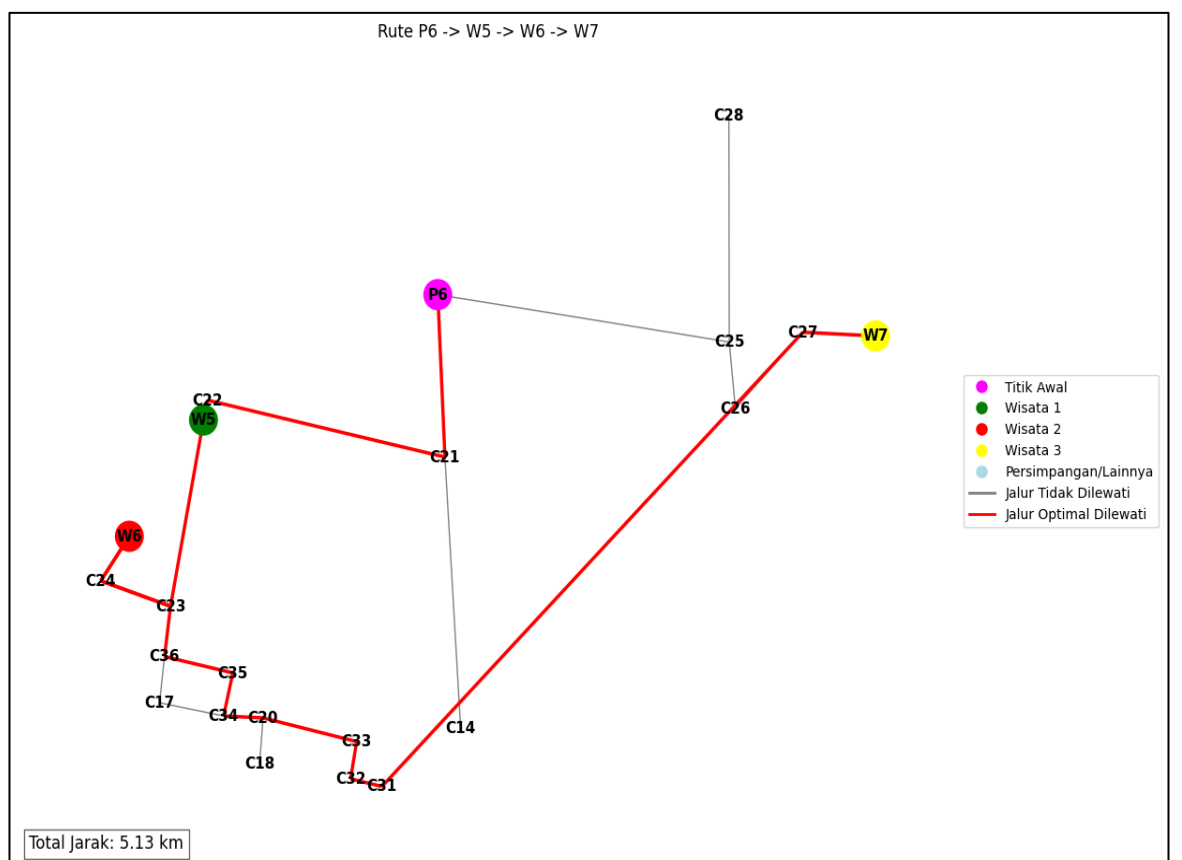
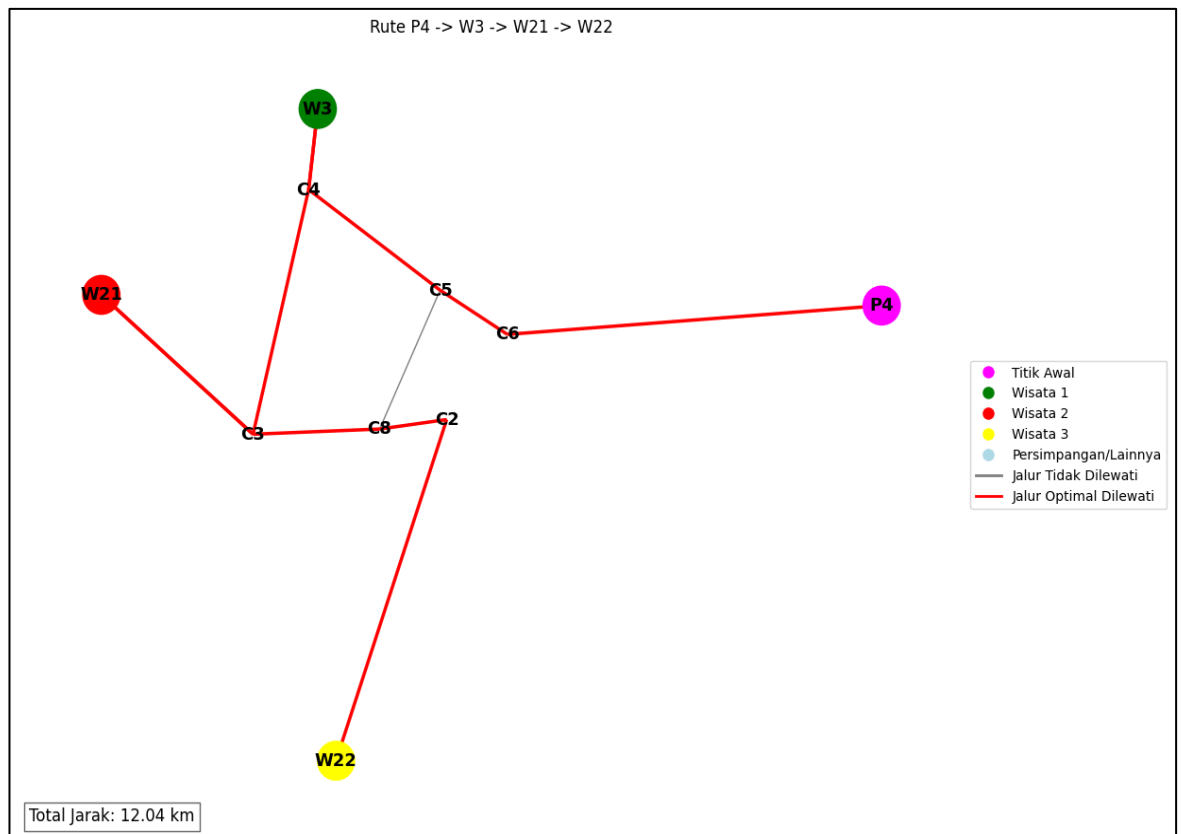
# -----proses pengujian rute terpendek----- #
def seleksi(A, node_tujuan, jalur, dist, graf, heuristik, bobot):
    seleksi1 = []
    if not node_tujuan:
        total = sum(dist)
        print('Jalur : ', jalur)
        print('Total jarak : ', total, ' km')
        return
    else:
        for i in node_tujuan:
            lintasan = astar_wisatamojokerto(graf, A, i, heuristik=heuristik, bobot=bobot)
            if lintasan:
                jarak = fungsi_jarak(graf, A, i, heuristik=heuristik, bobot=bobot)
                seleksi1.append([lintasan, jarak])
        if not seleksi1:
            print(f'Tidak ada jalur yang valid dari {A} ke node tujuan.')
