

**PENERAPAN ALGORITMA BELLMAN-FORD
PADA RUTE TERPENDEK DISTRIBUSI AIR MINERAL SANTRI
DI PASURUAN**

SKRIPSI

**OLEH:
SULASTRI
NIM. 210601110002**



**PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2025**

**PENERAPAN ALGORITMA BELLMAN-FORD
PADA RUTE TERPENDEK DISTRIBUSI AIR MINERAL SANTRI
DI PASURUAN**

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:
SULASTRI
NIM.210601110002**

**PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2025**

**PENERAPAN ALGORITMA BELLMAN-FORD
PADA RUTE TERPENDEK DISTRIBUSI AIR MINERAL SANTRI
DI PASURUAN**

SKRIPSI

**Oleh:
Sulastri
NIM. 210601110002**

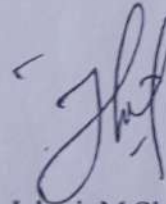
Telah Disetujui Untuk Diuji
Malang, 14 April 2025

Dosen Pembimbing I



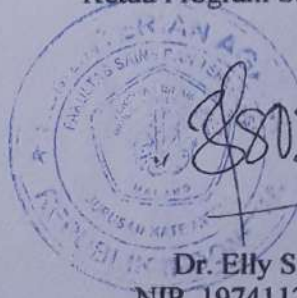
Prof. Dr. H. Turmudi, M.Si., Ph.D
NIP. 19571005 198203 1 006

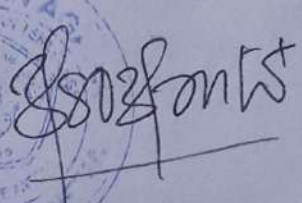
Dosen Pembimbing II



Juhari, M.Si
NIPPPK. 19840209 202321 1 010

Mengetahui,
Ketua Program Studi Matematika




Dr. Elly Susanti, M.Sc
NIP. 19741129 200012 2 00

**PENERAPAN ALGORITMA BELLMAN-FORD
PADA RUTE TERPENDEK DISTRIBUSI AIR MINERAL SANTRI
DI PASURUAN**

SKRIPSI

**Oleh:
Sulastri
NIM. 210601110002**

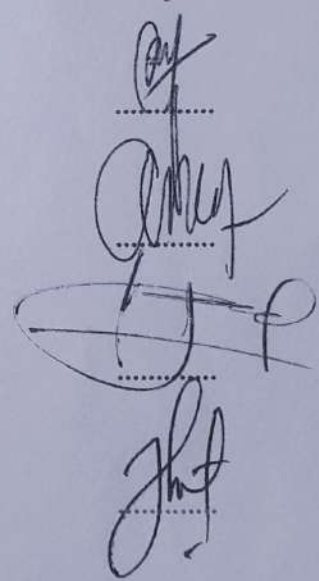
Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Matematika (S.Mat)
Tanggal 10 Juni 2025

Ketua Penguji : Ari Kusumastuti, M.Si., M.Pd

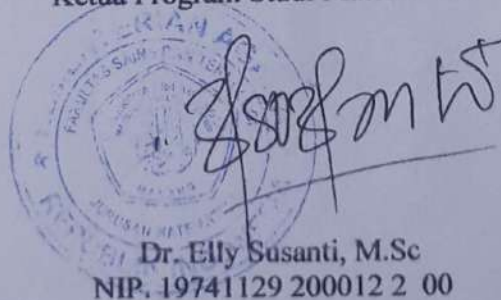
Anggota Penguji 1 : Dr. Mohammad Jamhuri, M.Si

Anggota Penguji 2 : Prof. Dr. H. Turmudi, M.Si., Ph.D

Anggota Penguji 3 : Juhari, M.Si



Mengetahui,
Ketua Program Studi Matematika


Dr. Elly Susanti, M.Sc
NIP. 19741129 200012 2 00

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini

Nama : Sulastri

NIM : 210601110002

Program Studi : Matematika

Fakultas : Sains dan Teknologi

Judul Skripsi : Penerapan Algoritma Bellman-Ford pada Rute Terpendek
Distribusi Air Mineral Santri di Pasuruan

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini merupakan hasil karya sendiri. Bukan pengambilan tulisan, data, atau pemikiran orang lain yang saya akui sebagai hasil tulisan dan pemikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila di kemudian hari terbukti skripsi ini adalah hasil jiplakan atau tiruan, maka saya bersedia menerima sanksi yang berlaku atas perbuatan tersebut.

Malang, 10 Juni 2025

Yang membuat pernyataan,



Sulastri

NIM. 210601110002

MOTO

“Perlakuanmu kepada orang tuamu saat ini akan menjadi cerminan
perlakuan anakmu kepadamu di masa depan”

(Huququ Da'at ilaihal Fitroh: 12)

HALAMAN PERSEMBAHAN

Bismillahirrahmanirrahim, Hamdan wa Syukron Lillah. Skripsi ini dipersembahkan untuk kedua orang tua penulis yakni Ayahanda Suyani dan almarhumah Ibunda Maryam, semua saudara penulis baik saudara kandung maupun saudara ipar dan seluruh keluarga besar yang senantiasa memberikan dukungan juga doa kepada penulis dalam menyelesaikan skripsi ini. Semua dosen yang membantu dan membimbing proses pengerjaan skripsi ini. Ustadz dan Ustadzah Pondok Pesantren Imam Ad-Damanhuri yang selalu memberikan motivasi dan doa kepada penulis. Teman-teman yang membersamai, memberikan semangat, dan bantuan kepada penulis untuk menyelesaikan skripsi ini.

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT yang telah melimpahkan rahmat, hidayah, dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul "Penerapan Algoritma Bellman-Ford pada Rute Terpendek Distribusi Air Mineral Santri di Pasuruan". Sholawat serta salam semoga senantiasa tercurah kepada junjungan kita Nabi Muhammad SAW, yang telah membawa umat manusia dari zaman kegelapan menuju zaman yang penuh cahaya ilmu pengetahuan, serta kepada keluarga, sahabat, dan seluruh pengikut beliau hingga akhir zaman.

Penelitian ini disusun sebagai salah satu syarat untuk menyelesaikan studi di Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Dalam proses penyusunan hasil penelitian ini, penulis mendapatkan bimbingan, dukungan, dan bantuan dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati, penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Prof. Dr. H. M. Zainuddin, M.A., selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Sri Harini, M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Elly Susanti, S.Pd., M.Sc, selaku Ketua Program Studi Matematika Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Prof. Dr. H. Turmudi, M. Si., Ph.D, selaku Dosen Pembimbing I, yang telah memberikan bimbingan, masukan, dan kritik yang sangat berharga dalam penyelesaian penelitian ini.
5. Juhari, M.Si, selaku Dosen Pembimbing II, yang dengan sabar memberikan arahan dan motivasi dalam penyusunan penelitian ini.
6. Seluruh Dosen Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim, yang telah memberikan ilmu dan pengetahuan yang sangat berharga selama masa studi.
7. Terkhusus Bapak Suyani dan Ibu Maryam selaku orang tua penulis, beserta seluruh keluarga, yang selalu memberikan dukungan moral, material, serta doa yang tidak pernah henti kepada penulis.

8. Keluarga besar PP Imam Addamanhuri, atas dukungan dan doa yang senantiasa menguatkan penulis dalam menyelesaikan penelitian ini.
9. Seluruh mahasiswa angkatan 2021 “TEOREMA”, yang telah menjadi teman dan partner belajar yang saling mendukung selama masa perkuliahan.
10. Semua pihak yang tidak dapat penulis tuliskan secara keseluruhan, yang telah membantu penulis dalam menyelesaikan proposal ini.

Malang, 10 Juni 2025

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTO	vi
HALAMAN 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	5
1.3 Tujuan Penelitian	5
1.4 Manfaat Penelitian	5
1.5 Batasan Masalah	6
BAB II KAJIAN TEORI	7
2.1 Graf	7
2.1.1 Definisi Graf	7
2.1.2 Jenis-Jenis Graf	7
2.1.3 Rute Terpendek	10
2.1.4 <i>Hamiltonian Cycle</i>	11
2.2 Algoritma Bellman-Ford	11
2.3 Distribusi Air Mineral Santri	13
2.4 Kajian Integrasi dengan Al-Quran dan Hadits	14
2.5 Kajian Penerapan Algoritma Bellman-Ford dengan Teori Pendukung	16
BAB III METODE PENELITIAN	18
3.1 Jenis Penelitian	18
3.2 Data dan Sumber Data	18
3.3 Tahapan Penelitian	20
3.3.1 Algoritma Bellman-Ford dengan Siklus <i>Hamiltonian Cycle</i>	20
3.3.2 Tahapan Implementasi Algoritma Bellman-Ford	23
3.3.3 Tahapan Evaluasi Penelitian	25
3.3.4 <i>Flowchart</i> Tahapan Penelitian	25
BAB IV HASIL DAN PEMBAHASAN	27
4.1 Algoritma Bellman-Ford dengan Siklus <i>Hamiltonian</i> <i>Cycle</i> guna Menentukan Rute Terpendek Distribusi Air Mineral Santri di Pasuruan	27

4.2 Implementasi Algoritma Bellman-Ford dalam Menentukan Rute Terpendek Distribusi Air Mineral Santri.	54
4.3 Evaluasi Penelitian	56
4.4 Integrasi Al-Qur'an dalam Penentuan Rute Terpendek Distribusi	57
BAB V KESIMPULAN DAN SARAN	62
5.1 Kesimpulan.....	62
5.2 Saran	63
DAFTAR RUJUKAN	64
LAMPIRAN.....	66
RIWAYAT HIDUP	72

DAFTAR TABEL

Tabel 3.1	Titik Koordinat Lokasi Distribusi	19
Tabel 4.1	Data ke 1, Jarak Sebelum Penambahan Bobot Negatif.....	27
Tabel 4.2	Data ke 2, Jarak Sebelum Penambahan Bobot Negatif.....	28
Tabel 4.3	Data Ke 3, Jarak Sebelum Penambahan Bobot Negatif.....	28
Tabel 4.4	Data ke 2, Jarak Sebelum Penambahan Bobot Negatif.....	29
Tabel 4.5	Besar Pengurangan Bobot pada Titik Tertentu	29
Tabel 4.6	Data ke 1, Jarak Setelah Penambahan Bobot Negatif.....	30
Tabel 4.7	Data ke 2, Jarak Setelah Penambahan Bobot Negatif.....	30
Tabel 4.8	Data ke 3, Jarak Setelah Penambahan Bobot Negatif.....	31
Tabel 4.9	Data ke 4, Jarak Setelah Penambahan Bobot Negatif.....	31
Tabel 4.10	Iterasi 1	34
Tabel 4.11	Iterasi 2.....	35
Tabel 4.12	Iterasi 3	36
Tabel 4.13	Iterasi 4.....	36
Tabel 4.14	Iterasi 5.....	37
Tabel 4.15	Iterasi 6.....	38
Tabel 4.16	Iterasi 7.....	39
Tabel 4.17	Iterasi 8.....	40
Tabel 4.18	Iterasi 9.....	40
Tabel 4.19	Iterasi 10.....	41
Tabel 4.20	Iterasi 11.....	42
Tabel 4.21	Iterasi 12.....	42
Tabel 4.22	Iterasi 13.....	43
Tabel 4.23	Iterasi 14.....	44
Tabel 4.24	Iterasi 15.....	44
Tabel 4.25	Iterasi 16.....	45
Tabel 4.26	Iterasi 17.....	46
Tabel 4.27	Iterasi 18.....	46
Tabel 4.28	Iterasi 19.....	47
Tabel 4.29	Iterasi 20.....	47
Tabel 4.30	Iterasi 21.....	48
Tabel 4.31	Iterasi 22.....	48
Tabel 4.32	Iterasi 23.....	49
Tabel 4.33	Iterasi 24.....	49
Tabel 4.34	Iterasi 25.....	50
Tabel 4.35	Iterasi 26.....	50
Tabel 4.36	Iterasi 27.....	50
Tabel 4.37	Iterasi 28.....	51
Tabel 4.38	Iterasi 29.....	51
Tabel 4.39	Rute terpendek dari titik A ke titik lainnya	55

DAFTAR GAMBAR

Gambar 2.1	Graf Sederhana dengan 5 simpul	7
Gambar 2.2	Graf Sederhana	8
Gambar 2.3	Graf Ganda dan Graf Semu	9
Gambar 2.4	Graf Berarah	9
Gambar 2.5	Graf Berbobot	10
Gambar 3.1	<i>Flowchart</i> Tahapan Penelitian	26
Gambar 4.1	Graf yang menghubungkan semua toko di Pasuruan, dengan A adalah	33
Gambar 4.2	Graf Rute Terpendek dengan Sifat <i>Hamiltonian Cycle</i>	52
Gambar 4.3	Visualisasi Graf dengan Sifat <i>Hamiltonian Cycle</i> pada Peta	53
Gambar 4.4	Graf Rute Terpendek Tanpa Siklus <i>Hamiltonian Cycle</i>	56
Gambar 4.5	Visualisasi Graf Rute Terpendek Tanpa Siklus <i>Hamiltonian Cycle</i> pada Peta	57

DAFTAR SIMBOL

$ V $	: Jumlah simpul pada graf
$ E $	: Jumlah sisi pada graf
G	: Graf G
$V(G)$	: Himpunan simpul pada graf
$E(G)$	: Himpunan sisi pada graf
$V(D)$	: Himpunan simpul pada digraf
$E(D)$	: Himpunan sisi pada digraf
u	: Anggota Himpunan simpul pada graf
v	: Anggota Himpunan simpul pada graf
e	: Sisi pada graf yang menghubungkan u ke v dan sebaliknya
$w(e)$	: Bobot pada sisi-sisi graf
$w(u, v)$	: Jarak antara simpul u ke simpul v yang dijadikan bobot pada sisi uv
$d[u, v]$	: Jarak dari simpul asal u ke simpul v
v_n	: Simpul ke- n pada graf
$\mathbb{R}^+$	: Himpunan bilangan riil positif
D	: Himpunan data yang menyimpan jarak minimum sementara

DAFTAR LAMPIRAN

Lampiran 1	Kode <i>Google Collab</i> untuk Membuat Graf Berarah dan Berbobot.....	66
Lampiran 2	Kode <i>Google Collab</i> untuk Membuat Graf Rute Terpendek dengan Sifat <i>Hamiltonian Cycle</i>	67
Lampiran 3	Kode <i>Google Collab</i> Membuat Graf Rute Terpendek dari A ke titik lainnya	69
Lampiran 4	Mencari rute terpendek dengan Algoritma Bellman-Ford memakai bahasa pemrograman <i>python</i>	71

ABSTRAK

Sulastrri, Sulastrri. 2025. **Penerapan Algoritma Bellman-Ford pada Rute Terpendek Distribusi Air Mineral Santri di Pasuruan**. Skripsi. Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Prof. Dr. H. Turmudi, M.Si., Ph.D. (II) Juhari, M.Si.

Kata Kunci: Algoritma Bellman-Ford, Rute Terpendek, Graf.

Algoritma Bellman-Ford yang dikenal mampu menangani graf berbobot negatif, terutama pada kondisi distribusi yang kompleks. Distribusi Air Minum Dalam Kemasan (AMDK) Santri di wilayah Pasuruan menghadapi tantangan dalam penjadwalan rute yang belum optimal, sehingga dapat menyebabkan ketidakefisienan dalam pengiriman. Penelitian ini dilakukan untuk mengembangkan modifikasi Algoritma Bellman-Ford agar dapat menemukan rute optimal dengan sifat *Hamiltonian Cycle* serta mengimplementasikan Algoritma Bellman-Ford ke dalam model jaringan distribusi. Penelitian ini menerapkan Penelitian dilakukan menggunakan metode studi literatur serta studi lapangan melalui pemetaan rute dengan *Google Maps*. Hasil penelitian menunjukkan bahwa total jarak rute terpendek dengan siklus *Hamiltonian Cycle* adalah 97,48 km, sedangkan rute terpendek tanpa siklus *Hamiltonian Cycle* adalah titik A , v_{12} , v_{13} , v_{10} , v_{17} sejauh 3,49 km (pulang-pergi). Kesimpulan dari penelitian ini menunjukkan bahwa penerapan Algoritma Bellman-Ford dengan siklus Hamiltonian memberikan hasil yang lebih optimal untuk kebutuhan distribusi menyeluruh dibandingkan hanya mencari rute tercepat antara dua titik. Apabila hanya menggunakan Algoritma Bellman-Ford tanpa modifikasi, maka beberapa titik distribusi tidak dapat dijangkau. Penelitian selanjutnya disarankan untuk membandingkan algoritma ini dengan pendekatan lain seperti A^* atau Floyd-Warshall guna mengevaluasi efisiensi dan keakuratannya secara lebih mendalam.

ABSTRACT

Sulastri, Sulastri. 2025. **Application of the Bellman-Ford Algorithm to the Shortest Route of Santri Mineral Water Distribution in Pasuruan.** Thesis.
Department of Mathematics, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Supervisors: (I) Prof. Dr. H. Turmudi, M.Si., Ph.D. (II) Juhari, M.Si.

Keywords: Bellman-Ford Algorithm, Shortest Route, Graph.

The Bellman-Ford algorithm is known for its ability to handle graphs with negative weights, particularly in complex distribution conditions. The distribution of Santri Bottled Drinking Water (AMDK Santri) in the Pasuruan region faces challenges due to suboptimal route scheduling, which can lead to inefficiencies in delivery. This research was conducted to develop a modification of the Bellman-Ford Algorithm in order to find the optimal route with the properties of a Hamiltonian Cycle and to implement the Bellman-Ford Algorithm into a distribution network model. This study was conducted using literature review and field study methods through route mapping with Google Maps. The results show that the total shortest route distance using the Hamiltonian Cycle is 97,48 km, while the shortest route without the Hamiltonian Cycle is between point A, $v_{12}, v_{13}, v_{10}, v_{17}$ with a round-trip distance of 3,49 km. The conclusion of this study indicates that applying the Bellman-Ford Algorithm with the Hamiltonian Cycle yields more optimal results for comprehensive distribution needs compared to simply finding the fastest route between two points. If the unmodified Bellman-Ford Algorithm is used, several distribution points cannot be reached.

مستخلص البحث

سولاستري, سولاستري. ٢٠٢٥. "تطبيق خوارزمية بيلمان-فورد في أقصر مسار لتوزيع مياه الشرب المعبأة سانتري في باسوروان". البحث العلمي. قسم الرياضيات, كلية العلوم والتكنولوجيا, جامعة مولانا مالك إبراهيم الإسلامي الحكومية مالانج. المشرف : (١) الأستاذ الدكتور الأساسية تورمذي, الماجستير في العلوم والفلسفة, الحاج. (٢) جوهري الماجستير في العلوم.

الكلمات المفتاحية: خوارزمية بيلمان-فورد، أقصر مسار، الرسم البياني.

تُعرف خوارزمية بيلمان-فورد بقدرتها على معالجة الرسوم البيانية ذات الأوزان السالبة، وخاصة في ظروف التوزيع المعقدة. يواجه توزيع مياه الشرب المعبأة (AMDK) سانتري في منطقة باسوروان تحديات تتعلق بجدولة المسارات التي لم تُحسن بالشكل الأمثل، مما قد يؤدي إلى عدم الكفاءة في عمليات التوصيل. أُجري هذا البحث لتطوير تعديل على خوارزمية بيلمان-فورد بهدف إيجاد المسار الأمثل الذي يتميز بخاصية الدورة الهاميلتونية، وكذلك لتطبيق خوارزمية بيلمان-فورد في نموذج شبكة التوزيع. تم إجراء هذا البحث باستخدام منهجية الدراسة الأدبية بالإضافة إلى الدراسة الميدانية من خلال رسم خرائط المسارات باستخدام خرائط جوجل. أظهرت نتائج البحث أن المسافة الإجمالية لأقصر مسار باستخدام دورة هاميلتوني تبلغ ٩٧,٤٨ كيلومتر، في حين أن أقصر مسار بدون دورة هاميلتوني هو بين النقطة A و v_{12} و v_{13} و v_{10} و v_{17} يبلغ ٣,٤٩ كيلومتر ذهابًا وإيابًا. وتشير نتائج هذا البحث إلى أن تطبيق خوارزمية بيلمان-فورد مع دورة هاميلتوني يوفر نتائج أكثر كفاءة لاحتياجات التوزيع الشاملة مقارنةً بمجرد البحث عن أسرع طريق بين نقطتين فقط. وإذا تم استخدام خوارزمية بيلمان-فورد دون تعديل، فلن يكون بالإمكان الوصول إلى بعض نقاط التوزيع. ويوصى في الأبحاث القادمة بمقارنة هذه الخوارزمية مع مناهج أخرى مثل خوارزمية A* أو فلويد-ورشال لتقييم الكفاءة والدقة بشكل أعمق.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Algoritma Bellman-Ford pertama kali diperkenalkan oleh Richard Bellman pada tahun 1958. Pada tahun yang hampir bersamaan Lester Ford juga mengemukakan Algoritma yang serupa, sehingga algoritma ini disebut dengan istilah Algoritma Bellman-Ford (Neshia et al., 2016). Algoritma ini hampir mirip dengan Algoritma Dijkstra akan tetapi algoritma ini dapat menangani bobot negatif untuk menemukan rute terpendek. Algoritma Bellman-Ford adalah pengembangan dari algoritma Dijkstra, pengembangan tersebut berfokus pada strategi pemilihan simpul yang mana pada algoritma Dijkstra bekerja secara *greedy* dengan memilih simpul berjarak terpendek yang sudah diketahui, sedangkan Bellman-Ford menggantikan metode ini dengan melakukan relaksasi pada semua sisi sebanyak $|V| - 1$ kali, sehingga lebih cocok untuk graf dengan bobot negatif. Algoritma Bellman-Ford akan benar jika dan hanya jika graf tidak memiliki siklus berbobot negatif yang diperoleh dari titik asal (Hamdi & Prihandoko, 2018).