            return
        K1 = min(seleksi1, key=lambda x: x[1])
        seleksi1.clear()
        jalur.extend(K1[0])
        dist.append(K1[1])
        K2 = K1[0][-1]
        if K2 in node_tujuan:
            node_tujuan.remove(K2)
        if not node_tujuan:
            total = sum(dist)
            print('Jalur : ', jalur)
            print('Total jarak : ', total, ' km')
            return
        return seleksi(K2, node_tujuan, jalur, dist, graf, heuristik, bobot)

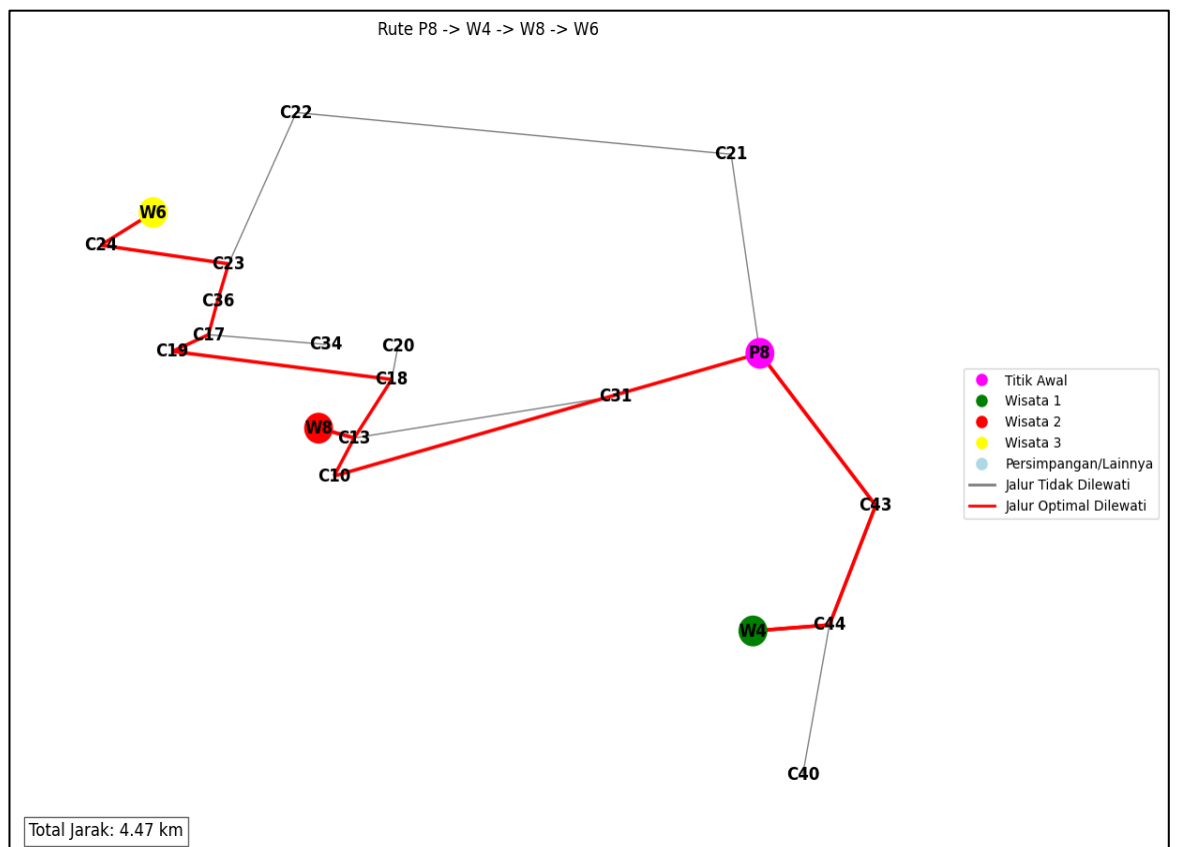
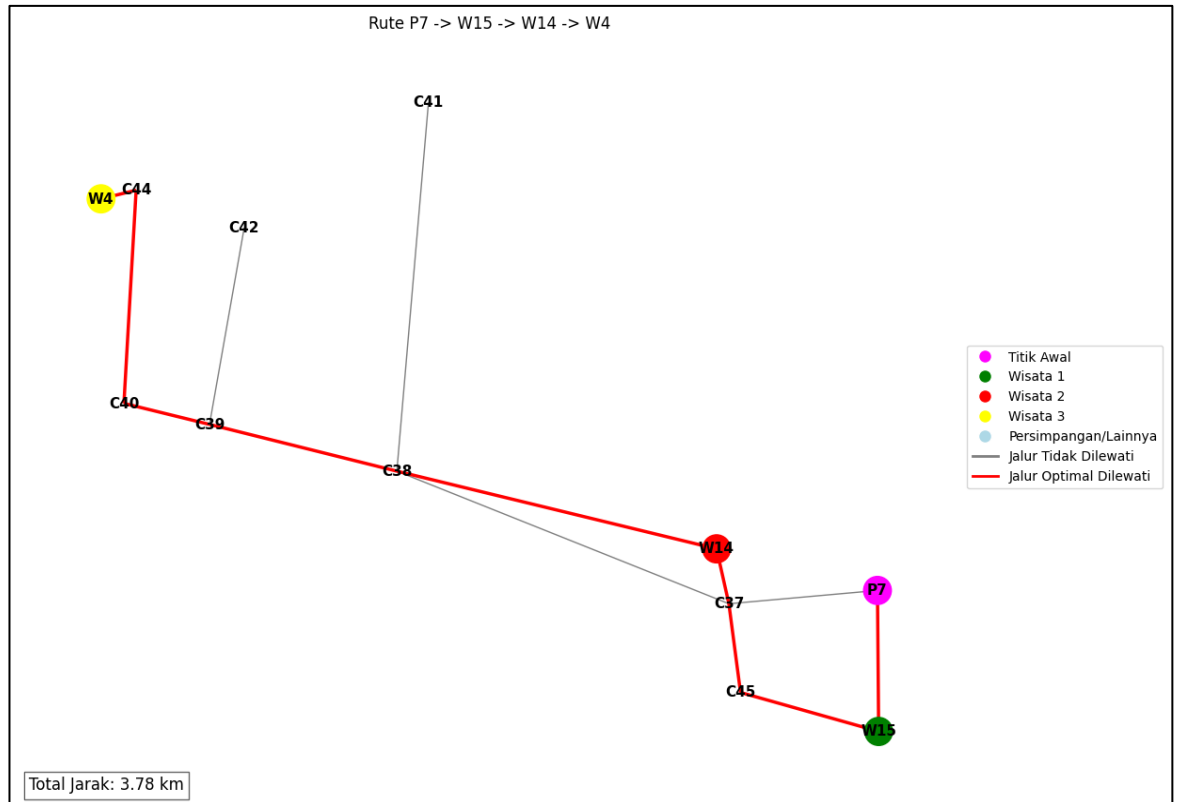
# -----contoh graf dan bobot antar node----- #
def bobot(asal, tujuan, jarak):
    return jarak
# Contoh graf
graf1 = {
    'TitikNode1': [('TitikNode2', 1.34)],
    # Tambahkan titik dan jarak lainnya sesuai kebutuhan
}
# Pengujian
def test_multiple_paths():
    print("Testing Path:")
    node_tujuan = ['TitikNode2']
    seleksi('TitikNode1', node_tujuan, [], [], graf1, AmbilJarak, bobot)
test_multiple_paths()

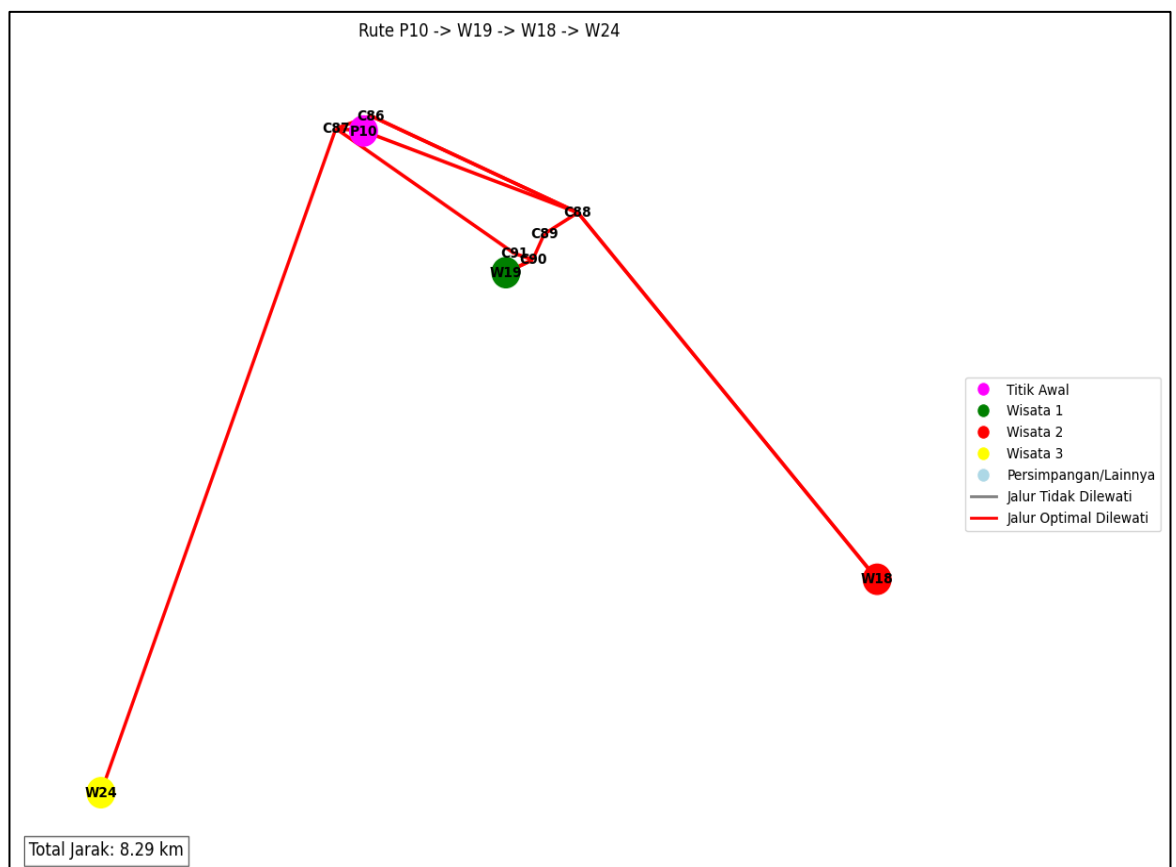
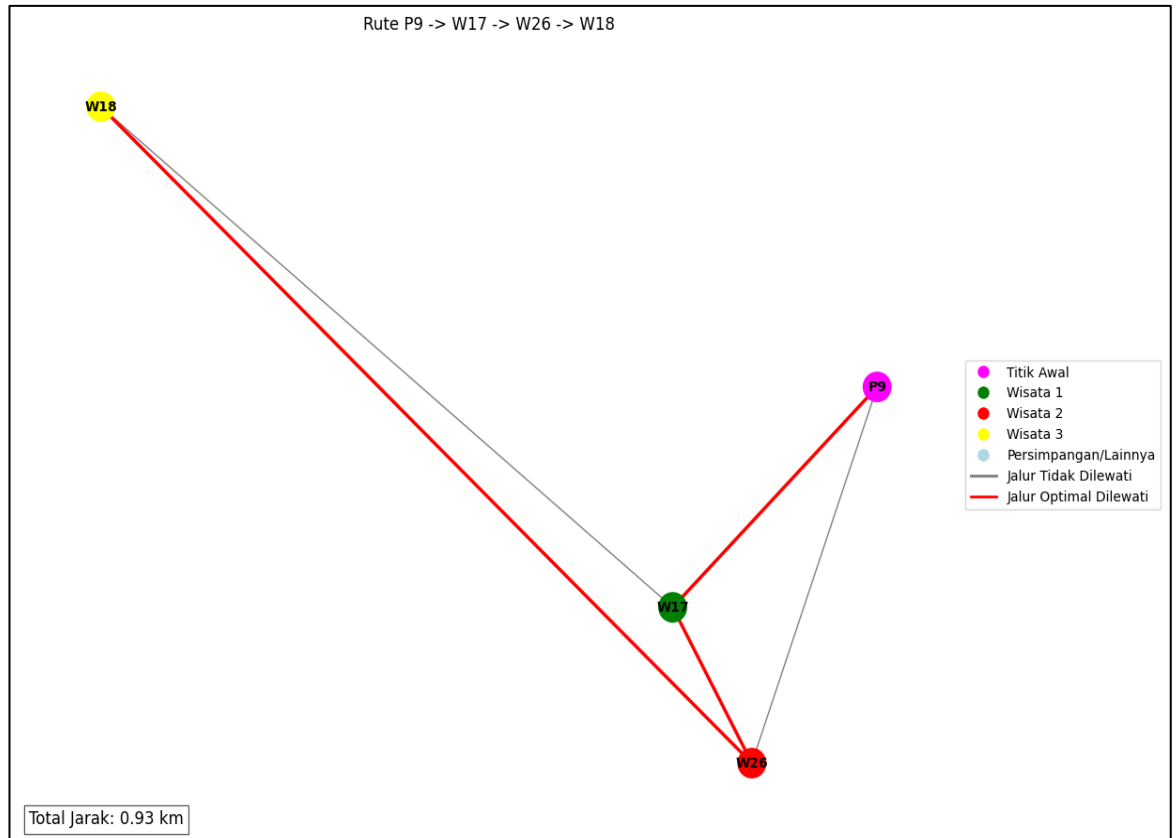
```

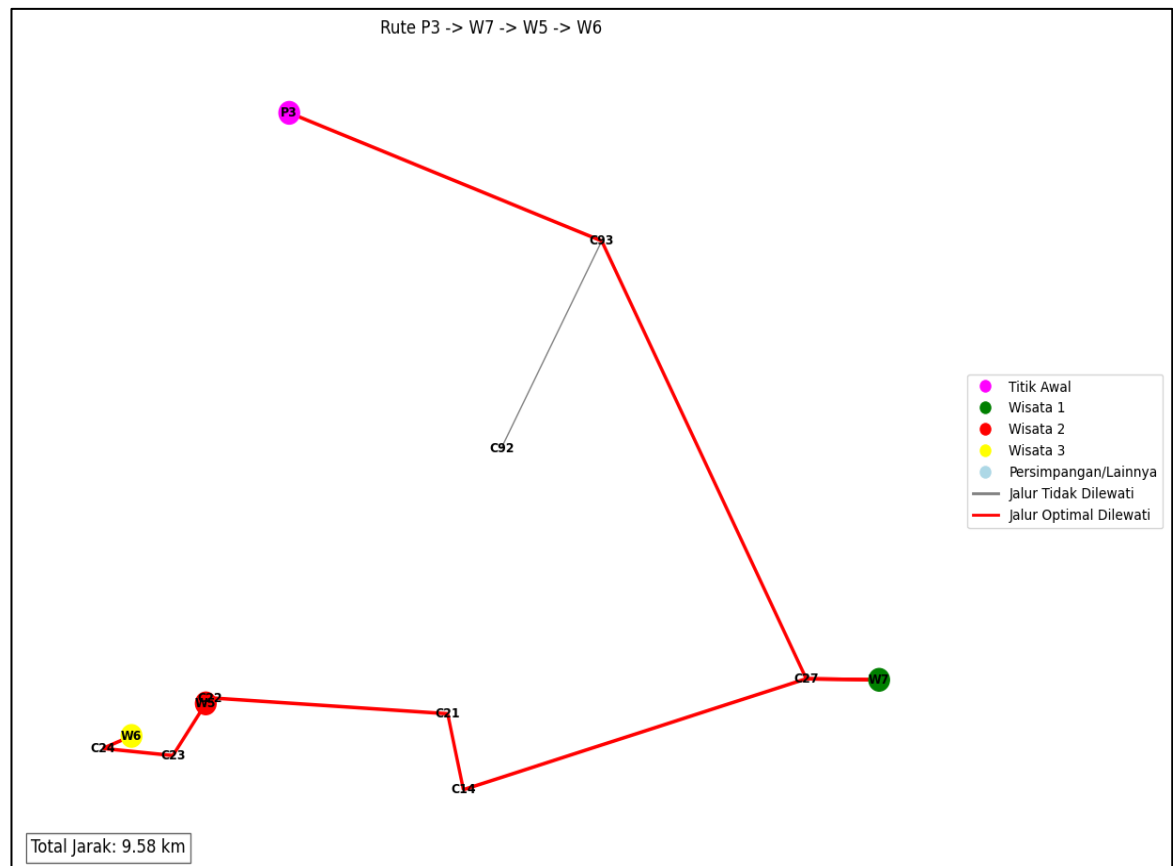
Lampiran 5. Hasil Visualisasi Rute Terpendek

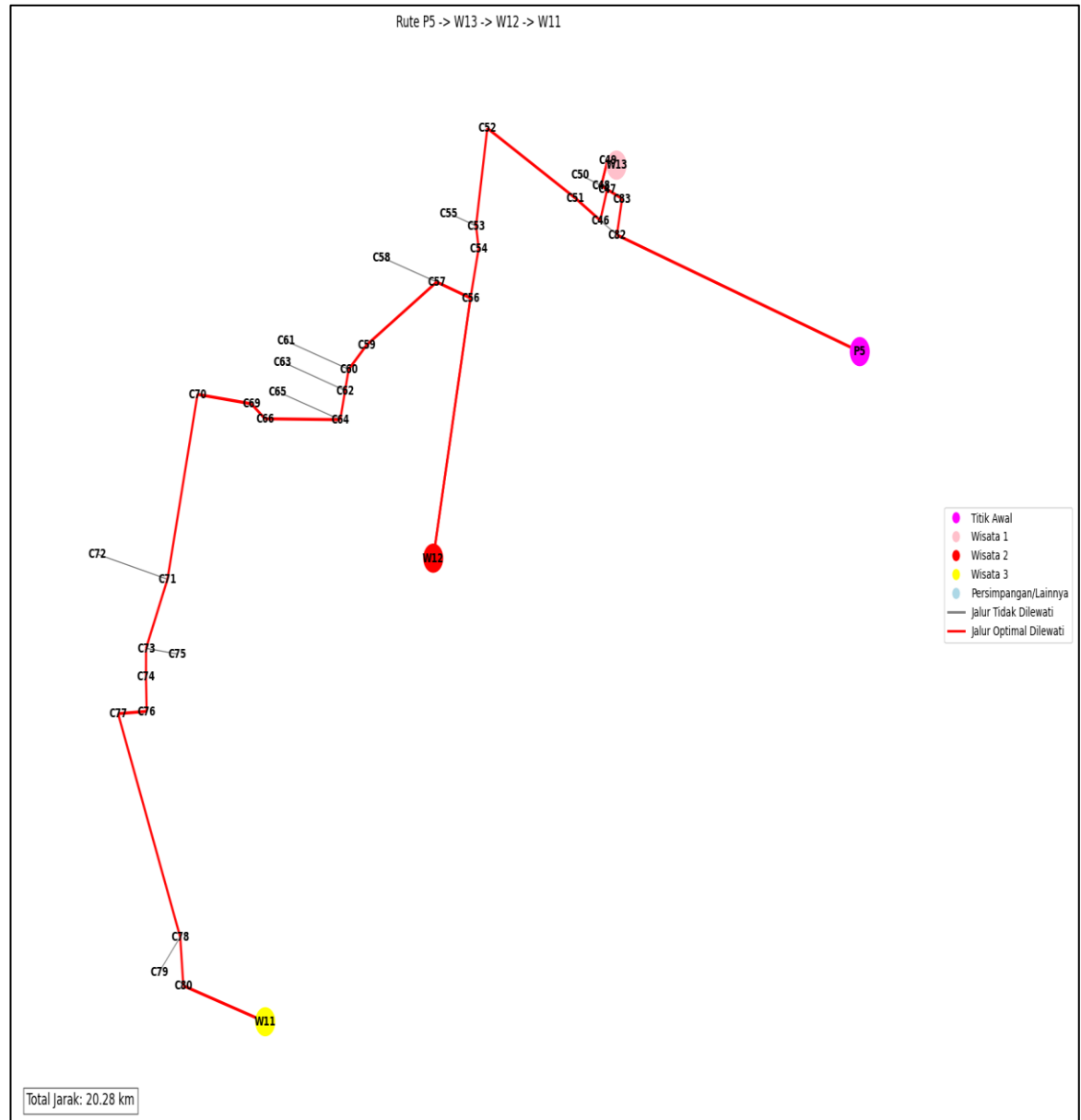












RIWAYAT HIDUP