Algoritma Bellman-Ford memiliki kompleksitas waktu dengan sebuah kemungkinan, yakni dengan memakai notasi *Big-O* dituliskan dengan $O(|V| \cdot |E|)$. Di mana $|V|$ merupakan jumlah simpul dan $|E|$ merupakan jumlah sisi pada graf. Oleh karena itu untuk simpul atau sisi dalam jumlah besar akan mengakibatkan algoritma Bellman-Ford berjalan lebih lama dari pada Algoritma Dijkstra. Meskipun memiliki kompleksitas waktu yang lebih tinggi dari pada Algoritma Dijkstra (yang memiliki kompleksitas $O((V + E) \log V)$) Algoritma Bellman-Ford mempunyai keunggulan dalam menangani graf dengan bobot negatif. (Retanto, 2009).

Menurut karakteristik tersebut, Algoritma Bellman-Ford dipilih dalam penelitian ini untuk menyelesaikan permasalahan distribusi rute yang kompleks, terutama dalam kasus distribusi air mineral yang titik awal sudah diketahui dan dapat melibatkan kondisi graf dengan bobot negatif, seperti pengurangan bobot pada lokasi yang strategis dikarenakan kebutuhan produk lebih banyak dari pada lokasi lainnya. Keunggulan algoritma ini dalam mengakomodasi kondisi graf yang tidak ideal menjadikannya pilihan yang tepat untuk studi rute terpendek. Implementasi algoritma yang relatif sederhana memungkinkan analisis dilakukan secara menyeluruh terhadap semua sisi graf, sehingga tidak ada jalur potensial yang terlewatkan selama semua jalur terhubung (Br Ketaren & Marpaung, 2023).

Algoritma Bellman-Ford dapat digunakan untuk diterapkan pada sistem distribusi di dunia nyata. Algoritma ini sangat berguna dalam optimasi jaringan, transportasi, dan distribusi barang, di mana pemilihan rute yang optimal berperan penting dalam mengurangi jarak tempuh. Titik distribusi yang tersebar di seluruh kota dan kabupaten mengharuskan penentuan rute yang tepat untuk meningkatkan efisiensi pengiriman (Fazri & Puspita, 2015).

Air Minum Dalam Kemasan (AMDK) merupakan salah satu kebutuhan pokok yang dikonsumsi masyarakat secara luas. AMDK Santri sebagai salah satu produk lokal mengalami peningkatan permintaan, terutama setelah munculnya tren peralihan konsumen terhadap produk halal dan lokal, serta adanya boikot terhadap beberapa produk air mineral lainnya. Hal ini menjadikan distribusi AMDK Santri sebagai bagian penting dalam menjaga ketersediaan produk di tingkat konsumen.

Proses pengiriman armada di wilayah Pasuruan masih belum optimal karena pengiriman dilakukan hanya berdasarkan permintaan toko pada hari itu tanpa sistem penjadwalan atau rute yang efisien. Ketidakteraturan ini menyebabkan

rute pengiriman kurang efisien dan berisiko menghambat distribusi, terutama saat permintaan meningkat. Pendekatan algoritmik merupakan salah satu upaya untuk menentukan rute terpendek dalam distribusi, yang dapat dimodelkan sebagai graf berarah berbobot. Algoritma Bellman-Ford sangat tepat digunakan karena titik awalnya, yakni pabrik AMDK Santri, sudah diketahui.

Berdasarkan studi-studi sebelumnya, algoritma ini sudah diimplementasikan guna menyelesaikan berbagai masalah praktis. Beberapa di antaranya ialah penelitian yang dilakukan oleh Hasugian (2015) menunjukkan bahwa Algoritma Bellman-Ford merupakan prosedur yang digunakan dalam menemukan rute pengiriman barang secara efisien di daerah perkotaan. Hutasoit (2019) dalam penelitiannya memberikan jalur terpendek dalam pencarian dari kantor PT JNE Medan ke Haltevia dengan algoritma Bellman-Ford. Penelitian serupa juga dilakukan oleh Br Ketaren & Marpaung (2023), dalam penelitiannya Br Ketaren & Marpaung menerapkan algoritma ini dalam konteks distribusi barang oleh PT. Global Jet Cargo (J&T Cargo), yang mengalami permasalahan dalam penyusunan rute distribusi tetap. Penelitian tersebut menunjukkan bahwa dengan menggunakan Algoritma Bellman-Ford, dapat ditemukan jalur distribusi dengan jarak tempuh paling pendek, yakni 55,9 km, yang membantu mengurangi ketidaktepatan waktu pengiriman.

Berdasarkan beberapa penelitian di atas, terlihat bahwa Algoritma Bellman-Ford tidak hanya relevan dalam konteks teoritis tetapi juga memiliki aplikasi nyata dalam kehidupan sehari-hari, khususnya dalam bidang logistik dan distribusi dengan jasa J&T dan JNE. Studi sebelumnya hanya terbatas pada lingkup kota, sebagaimana penelitian yang dilakukan Hutasoit pada tahun 2019. Adanya penelitian ini menghadirkan nilai tambah dengan menerapkan Algoritma Bellman-

Ford dalam konteks distribusi air mineral *Santri* di wilayah Kota dan Kabupaten Pasuruan. Penelitian ini juga mengkombinasikan antara Algoritma Bellman-Ford dengan *Hamiltonian Cycle* guna mencari rute terpendek, hal ini memastikan semua titik terjangkau oleh rute yang dilewati, yang mana hal ini belum ada pada penelitian sebelumnya.

Menurut perspektif Islam, ilmu pengetahuan dan teknologi dipandang sebagai alat untuk mendekatkan diri kepada Allah dan meningkatkan kesejahteraan manusia. Al-Qur'an mengandung banyak ayat yang mendorong manusia untuk berpikir dan menggunakan akal dalam memahami ciptaan-Nya. Salah satu ayat yang relevan dalam konteks algoritma dan upaya mencari solusi optimal adalah Surah Al-Mulk ayat 15 yang artinya:

"Dialah yang menjadikan bumi itu mudah bagi kamu, maka berjalanlah di segala penjurunya dan makanlah sebagian dari rezeki-Nya. Dan hanya kepada-Nya-lah kamu (kembali setelah) dibangkitkan." (QS. Al-Mulk: 15)

Ayat di atas mengandung pesan agar manusia memanfaatkan segala sumber daya yang ada di bumi, termasuk ilmu pengetahuan, untuk memperoleh manfaat dan mencapai tujuan hidup yang lebih baik. Prinsip algoritma, seperti Bellman-Ford, dapat diinterpretasikan sebagai bentuk ikhtiar manusia dalam mencari jalur terbaik atau solusi optimal di antara berbagai pilihan. Sama seperti dalam ayat tersebut, di mana manusia diajak untuk berjalan dan menjelajah berbagai arah, algoritma ini berfungsi untuk menemukan jalur terbaik dari satu titik ke titik lainnya di dalam graf.

Surah Ar-Ra'd ayat 11 juga memberikan pandangan tentang pentingnya usaha manusia dalam memperbaiki keadaan. Berikut ini arti dari Surah Ar-Ra'd ayat 11:

"Sesungguhnya Allah tidak akan mengubah keadaan suatu kaum sebelum mereka mengubah keadaan diri mereka sendiri." (QS. Ar-Ra'd: 11)

Ayat di atas menunjukkan bahwasannya perubahan hanya dapat terjadi jika manusia melakukan usaha atau ikhtiar. Algoritma Bellman-Ford dapat dilihat sebagai suatu cara untuk mengubah keadaan, dari kondisi yang belum optimal menjadi kondisi yang lebih baik melalui pencarian jalur terpendek. Algoritma ini berusaha untuk mencari solusi terbaik melalui proses perhitungan yang sistematis, sebagaimana manusia diminta untuk terus berusaha memperbaiki diri dan lingkungan.

1.2 Rumusan Masalah

Berdasarkan persoalan dalam latar belakang, maka rumusan masalah pada penelitian ini antara lain.

1. Bagaimana Algoritma Bellman-Ford dengan *Hamiltonian Cycle* diterapkan dalam mencari rute terpendek distribusi air mineral Santri di Pasuruan?
2. Bagaimana implementasi Algoritma Bellman-Ford guna menentukan rute terpendek distribusi air mineral Santri di Pasuruan?

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut.

1. Untuk mengembangkan modifikasi Algoritma Bellman-Ford agar dapat menemukan rute optimal dengan sifat *Hamiltonian Cycle*.
2. Untuk mengimplementasikan Algoritma Bellman-Ford ke dalam model jaringan distribusi.

1.4 Manfaat Penelitian

Melalui penelitian ini, penulis berharap dapat

1. Memberikan wawasan mengenai modifikasi Algoritma Bellman-Ford dalam menentukan rute terpendek distribusi air mineral Santri di Pasuruan.

2. Membantu pihak AMDK Santri dalam menentukan rute distribusi air mineral yang paling efisien di wilayah Pasuruan.

1.5 Batasan Masalah

Penulis membatasi permasalahan pada penelitian ini dengan beberapa poin antara lain.

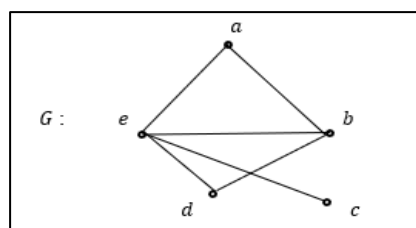
1. Graf yang digunakan adalah graf berbobot tanpa *looping*, di mana setiap bobot pada lintasan merepresentasikan jarak antar titik.
2. Distribusi barang dimulai dari Pabrik Air Mineral Santri dan kembali ke Pabrik Air Mineral Santri.
3. Titik distribusi yang menjadi objek penelitian ini adalah semua Toko Basmalah di Pasuruan.
4. Penelitian ini mengasumsikan bahwa semua rute yang digunakan dalam distribusi dapat diakses.
5. Terdapat modifikasi berupa penambahan *Hamiltonian Cycle* pada Algoritma Bellman-Ford.

BAB II KAJIAN TEORI

2.1 Graf

2.1.1 Definisi Graf

Graf adalah susunan dari dua himpunan terbatas dan tak kosong, yakni himpunan objek yang dinamai simpul (*vertex*) dan himpunan sisi (*edges*). Guna menunjukkan bahwasannya graf G tidak kosong, graf G dinyatakan sebagai $G = (V(G), E(G))$. Himpunan simpul pada graf sering ditulis sebagai $V(G)$, serta himpunan sisi ditulis sebagai $E(G)$, yang mana sisi-sisi tersebut menghubungkan dua simpul u dan v yang dituliskan dengan uv atau vu . Graf sering kali direpresentasikan dengan diagram, dimana titik atau lingkaran kecil sebagai simpul dan segmen garis penghubung sebagai sisi. Jika $uv = e$ adalah sisi G , maka e menghubungkan simpul u dan v , sehingga dapat dikatakan bahwa u dan v saling bertetangga (Chartrand et al., 2016). Sebagai contoh, graf yang memiliki himpunan simpul $V(G) = \{a, b, c, d, e\}$ juga himpunan sisi $E(G) = \{ab, ac, ad, ce, de\}$ yang ditunjukkan pada Gambar 2.1



Gambar 2.1 Graf Sederhana dengan 5 simpul

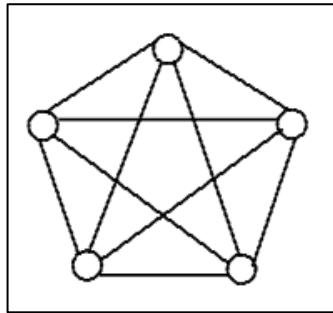
2.1.2 Jenis-Jenis Graf

Graf dapat diklasifikasikan ke dalam berbagai jenis tergantung pada sudut pandang pengelompokannya. Klasifikasi ini bisa dilihat berdasarkan ada atau tidak adanya sisi ganda maupun *loop* (gelang), jumlah simpul, ada atau tidak adanya nilai

(bobot), atau arah orientasi pada sisi (Morihito et al., 2017). Adapun jenis-jenis graf sebagai berikut.