Silvia Rohmadina, yang akrab dipanggil Silvia atau Dina, lahir di Mojokerto pada 19 November 2002. Penulis merupakan anak kedua dari dua bersaudara, putri dari alm Bapak Solikin dan Ibu Fitriyah. Penulis memulai pendidikannya di TK Dharma Wanita dan lulus pada tahun 2008. Kemudian, ia melanjutkan pendidikan dasar di SDN Purwojati 2 dan menyelesaikannya pada tahun 2014. Pendidikan menengahnya ditempuh di SMP Islam Sedati, dan lulus pada tahun 2017. Setelah itu, penulis bersekolah di SMAN 1 Ngoro jurusan Matematika dan Ilmu Alam (MIA) dan lulus pada tahun 2020. Penulis kemudian melanjutkan ke jenjang Strata-1 (S1) di Universitas Islam Negeri Maulana Malik Ibrahim Malang, mengambil program studi Matematika, Fakultas Sains dan Teknologi. Penulis memiliki pengalaman bekerja di bidang *food and beverage* (FnB). Penulis juga pernah mengikuti praktik kerja lapangan di Badan Amil Zakat Nasional (BAZNAS) Kabupaten Jombang. Selain itu, penulis aktif berpartisipasi dalam berbagai kegiatan di luar kampus, seperti mengikuti pelatihan dan seminar.



KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

BUKTI KONSULTASI SKRIPSI

Nama : Silvia Rohmadina
NIM : 200601110038
Fakultas/Jurusan : Sains dan Teknologi / Matematika
Judul Skripsi : Optimasi Rute Wisata Kabupaten Mojokerto
Menggunakan Algoritma *A-Star* untuk Penentuan Rute Terpendek