1. Graf Sederhana

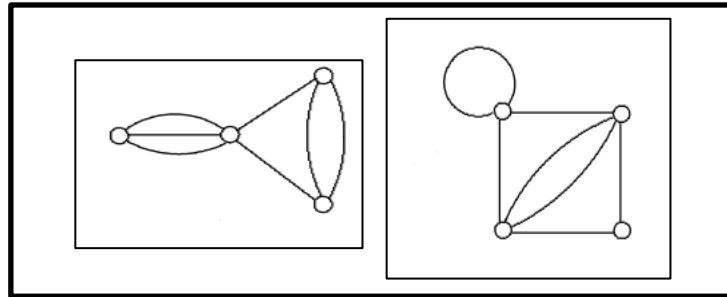
Graf yang tidak memiliki sisi ganda disebut graf sederhana. Sisi dari graf sederhana merupakan pasangan tak-terurut. Sisi (u, v) dapat ditulis dengan (v, u) . Graf sederhana dapat didefinisikan sebagai $G = (V, E)$ yang tersusun atas himpunan tak kosong simpul-simpul serta E merupakan sisi yang terdiri dari himpunan pasangan tak terurut.



Gambar 2.2 Graf Sederhana
Sumber: Chartrand et al. (2016)

2. Graf Tak Sederhana

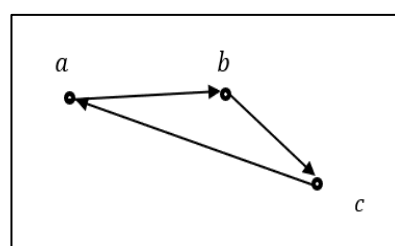
Graf tak sederhana ialah graf yang di dalamnya terdapat sisi ganda. Graf ini terdiri dari dua macam yaitu *multigraph* atau graf ganda dan *pseudograph* atau graf semu. Graf ganda memiliki sisi ganda yang menjadi penghubung dua buah simpul atau lebih. Graf ganda dapat didefinisikan sebagai $G = (V, E)$ yang tersusun atas himpunan ganda yang mengandung sisi ganda. Sedangkan, graf semu ialah graf yang mengandung sisi yang menghubungkan suatu simpul dengan dirinya sendiri.



Gambar 2.3 Graf Ganda dan Graf Semu
Sumber: Chartrand et al. (2016)

3. Graf Berarah

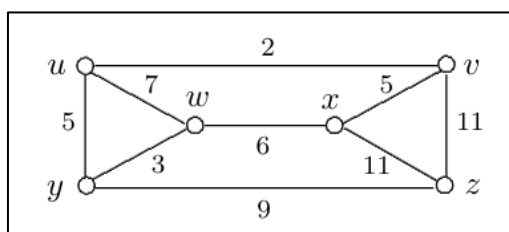
Graf berarah atau Digraf merupakan kumpulan beberapa simpul yang dihubungkan oleh busur. Busur-busur pada graf menunjukkan arah tertentu antar simpul. Contohnya, jika terdapat dua simpul u dan v busur di antaranya dapat dilambangkan dengan $u \rightarrow v$ (arah dari u menuju v). Digraf dapat disimbolkan dengan $V(D)$ atau V untuk simpul, dan $E(D)$ atau E untuk busur. Berikut contoh suatu digraf yang memiliki simpul $V(D) = \{a, b, c, d\}$ dan himpunan busur $E(D) = \{ab, bc, cd, da\}$ menunjukkan arah hubungan berarah antar simpul. Ketika digraf digambarkan dalam bentuk diagram, busur-busur akan ditunjukkan dengan panah yang dapat diartikan sebagai arah hubungan di antara simpul-simpul. Suatu digraf boleh memiliki dua busur yang menghubungkan satu titik yang sama, akan tetapi harus arah yang berlawanan seperti $u \rightarrow v$ dan $v \rightarrow u$ (Chartrand et al., 2016).



Gambar 2.4 Graf Berarah

4. Graf Berbobot

Graf berbobot merupakan graf yang sisi-sisinya diberikan harga (bobot). Bobot untuk semua sisi menunjukkan jarak antar kota, waktu tempuh perjalanan antar kota, juga bisa berupa biaya perjalanan antar kota (Buhaerah et al., 2022). Menurut Chartrand (2016) graf terhubung G , yang mana semua sisi dari graf diberikan nilai bilangan real yang dinamakan biaya atau bobot. Bobot pada sisi-sisi tersebut dinotasikan dengan $w(e)$, dan graf tersebut dinamakan graf berbobot.



Gambar 2.5 Graf Berbobot
Sumber: Chartrand et al. (2016)

2.1.3 Rute Terpendek

Rute terpendek adalah subjek di teori graf yang berfokus terhadap pencarian rute dengan panjang minimal antar simpul dalam suatu graf. Pada graf berbobot, panjang rute diukur melalui jumlah bobot setiap sisi yang dilewati. Beberapa algoritma yang dipakai dalam penyelesaian kasus terkait rute terpendek ialah Algoritma Dijkstra dan Algoritma Bellman-Ford. Algoritma Dijkstra paling efisien untuk menemukan rute terpendek antar simpul dalam graf berbobot non-negatif, sedangkan Algoritma Bellman-Ford dapat menangani graf yang mempunyai bobot negatif serta tetap mampu menemukan lintasan terpendek dengan akurat (Cormen, 2009).

Bobot pada suatu graf umumnya wajib bernilai positif, meskipun pada beberapa kasus bobot negatif bisa terjadi dan digunakan. Rute terpendek dari titik

awal s menuju ke t diartikan sebagai rute dengan bobot minimal yang merupakan rute sederhana (*simple routh*). Salah satu aplikasi general dari graf berarah dan berbobot ialah untuk menemukan rute terpendek (Hasugian, 2015). Terdapat beberapa jenis kasus rute terpendek, yaitu.

1. Rute terpendek antara dua simpul tertentu.
2. Rute terpendek untuk setiap pasangan simpul dalam graf.
3. Rute terpendek dari satu simpul ke semua simpul lainnya.
4. Rute terpendek antara dua simpul yang melibatkan beberapa simpul lainnya.

2.1.4 *Hamiltonian Cycle*

Menurut Chartrand (2016) misalkan G merupakan suatu graf. Suatu lintasan dalam G yang memuat semua simpul dalam G disebut Lintasan Hamilton, sedangkan siklus dalam G yang memuat semua simpul dari G disebut siklus hamilton *Hamiltonian cycle*. Sebuah graf yang memuat siklus hamilton disebut graf hamilton, dengan orde (jumlah simpul) dari graf hamilton setidaknya adalah 3, dan setiap graf hamilton pasti mengandung lintasan hamilton.

Hamiltonian cycle dalam graf G merupakan siklus sederhana yang melintasi setiap titik di G tepat satu kali. Jika graf G memiliki *Hamiltonian cycle*, maka graf tersebut disebut *Hamiltonian*. Sementara itu, G dikatakan *hypo-Hamiltonian* jika graf tersebut tidak *Hamiltonian*, tetapi setiap subgraf yang diperoleh dengan menghapus satu titik dari G , yaitu $G - v$, bersifat *Hamiltonian* (Bondy, 1972).

2.2 Algoritma Bellman-Ford

Algoritma adalah suatu prosedur komputasi yang memperoleh nilai sebagai *input* lalu mengolahnya sehingga menghasilkan *output*. Algoritma merupakan sekumpulan langkah-langkah komputasi yang mengubah *input* menjadi *output*.

Sama halnya dengan resep memasak, algoritma memberikan panduan secara bertahap dalam menyelesaikan persoalan di bidang komputasi (Cormen, 2009).

Algoritma Bellman-Ford diperluas oleh Richard Bellman dan Lester Ford, Jr. Algoritma ini nyaris sama dengan Algoritma Dijkstra, akan tetapi Algoritma Bellman-Ford dapat menangani bobot negatif dalam menemukan lintasan terpendek pada suatu graf berbobot, baik yang memuat *loop* atau tidak, sedangkan algoritma Dijkstra hanya mampu menangani bobot positif. Algoritma Bellman-Ford akan menunjukkan hasil benar apabila tidak terdapat lintasan berbobot negatif yang bisa diakses dari titik pusat dalam graf (Hasugian, 2015).

Algoritma Bellman-Ford menghitung semua jarak yang lebih pendek dari titik v dari titik sumber. Algoritma ini juga menggunakan $d[u]$ sebagai batas atas dengan jarak $d[u, v]$ dari titik sumber ke titik v . Algoritma ini menginisialisasi jarak dari titik sumber ke titik nol dan semua titik lainnya (hingga tak terbatas). Algoritma ini secara progresif melakukan pembaruan jarak pada setiap titik sumber menuju titik v di dalam V , hingga rute *dalil Boolean TRUE* tercapai, yang menyatakan bahwa titik tersebut dapat dicapai dari titik sumber jika grafik tersebut memiliki siklus non-negatif, dan jika tidak, dikatakan sebagai *Boolean FALSE* (Hasugian, 2015). Berikut adalah langkah-langkah kerja dari algoritma Bellman-Ford.

1. Menginisialisasi jarak semua simpul selain simpul asal dengan nilai tak terhingga.
2. Melakukan relaksasi untuk semua sisi dalam graf sebanyak $|V|-1$ kali ($|V|$ adalah jumlah simpul).
3. Memeriksa apakah ada pembaruan setelah iterasi terakhir untuk mendeteksi adanya siklus negatif.

4. Jika tidak ada siklus negatif, maka jarak terpendek dari simpul asal ke setiap simpul tujuan akan diperoleh.

2.3 Distribusi Air Mineral Santri

Pabrik air minum dalam kemasan “Santri” ialah produsen air minum di bawah naungan Koperasi Pondok Pesantren Sidogiri (Kopontren Sidogiri) yang beroperasi di Desa Umbulan, Kabupaten Pasuruan. Sumber air di Desa Umbulan sangat tersohor sebagai salah satu sumber air terbaik di dunia. Mesin pada pabrik ini dapat menghasilkan 9.000 gelas air ukuran 250 ml, 126.000 botol, serta 4500 galon dalam sehari (Fuadah & Fauzi, 2019). Hasil dari pabrik ini disebar ke beberapa toko atau minimarket, salah satunya ialah Toko Basmalah yang juga dinaungi oleh Kopontren Sidogiri. Toko Basmalah ialah toserba (toko serba ada) yang menjual berbagai jenis kebutuhan sehari-hari dengan menjunjung nilai-nilai keislaman dan menerapkan konsep kemandirian. Toko ini sudah tersebar di berbagai wilayah Jawa Timur dengan jumlah gerai mencapai 161, dengan 34 di antaranya berada di Pasuruan Pabrik AMDK Santri perlu mengoptimalkan rute distribusi air mineral yang telah diproduksi ke toko-toko tersebut, sehingga meminimalkan biaya serta waktu tempuh. (Pambudi & Hendratmi, 2020).

Distribusi AMDK Santri dilakukan dengan menggunakan 13 armada yang terdiri dari 2 unit Fuso FE74 Super (Truk besar) dengan Kapasitas $\pm 1000 - 1150$ karton, 4 unit Mitsubishi Fuso Colt Diesel (Truk sedang/medium) dengan kapaitas $\pm 550 - 700$ karton, 1 unit truk hino 500 FL (box besar) dengan kapasitas $\pm 1500 - 1800$ karton, 6 unit Isuzu ELF NLR 55 (mobil box medium) dengan kapasitas $\pm 350 - 450$ karton, yang siap mengirim ke semua Toko Basmalah.

Rute distribusi air mineral Santri hanya berdasarkan toko yang melakukan pemesanan produk serta pengetahuan lapangan tanpa ada pendekatan yang terorganisir pada pemilihan rute. Hal ini mengakibatkan terjadinya penggunaan rute yang tidak optimal. Jangkauan pasar yang luas serta jaringan jalan yang kompleks antara pabrik dengan berbagai cabang Toko Basmalah mengakibatkan diperlukannya suatu metode yang bisa mengidentifikasi rute terpendek dari pabrik ke seluruh cabang Toko Basmalah (SM et al., 2017)

2.4 Kajian Integrasi dengan Al-Quran dan Hadits

Islam adalah agama *rahmatat lil alamin*, yang mengatur hubungan antara pencipta (*khaliq*) dan yang diciptakan (*mahkluk*) serta hubungan antar manusia melalui muamalah. Tujuan muamalah adalah mendapatkan kedamaian serta kesejahteraan di dunia dan akhirat (Munib, 2018). Allah SWT telah memberikan petunjuk yang jelas bagi umat manusia melalui Al-Qur'an, yang menjadi sumber pedoman dalam menjalani kehidupan sesuai dengan syariat Islam. Hadits Nabi Muhammad SAW menjadi pedoman penting setelah Al-Qur'an, hadits Nabi Muhammad SAW dalam menjelaskan dan memperjelas ajaran-ajaran yang ada dalam Al-Qur'an. Hadits, sebagai perkataan, perbuatan, dan persetujuan Nabi, memberikan tuntunan praktis yang sangat berharga dalam menjalankan kehidupan sehari-hari (Khair, 2022).

Islam juga mendukung penggunaan ilmu pengetahuan dan teknologi yang bermanfaat bagi umat di era modern ini. Penerapan teknologi yang dapat diintegrasikan dengan ajaran Islam salah satunya adalah algoritma Bellman-Ford. Algoritma ini digunakan untuk menemukan rute terpendek dalam suatu jaringan, seperti dalam konteks distribusi logistik. Konsep efisiensi yang diusung dalam algoritma Bellman-Ford ini selaras dengan ajaran Islam, yang mengajarkan untuk

tidak berlebih-lebihan dan menggunakan sumber daya dengan bijak, sebagaimana dijelaskan dalam QS. Al-A'raf [7]: 31, yang artinya:

"Makan dan minumlah, tetapi jangan berlebihan. Sesungguhnya Allah tidak menyukai orang yang berlebih-lebihan."

Algoritma Bellman-Ford, yang bekerja dengan cara memperbarui jarak terpendek secara bertahap hingga menemukan solusi optimal, mencerminkan prinsip perencanaan yang bijak dalam Islam. Ajaran ini tercermin dalam QS. An-Nahl [16]: 90, yang artinya:

"Sesungguhnya Allah menyuruh (kamu) berlaku adil dan berbuat kebajikan, memberi kepada kaum kerabat, dan Allah melarang dari perbuatan keji, kemungkaran, dan permusuhan. Dia memberi pengajaran kepadamu agar kamu dapat mengambil pelajaran."

Ayat tersebut mengajarkan manusia berlaku adil dan berbuat kebajikan. , Keadilan dan kebajikan dalam konteks distribusi barang berarti meminimalkan jarak tempuh yang dapat dioptimalkan dengan algoritma Bellman-Ford. Hal ini menunjukkan bahwa adanya Algoritma Bellman-ford dapat memudahkan urusan manusia.

Hadits ke-36 dalam kitab Arbai'in Nawawi menyebutkan bahwa Nabi Muhammad SAW juga memberikan dorongan untuk memudahkan urusan manusia. Dalam hadits riwayat Muslim, Rasulullah bersabda yang artinya:

"Dari Abu Hurairah radhiyallahu 'anhu dia berkata: Rasulullah shallallahu 'alaihi wa sallam bersabda: "Barangsiapa yang menghilangkan satu kesulitan seorang mukmin yang lain dari kesulitannya di dunia, niscaya Allah akan menghilangkan darinya satu kesulitan pada hari kiamat. Barangsiapa yang meringankan orang yang kesusahan (dalam hutangnya), niscaya Allah akan meringankan baginya (urusannya) di dunia dan akhirat. Barangsiapa yang menutupi aib seorang muslim, niscaya Allah akan menutupi aibnya di dunia dan akhirat. Dan Allah akan senantiasa menolong hamba-Nya, selama hamba tersebut mau menolong saudaranya." (HR. Muslim) "

Integrasi Hadist di atas dengan Algoritma Bellman-Ford menunjukkan relevansi nilai-nilai moral dan etika dalam teknologi serta ilmu pengetahuan.

Hadits di atas menjelaskan bahwa barang siapa yang menghilangkan kesulitan orang lain, maka Allah SWT akan menghilangkan kesulitan orang tersebut. Algoritma Bellman-Ford bisa dipandang sebagai suatu usaha dalam mengatasi persoalan distribusi barang. Proses menentukan jalur terpendek dengan algoritma ini bermanfaat untuk meringankan kesulitan terkait waktu dan biaya transportasi, yang selaras dengan prinsip membantu orang lain mencapai tujuan mereka.

2.5 Kajian Penerapan Algoritma Bellman-Ford dengan Teori Pendukung

Implementasi Algoritma Bellman-Ford terhadap teori-teori yang dijadikan pendukung bertujuan untuk memperoleh rute terpendek dalam distribusi air mineral santri ke semua Toko Basmalah di Kota Pasuruan. Rute terpendek yang ditemukan diharapkan dapat membantu mempercepat transportasi dalam proses distribusi air mineral tersebut. Rute yang lebih efisien berdampak pada kepuasan konsumen sebab, produk yang dibutuhkan selalu tersedia dan tidak mengalami keterlambatan.

Penerapan Algoritma Bellman-Ford dengan teori pendukung akan dilaksanakan dengan melewati beberapa tahapan. Tahap pertama adalah mendeskripsikan data. Data yang didapatkan dari lapangan, yakni beberapa titik distribusi yang tersebar di Kota Pasuruan, di mana setiap titik merupakan lokasi Toko Basmalah yang harus dijangkau. Setiap titik distribusi ini dinotasikan untuk mempermudah proses perhitungan, misalnya titik distribusi di Toko Basmalah *a* dideskripsikan sebagai simpul (*vertex*) v_1 , titik distribusi di Toko Basmalah *b* dinotasikan dengan v_2 , dan seterusnya. Rute antar titik ini kemudian diberi bobot sesuai dengan jarak atau waktu tempuh yang dibutuhkan.

Titik distribusi air mineral santri ke Toko Basmalah digambarkan dengan *vertex* yang dinotasikan secara berurutan, misalnya titik distribusi Toko Basmalah

b dideskripsikan dengan *vertex* 2 dan dinotasikan dengan v_2 , dan seterusnya sampai *vertex* distribusi ke- n . Data dideskripsikan dengan menyertakan titik koordinat dari lokasi titik-titik Toko Basmalah di seluruh Pasuruan, untuk menyampaikan gambaran lebih jelas kepada pembaca. Tabel berikut berisi jarak antar *vertex* distribusi juga akan disediakan sebagai dasar penelitian pada tahapan selanjutnya. Deskripsi data ini bertujuan untuk memetakan wilayah-wilayah distribusi air mineral dengan jelas dan terperinci.

Langkah selanjutnya ialah mencari rute terpendek menggunakan algoritma Bellman-Ford. Pencarian rute terpendek ini diawali dengan membuat matriks ketetanggaan dengan mengumpulkan data jarak antar titik yang telah disusun dalam tabel sebelumnya. Jika terdapat titik-titik yang tidak memiliki tetangga langsung atau tidak mempunyai jalur distribusi langsung, maka proses berikutnya ialah menjumlahkan bobot sesuai dengan aturan algoritma Bellman-Ford. Algoritma ini kemudian akan menentukan jalur optimal dengan memperbarui jarak antar titik secara bertahap.

Rute terpendek ditentukan berdasarkan parameter-parameter penting yang digunakan merujuk pada penelitian sebelumnya. Adapun salah satu referensi adalah penelitian karya Paska Marto Hasugian tahun 2015 tentang pencarian rute terpendek dalam pengantaran barang dalam kota. Algoritma Bellman-Ford digunakan dalam kasus ini, prinsip optimasi yang diterapkan tetap memperhatikan jarak dengan tujuan meningkatkan efisiensi distribusi barang, dalam hal ini air mineral santri ke Toko Basmalah di Pasuruan.

BAB III METODE PENELITIAN

3.1 Jenis Penelitian

Penelitian ini menggunakan metode studi literatur dan pendekatan studi lapangan melalui *Google Maps*. Studi literatur dilakukan dengan mengkaji buku-buku dan beberapa artikel jurnal terkait teori graf, Algoritma Bellman-Ford dan rute terpendek. Sedangkan pendekatan studi lapangan dilakukan dengan mencari sumber informasi dan melakukan pengamatan data berdasarkan objek penelitian. Data dari hasil studi lapangan disajikan secara utuh untuk mendapat informasi yang lengkap, lalu diinterpretasikan untuk memperoleh kesimpulan (Darmalaksana, 2020).

3.2 Data dan Sumber Data

Distribusi air mineral dapat dimodelkan sebagai graf berbobot $G = (V(G), E(G), w)$, di mana $V(G)$ adalah himpunan simpul, yang merepresentasikan lokasi-lokasi (pabrik dan toko-toko Basmalah). $E(G)$ adalah himpunan sisi (*edges*), yang merepresentasikan hubungan antar lokasi dengan jalur distribusi yang memungkinkan. $w: E \rightarrow \mathbb{R}^+$ adalah fungsi bobot yang merepresentasikan jarak antar lokasi (dalam kilometer).