Pembimbing I : Juhari, M.Si
Pembimbing II : Evawati Alisah, M.Pd

No	Tanggal	Hal	Tanda Tangan
1.	25 Maret 2024	Konsultasi Bab I, II, dan III	1.
2.	12 Juni 2024	Konsultasi Kajian Agama	2.
3.	14 Juni 2024	Konsultasi Revisi Kajian Agama	3.
4.	14 Juni 2024	ACC Kajian Agama Bab I dan II	4.
5.	12 Juni 2024	Konsultasi Revisi Bab I, II, dan III	5.
6.	13 Juni 2024	ACC Bab I, II, dan III	6.
7.	13 Juni 2024	ACC Seminar Proposal	7.
8.	5 September 2024	Konsultasi Revisi Seminar Proposal	8.
9.	5 September 2024	Konsultasi Bab IV dan V	9.
10.	6 September 2024	Konsultasi Revisi Bab IV dan V	10.
11.	6 September 2024	Konsultasi Kajian Agama Bab IV	11.
12.	6 September 2024	ACC Kajian Agama Bab IV	12.
13.	6 September 2024	ACC Bab IV dan V	13.
14.	6 September 2024	ACC Seminar Hasil	14.



**KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI**

Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

No	Tanggal	Hal	Tanda Tangan
15.	1 November 2024	Konsultasi Revisi Seminar Hasil	15. <i>JH</i>
16.	1 November 2024	ACC Matriks Revisi Seminar Hasil	16. <i>JH</i>
17.	1 November 2024	ACC Sidang Skripsi	17. <i>JH</i>
18.	23 Desember 2024	ACC Keseluruhan	18. <i>JH</i>

Malang,

Mengetahui,

Ketua Program Studi Matematika



Dr. *EDY* Susanti, M.Sc.

NIP. 19741129 200012 2 005