Simpul-simpul pada graf berbobot $G = (V(G), E(G), w)$ diinisialisasikan sebagai

1. Pabrik Air Mineral Santri: A .
2. Toko Basmalah di Pasuruan: $v_1, v_2, \dots, v_{28}$ sebagai titik tujuan distribusi.
3. Himpunan simpul: $V(G) = \{A, v_1, v_2, v_3, \dots, v_{29}\}$.

Berdasarkan data yang diperoleh, terdapat setidaknya 28 Toko Basmalah di Pasuruan yang tersebar di beberapa kecamatan dan termuat di graf berbobot

$$G = (V(G), E(G), w) \text{ yaitu } A, v_1, v_2, v_3, \dots, v_{28}$$

Berikut merupakan titik koordinat yang menunjukkan letak dari masing-masing toko Basmalah di Pasuruan.

Tabel 3.1 Titik Koordinat Lokasi Distribusi

Inisial	Nama Tempat	Jarak awal	Titik Koordinat
A	Pabrik AMDK Santri	0	-7.7598876,112.9327587
v_1	Toko Basmalah Sidogiri	∞	-7.6705829,112.8310441
v_2	Toko Basmalah Pusat	∞	-7.6706883,112.8341549
v_3	Toko Basmalah unit 3	∞	-7.6688276,112.8363952
v_4	Toko Basmalah Ngempit	∞	-7.6673642,112.8368768
v_5	Toko Basmalah Rest Area KM 64	∞	-7.6559664,112.8357438
v_6	Toko Basmalah Selotambak	∞	-7.6416277,112.9007693
v_7	Toko Basmalah Gambir Kuning	∞	-7.6996056,112.8377162
v_8	Toko Basmalah Seban	∞	-7.6496078,112.8703608
v_9	Toko Basmalah Kebonsari	∞	-7.6420066,112.9033673
v_{10}	Toko Basmalah Kejayan	∞	-7.6989441,112.8570664
v_{11}	Toko Basmalah Rembang	∞	-7.6397052,112.7829982
v_{12}	Toko Basmalah Wonosari	∞	-7.7129144,112.8969039
v_{13}	Toko Basmalah Karang Sentul	∞	-7.7024837,112.8827123
v_{14}	Toko Basmalah Pandaan	∞	-7.653754,112.6778982
v_{15}	Toko Basmalah Ngemplakrejo	∞	-7.632513,112.8916976
v_{16}	Toko Basmalah Baujeng	∞	-7.6251352,112.7136629
v_{17}	Toko Basmalah Warung Dowo	∞	-7.6900767,112.8622375
v_{18}	Toko Basmalah Karangsono	∞	-7.7324973,112.7059571
v_{19}	Toko Basmalah Purwosari	∞	-7.7690225,112.7287059
v_{20}	Toko Basmalah Oro-oro Ombokulon	∞	-7.6157069,112.7603345
v_{21}	Toko Basmalah Krampyangan	∞	-7.6620254,112.9061699
v_{22}	Toko Basmalah Kebonwaris	∞	-7.6445605,112.6907465
v_{23}	Toko Basmalah Tenggilis Rejo	∞	-7.721524,112.9130406
v_{24}	Toko Basmalah Pasrepan	∞	-7.7744448,112.8592286
v_{25}	Toko Basmalah Plumbon	∞	-7.6478844,112.6787975
v_{26}	Toko Basmalah Wonorejo	∞	-7.7199181,112.7814007
v_{27}	Toko Basmalah Sengonagung	∞	-7.7431726,112.7097226
v_{28}	Toko Basmalah Kebonsari 2	∞	-7.6423855,112.9035835

Berdasarkan data titik koordinat di atas akan dicari jarak antar simpul $w(u, v)$ menggunakan *Google Maps*. Setiap pasangan lokasi (u, v) memiliki bobot $w(u, v)$ yang menyatakan jarak dalam satuan kilometer ($w: E \rightarrow \mathbb{R}^+$), sehingga matriks bobot antar toko dapat dituliskan sebagai matriks ketetanggaan berbobot:

$$W = \begin{bmatrix} 0 & w(A, v_1) & w(A, v_2) & \dots & w(A, v_{28}) \\ w(v_1, A) & 0 & w(v_1, v_2) & \dots & w(v_1, v_{28}) \\ w(v_2, A) & w(v_2, v_1) & 0 & \dots & w(v_2, v_{28}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w(v_{28}, A) & w(v_{28}, v_1) & w(v_{28}, v_2) & \dots & 0 \end{bmatrix}$$

Di mana $W[i][j]$ menyatakan jarak antara lokasi i dan j .

3.3 Tahapan Penelitian

3.3.1 Algoritma Bellman-Ford dengan Siklus *Hamiltonian Cycle*

Proses kerja Algoritma Bellman-Ford dengan menerapkan *Hamiltonian Cycle* dalam menentukan rute terpendek, mengikutsertakan beberapa tahapan. Tahapan selengkapnya adalah sebagai berikut.

1. Inisialisasi

- a. Menentukan titik awal sebagai titik asal beserta beberapa titik tujuan.

$$V(G) = \{ A, v_1, v_2, \dots, v_n \}$$

Dengan A sebagai titik awal dan $v_1, v_2, \dots, v_n$ sebagai titik tujuan.

- b. Mengatur jarak antara titik awal dengan dirinya sendiri sebagai 0.

$$d(A) = 0$$

- c. Mengatur jarak dari titik awal ke setiap titik lain sebagai tak hingga ∞ .

$$d(v_i) = \infty, v_i \in V, i = 1, 3, 4, \dots, 28$$

di mana $d(v_i)$ menyatakan jarak dari titik awal A ke titik v_i .

- d. Menyimpan jarak semua titik ke dalam satu struktur data.

$$D = \{d(A), d(v_1), d(v_2), d(v_3), \dots, d(v_n)\}$$

2. Membuat graf awal dengan *google collab*

- a. *Import Library*

pandas digunakan untuk membaca file *Excel* yang berisi data graf.

networkx digunakan untuk membuat dan mengelola graf.

matplotlib.pyplot digunakan untuk menggambar graf.

b. Membaca Data dari File *Excel*

file_path menyimpan lokasi file *Excel* yang berisi data jarak antar

simpul. *pd.read_excel(file_path)* membaca file *Excel* ke dalam

DataFrame df.

c. Membuat Graf Berarah

nx.DiGraph() digunakan untuk membuat graf berarah (*directedgraph*),

yaitu graf di mana setiap *edge* memiliki arah tertentu.

d. Menambahkan *Edge* ke Graf

Looping melalui setiap baris dalam *DataFrame (df.iterrows())* untuk

membaca nilai-nilai sumber (*source*), tujuan (*target*), dan bobot

(*weight*). *G.add_edge(source, target, weight=weight)* menambahkan

edge ke dalam graf dengan bobot tertentu.

e. Menggambar Graf

nx.spring_layout(G) menentukan tata letak graf agar tampilan lebih

estetis. *nx.draw()* menggambar graf dengan simpul berwarna biru muda

dan ukuran tertentu. *nx.draw_networkx_edge_labels()* menambahkan

label bobot pada setiap *edge*.

f. Menampilkan *Plot* Graf

plt.show() Menampilkan graf yang telah dibuat.

3. Relaksasi

a. Melakukan relaksasi untuk semua sisi yang terdapat di dalam graf

sebanyak $|V| - 1$ kali, yang mana $|V|$ merupakan jumlah titik pada graf.

Hal ini dilakukan sebab dalam graf dengan $|V|$ titik, rute terpendek akan selalu ditemukan setelah maksimal $|V| - 1$ iterasi relaksasi.

- b. Untuk setiap sisi (u, v, w) perlu diperiksa bahwa
 - i. $v \notin V(G)$, di mana $V(G)$ adalah himpunan simpul yang telah dikunjungi.
 - ii. $d[u] + w < d[v]$, di mana $d[v]$ adalah jarak minimum dari titik asal ke v .
 - iii. Tidak kembali ke simpul sebelumnya
 - iv. $v \neq p[u]$, di mana $p[u]$ adalah simpul sebelumnya dari u dalam jalur saat ini.
- c. Jika semua syarat terpenuhi, maka $d[v]$ diperbarui dan tandai v sebagai dikunjungi.

$$d[v] \leftarrow d[u] + w, V(G) \leftarrow V(G) \cup \{v\}$$

4. Setelah iterasi selesai, periksa apakah semua simpul telah dikunjungi. Jika ada simpul yang tidak terhubung, hentikan karena graf tidak memiliki *Hamiltonian Cycle*.
5. Menambahkan iterasi untuk kembali ke simpul awal, dengan memilih jalur dengan bobot minimum dari simpul terakhir ke awal.
6. Memeriksa siklus negatif

Setelah melakukan relaksasi sebanyak $|V| - 1$ kali sekaligus iterasi untuk menuju titik awal, proses selanjutnya ialah melakukan satu iterasi tambahan guna mendeteksi keberadaan siklus negatif. Apabila dimisalkan

- a. $d(u)$ adalah jarak minimum dari titik asal ke simpul u .

- b. $d(v)$ menyatakan jarak minimum (*shortest distance*) dari titik asal (*source node*) ke simpul v setelah menjalankan Algoritma Bellman-Ford.
- c. (u, v) adalah bobot sisi dari simpul u ke simpul v .
- d. $E(G)$ adalah himpunan sisi pada graf.

Dengan demikian, pada semua sisi $(u, v) \in E(G)$, akan diperiksa kondisi

$$d(v) > d(u) + w(u, v)$$

Apabila terdapat pasangan (u, v) yang memenuhi kondisi tersebut, maka terdapat siklus negatif dalam graf.

- 7. Membuat graf akhir dengan langkah yang sama dengan membuat graf awal.

3.3.2 Tahapan Implementasi Algoritma Bellman-Ford

Setelah memahami proses kerja Algoritma Bellman-Ford dengan menerapkan *Hamiltonian Cycle* dalam menentukan rute terpendek, langkah selanjutnya memahami proses implementasi Algoritma Bellman-Ford dalam mencari rute terpendek dari titik asal lalu kembali lagi ke titik asal. Adapun langkah-langkah yang dilakukan adalah sebagai berikut.

- 1. Inisialisasi
 - a. Menentukan titik awal sebagai titik asal beserta beberapa titik tujuan.

$$V(G) = \{ A, v_1, v_2, v_3, \dots, v_n \}$$

Dengan A sebagai titik awal dan $v_1, v_2, v_3, \dots, v_n$ sebagai titik tujuan.

- b. Mengatur jarak antara titik awal dengan dirinya sendiri sebagai 0.

$$d(A) = 0$$

- c. Mengatur jarak dari titik awal ke setiap titik lain sebagai tak hingga ∞ .

$$d(v_i) = \infty, \forall v_i \in A, i = 1, 2, 3, \dots, 28.$$

di mana $d(v_i)$ menyatakan jarak dari titik awal A ke titik v_i .

- d. Menyimpan jarak semua titik ke dalam satu struktur data.

$$D = \{d(A), d(v_1), d(v_2), d(v_3), \dots, d(v_n)\}$$

2. Relaksasi

Melakukan relaksasi untuk semua sisi yang terdapat di dalam graf sebanyak $|V| - 1$ kali, yang mana $|V|$ merupakan jumlah titik pada graf. Hal ini dilakukan sebab dalam graf dengan $|V|$ titik, rute terpendek akan selalu ditemukan setelah maksimal $|V| - 1$ iterasi relaksasi.

3. Memeriksa siklus negatif

Setelah melakukan relaksasi sebanyak $|V| - 1$ kali, proses selanjutnya ialah melakukan satu iterasi tambahan guna mendeteksi keberadaan siklus negatif. Apabila dimisalkan

- a. $d(u)$ adalah jarak minimum dari titik asal ke simpul u .
- b. $d(v)$ menyatakan jarak minimum (*shortest distance*) dari titik asal (*source node*) ke simpul v setelah menjalankan algoritma Bellman-Ford.
- c. (u, v) adalah bobot sisi dari simpul u ke simpul v .
- d. $E(G)$ adalah himpunan sisi pada graf.

Dengan demikian, pada semua sisi $(u, v) \in E(G)$, akan diperiksa kondisi

$$d(v) > d(u) + w(u, v)$$

Apabila terdapat pasangan (u, v) yang memenuhi kondisi tersebut, maka terdapat siklus negatif dalam graf.

4. Membuat graf akhir dengan langkah yang sama dengan membuat graf awal.

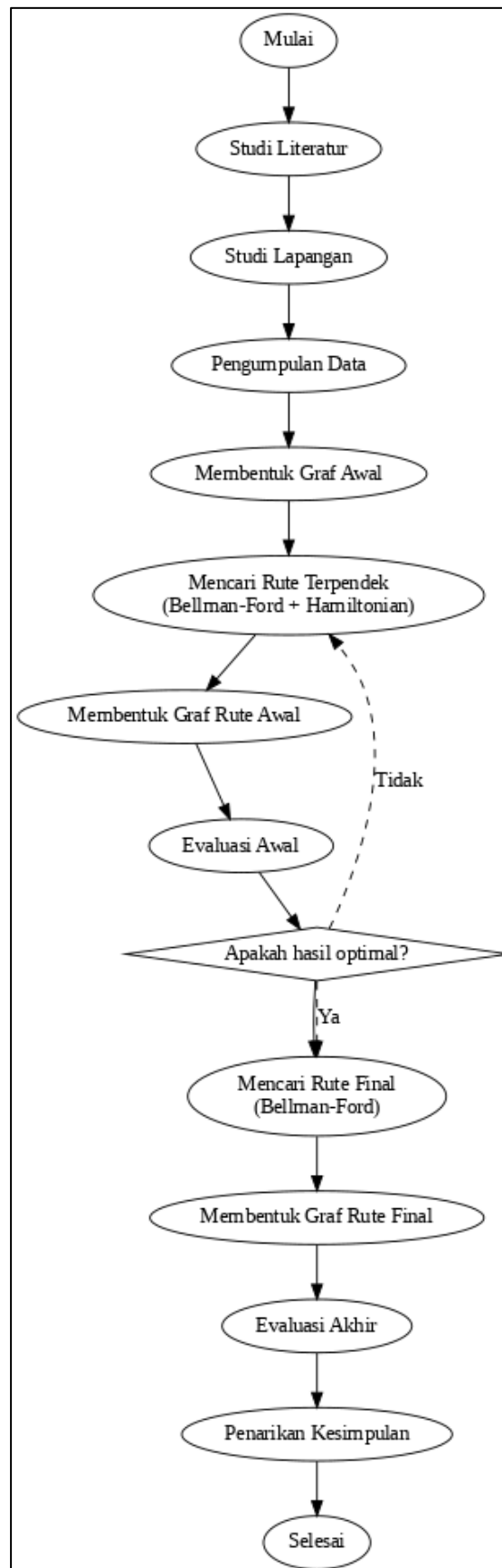
3.3.3 Tahapan Evaluasi Penelitian

Setelah mengetahui hasil dari penerapan siklus *hamiltonian cycle* pada Algoritma Bellman-Ford serta implementasi Algoritma Bellman-Ford maka langkah berikutnya ialah mengevaluasi algoritma tersebut guna mengetahui efektivitas penerapan siklus *hamiltonian cycle* pada Algoritma Bellman-Ford. Adapun langkah-langkahnya sebagai berikut.

1. Memvalidasi hasil rute.
2. Menganalisis performa dan efisiensi.
3. Mengidentifikasi kelemahan dan kendala.
4. Mencari potensi perbaikan.
5. Penarikan kesimpulan.

3.3.4 *Flowchart* Tahapan Penelitian

Berikut merupakan *flowchart* yang menggambarkan alur penelitian dari penerapan *Hamiltonian Cycle* pada Algoritma Bellman-Ford dan implementasi Algoritma Bellman-Ford pada penentuan rute terpendek.



Gambar 3.1 *Flowchart* Tahapan Penelitian

BAB IV HASIL DAN PEMBAHASAN

4.1 Algoritma Bellman-Ford dengan Siklus *Hamiltonian Cycle* guna Menentukan Rute Terpendek Distribusi Air Mineral Santri di Pasuruan

Distribusi air mineral santri dimulai dari pabrik AMDK Santri yang terletak di Desa Umbulan, Kecamatan Winongan, Kabupaten Pasuruan menuju semua toko Basmalah di Kota Pasuruan, dan kembali titik awal yakni pabrik air minum dalam kemasan “Santri”. Berikut adalah jarak antar titik distribusi air mineral santri di Pasuruan.

Tabel 4.1 Data ke 1, Jarak Sebelum Penambahan Bobot Negatif

	A	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
A	0	14,97	14,71	14,67	14,75	15,74	13,61	12,43	14,06	13,5	10,75
v_1	14,97	0	0,34	0,62	0,74	1,71	8,33	3,31	4,92	8,58	4,26
v_2	14,71	0,34	0	0,32	0,48	1,65	8,02	3,24	4,63	8,27	4,03
v_3	14,67	0,62	0,32	0	0,17	1,43	7,71	3,43	4,31	7,96	4,05
v_4	14,75	0,74	0,48	0,17	0	1,27	7,60	3,59	4,19	7,85	4,16
v_5	15,74	1,71	1,65	1,43	1,27	0	7,34	4,86	3,88	7,61	5,33
v_6	13,61	8,33	8,02	7,71	7,60	7,34	0	9,48	3,47	0,29	7,99
v_7	12,43	3,31	3,24	3,43	3,59	4,86	9,48	0	6,62	9,66	2,13
v_8	14,06	4,92	4,63	4,31	4,19	3,88	3,47	6,62	0	3,73	5,68
v_9	13,5	8,58	8,27	7,96	7,85	7,61	0,29	9,66	3,73	0	8,13
v_{10}	10,75	4,26	4,03	4,05	4,16	5,33	7,99	2,13	5,68	8,13	0
v_{11}	21,23	6,31	6,61	6,72	6,69	6,09	12,98	8,98	9,69	13,27	10,49
v_{12}	6,55	8,65	8,36	8,28	8,33	9,25	7,94	6,69	7,62	7,92	4,66
v_{13}	8,44	6,71	6,41	6,33	6,38	7,32	7,05	4,97	6,04	7,10	2,85
v_{14}	30,46	16,98	17,32	17,55	17,59	17,4	24,6	18,33	21,22	24,88	20,37
v_{15}	14,87	7,91	7,63	7,31	7,18	6,70	1,42	9,54	3,02	1,66	8,31
v_{16}	28,41	13,89	14,21	14,37	14,37	13,88	20,7	15,98	17,48	20,99	17,81
v_{17}	10,98	4,06	3,77	3,70	3,77	4,79	6,86	2,90	4,59	7,01	1,14
v_{18}	25,17	15,41	15,71	16,02	16,14	16,64	23,73	14,97	20,33	23,97	17,06
v_{19}	22,50	15,72	15,96	16,28	16,43	17,24	23,67	14,28	20,49	23,87	16,15
v_{20}	24,86	9,90	10,18	10,25	10,21	9,44	15,74	12,64	12,70	16,03	14,12
v_{21}	11,27	8,33	7,99	7,73	7,66	7,79	2,34	8,62	4,18	2,25	6,79
v_{22}	29,59	15,73	16,07	16,28	16,30	16,03	23,15	17,31	19,80	23,43	19,30
v_{23}	4,79	10,66	10,37	10,28	10,33	11,21	8,99	8,65	9,28	8,91	6,66
v_{24}	8,26	11,96	11,86	12,01	12,16	13,43	15,46	8,65	13,94	15,51	8,40
v_{25}	30,63	16,97	17,31	17,52	17,56	17,32	24,47	18,43	21,11	24,76	20,45
v_{26}	17,26	7,75	7,98	8,31	8,46	9,30	15,77	6,60	12,54	15,99	8,66
v_{27}	24,64	15,62	15,90	16,22	16,35	16,94	23,89	14,91	20,53	24,12	16,96
v_{28}	13,46	8,59	8,27	7,97	7,86	7,63	0,32	9,65	3,75	0,05	8,11

Tabel 4.2 Data ke 2, Jarak Sebelum Penambahan Bobot Negatif

	v_{11}	v_{12}	v_{13}	v_{14}	v_{15}	v_{16}	v_{17}	v_{18}	v_{19}	v_{20}	v_{21}
A	21,23	6,55	8,44	30,46	14,87	28,41	10,98	25,17	22,5	24,86	11,27
v_1	6,31	8,65	6,71	16,98	7,91	13,89	4,06	15,41	15,72	9,90	8,33
v_2	6,61	8,36	6,41	17,32	7,63	14,21	3,77	15,71	15,96	10,18	7,99
v_3	6,72	8,28	6,33	17,55	7,31	14,37	3,70	16,02	16,28	10,25	7,73
v_4	6,69	8,33	6,38	17,59	7,18	14,37	3,77	16,14	16,43	10,21	7,66
v_5	6,09	9,25	7,32	17,40	6,70	13,88	4,79	16,64	17,24	9,44	7,79
v_6	12,98	7,94	7,05	24,60	1,42	20,70	6,86	23,73	23,67	15,74	2,34
v_7	8,98	6,69	4,97	18,33	9,54	15,98	2,90	14,97	14,28	12,64	8,62
v_8	9,69	7,62	6,04	21,22	3,02	17,48	4,59	20,33	20,49	12,70	4,18
v_9	13,27	7,92	7,10	24,88	1,66	20,99	7,01	23,97	23,87	16,03	2,25
v_{10}	10,49	4,66	2,85	20,37	8,31	17,81	1,14	17,06	16,15	14,12	6,79
v_{11}	0	14,96	13,02	11,69	12,01	7,81	10,37	13,36	15,57	3,66	13,8
v_{12}	14,96	0	1,95	25,01	8,96	22,43	4,59	21,15	19,55	18,53	5,75
v_{13}	13,02	1,95	0	23,21	7,84	20,52	2,64	19,76	18,51	16,58	5,19
v_{14}	11,69	25,01	23,21	0	23,68	5,07	20,71	9,29	13,99	10,02	25,17
v_{15}	12,01	8,96	7,84	23,68	0	19,64	7,18	23,29	23,52	14,6	3,65
v_{16}	7,81	22,43	20,52	5,07	19,64	0	17,90	11,97	16,09	5,25	21,61
v_{17}	10,37	4,59	2,64	20,71	7,18	17,90	0	17,85	17,13	13,95	5,76
v_{18}	13,36	21,15	19,76	9,29	23,29	11,97	17,85	0	4,77	14,3	23,41
v_{19}	15,57	19,55	18,51	13,99	23,52	16,09	17,13	4,77	0	17,40	22,89
v_{20}	3,66	18,53	16,58	10,02	14,60	5,25	13,95	14,3	17,40	0	16,88
v_{21}	13,8	5,75	5,19	25,17	3,65	21,61	5,76	23,41	22,89	16,88	0
v_{22}	10,18	23,96	22,11	1,75	22,19	3,32	19,56	9,92	14,46	8,31	23,82
v_{23}	16,97	2,02	3,96	26,99	10,17	24,45	6,60	22,85	20,99	20,53	6,66
v_{24}	17,18	8,00	8,41	24,07	16,18	23,09	9,39	17,52	14,39	20,74	13,53
v_{25}	11,52	25,1	23,28	0,66	23,53	4,60	20,75	9,87	14,55	9,67	25,11
v_{26}	8,92	12,75	11,33	13,57	15,56	12,92	9,51	8,43	7,97	11,82	15,18
v_{27}	14,06	20,90	19,59	10,54	23,53	13,13	17,81	1,26	3,55	15,23	23,45
v_{28}	13,29	7,88	7,07	24,90	1,71	21,02	6,99	23,97	23,87	16,06	2,20

Tabel 4.3 Data Ke 3, Jarak Sebelum Penambahan Bobot Negatif

	v_{22}	v_{23}	v_{24}	v_{25}	v_{26}	v_{27}	v_{28}
A	29,59	4,79	8,26	30,63	17,26	24,64	13,46
v_1	15,73	10,66	11,96	16,97	7,75	15,62	8,59
v_2	16,07	10,37	11,86	17,31	7,98	15,90	8,27
v_3	16,28	10,28	12,01	17,52	8,31	16,22	7,97
v_4	16,30	10,33	12,16	17,56	8,46	16,35	7,86
v_5	16,03	11,21	13,43	17,32	9,30	16,94	7,63
v_6	23,15	8,99	15,46	24,47	15,77	23,89	0,32
v_7	17,31	8,65	8,65	18,43	6,60	14,91	9,65
v_8	19,80	9,28	13,94	21,11	12,54	20,53	3,75
v_9	23,43	8,91	15,51	24,76	15,99	24,12	0,05
v_{10}	19,3	6,66	8,40	20,45	8,66	16,96	8,11
v_{11}	10,18	16,97	17,18	11,52	8,92	14,06	13,29
v_{12}	23,96	2,02	8,00	25,10	12,75	20,90	7,88
v_{13}	22,11	3,96	8,41	23,28	11,33	19,59	7,07
v_{14}	1,75	26,99	24,07	0,66	13,57	10,54	24,9
v_{15}	22,19	10,17	16,18	23,53	15,56	23,53	1,71
v_{16}	3,32	24,45	23,09	4,60	12,92	13,13	21,02
v_{17}	19,56	6,60	9,39	20,75	9,51	17,81	6,99
v_{18}	9,92	22,85	17,52	9,87	8,43	1,26	23,97
v_{19}	14,46	20,99	14,39	14,55	7,97	3,55	23,87
v_{20}	8,31	20,53	20,74	9,67	11,82	15,23	16,06
v_{21}	23,82	6,66	13,53	25,11	15,18	23,45	2,20
v_{22}	0	25,95	23,52	1,37	13,04	11,16	23,46
v_{23}	25,95	0	8,35	27,08	14,51	22,53	8,86
v_{24}	23,52	8,35	0	24,36	10,50	16,84	15,48

Tabel 4.4 Data ke 2, Jarak Sebelum Penambahan Bobot Negatif

	v_{22}	v_{23}	v_{24}	v_{25}	v_{26}	v_{27}	v_{28}
v_{25}	1,37	27,08	24,36	0	13,86	11,13	24,78
v_{26}	13,04	14,51	10,50	13,86	0	8,31	15,99
v_{27}	11,16	22,53	16,84	11,13	8,31	0	24,12
v_{28}	23,46	8,86	15,48	24,78	15,99	24,12	0

Bobot asli berdasarkan jarak antar titik sudah diketahui, setelahnya akan dilakukan pengurangan bobot untuk beberapa titik tertentu yang memiliki tingkat prioritas lebih tinggi. Pengurangan ini bertujuan untuk mengutamakan titik-titik tersebut tersebut lebih awal tanpa mengabaikan keseluruhan efisiensi rute. Pengurangan bobot diberikan lebih besar kepada titik-titik dengan potensi distribusi, eksposur, dan pusat aktivitas yang lebih tinggi. Nilai pengurangan ditetapkan mulai dari -0,5 hingga -3, di mana semakin besar nilai pengurangan (lebih negatif), semakin tinggi pula prioritas titik tersebut dalam mendukung distribusi.

Adapun rincian pengurangan bobot dapat dilihat pada tabel berikut:

Tabel 4.5 Besar Pengurangan Bobot pada Titik Tertentu

Titik Prioritas	Besar Pengurangan Bobot	Alasan Prioritas
v_1	-0,5	Pusat jaringan Basmalah, dekat dengan Pondok Pesantren Sidogiri.
v_5	-1,5	Rest area jalan tol, lokasi strategis untuk eksposur dan penjualan cepat.
v_{10}	-2,5	Pusat aktivitas ekonomi Kejayan, simpul distribusi lokal.
v_{12}	-2	Wilayah transit Pandaan, dekat <i>exit</i> tol dan jalur utama ke arah Batu–Malang.
v_{14}	-3	Wilayah padat penduduk, memperluas distribusi ke arah timur dari Sidogiri.

Berikut adalah data jarak antar lokasi distribusi dengan beberapa titik yang telah ditambahkan dengan bobot negatif sesuai dengan tabel 4.5.

Tabel 4.6 Data ke 1, Jarak Setelah Penambahan Bobot Negatif

	A	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
A	0	14,47	14,71	14,67	14,75	14,24	13,61	12,43	14,06	13,5	8,25
v_1	14,47	0	-0,16	0,12	0,24	0,21	7,83	2,81	4,42	8,08	1,76
v_2	14,71	-0,16	0	0,32	0,48	0,15	8,02	3,24	4,63	8,27	1,53
v_3	14,67	0,12	0,32	0	0,17	-0,07	7,71	3,43	4,31	7,96	1,55
v_4	14,75	0,24	0,48	0,17	0	-0,23	7,60	3,59	4,19	7,85	1,66
v_5	14,24	0,21	0,15	-0,07	-0,23	0	5,84	3,36	2,38	6,11	2,83
v_6	13,61	7,83	8,02	7,71	7,60	5,84	0	9,48	3,47	0,29	5,49
v_7	12,43	2,81	3,24	3,43	3,59	3,36	9,48	0	6,62	9,66	-0,37
v_8	14,06	4,42	4,63	4,31	4,19	2,38	3,47	6,62	0	3,73	3,18
v_9	13,5	8,08	8,27	7,96	7,85	6,11	0,29	9,66	3,73	0	5,63
v_{10}	8,25	1,76	1,53	1,55	1,66	2,83	5,49	-0,37	3,18	5,63	0
v_{11}	21,23	5,81	6,61	6,72	6,69	4,59	12,98	8,98	9,69	13,27	7,99
v_{12}	4,55	6,65	6,36	6,28	6,33	7,25	5,94	4,69	5,62	5,92	2,66
v_{13}	8,44	6,21	6,41	6,33	6,38	5,82	7,05	4,97	6,04	7,10	0,35
v_{14}	27,46	13,98	14,32	14,55	14,59	14,40	21,60	15,33	18,22	21,88	17,37
v_{15}	14,87	7,41	7,63	7,31	7,18	5,20	1,42	9,54	3,02	1,66	5,81
v_{16}	28,41	13,39	14,21	14,37	14,37	12,38	20,7	15,98	17,48	20,99	15,31
v_{17}	10,98	3,56	3,77	3,70	3,77	3,29	6,86	2,90	4,59	7,01	-1,36
v_{18}	25,17	14,91	15,71	16,02	16,14	15,14	23,73	14,97	20,33	23,97	14,56
v_{19}	22,50	15,22	15,96	16,28	16,43	15,74	23,67	14,28	20,49	23,87	13,65
v_{20}	24,86	9,40	10,18	10,25	10,21	7,94	15,74	12,64	12,70	16,03	11,62
v_{21}	11,27	7,83	7,99	7,73	7,66	6,29	2,34	8,62	4,18	2,25	4,29
v_{22}	29,59	15,23	16,07	16,28	16,30	14,53	23,15	17,31	19,80	23,43	16,8
v_{23}	4,79	10,16	10,37	10,28	10,33	9,71	8,99	8,65	9,28	8,91	4,16
v_{24}	8,26	11,46	11,86	12,01	12,16	11,93	15,46	8,65	13,94	15,51	5,90
v_{25}	30,63	16,47	17,31	17,52	17,56	15,82	24,47	18,43	21,11	24,76	17,95
v_{26}	17,26	7,25	7,98	8,31	8,46	7,80	15,77	6,60	12,54	15,99	6,16
v_{27}	24,64	15,12	15,9	16,22	16,35	15,44	23,89	14,91	20,53	24,12	14,46
v_{28}	13,46	8,09	8,27	7,97	7,86	6,13	0,32	9,65	3,75	0,05	5,61

Tabel 4.7 Data ke 2, Jarak Setelah Penambahan Bobot Negatif

	v_{11}	v_{12}	v_{13}	v_{14}	v_{15}	v_{16}	v_{17}	v_{18}	v_{19}	v_{20}	v_{21}
A	21,23	4,55	8,44	27,46	14,87	28,41	10,98	25,17	22,50	24,86	11,27
v_1	5,81	6,65	6,21	13,98	7,41	13,39	3,56	14,91	15,22	9,40	7,83
v_2	6,61	6,36	6,41	14,32	7,63	14,21	3,77	15,71	15,96	10,18	7,99
v_3	6,72	6,28	6,33	14,55	7,31	14,37	3,70	16,02	16,28	10,25	7,73
v_4	6,69	6,33	6,38	14,59	7,18	14,37	3,77	16,14	16,43	10,21	7,66
v_5	4,59	7,25	5,82	14,40	5,20	12,38	3,29	15,14	15,74	7,94	6,29
v_6	12,98	5,94	7,05	21,60	1,42	20,70	6,86	23,73	23,67	15,74	2,34
v_7	8,98	4,69	4,97	15,33	9,54	15,98	2,90	14,97	14,28	12,64	8,62
v_8	9,69	5,62	6,04	18,22	3,02	17,48	4,59	20,33	20,49	12,70	4,18
v_9	13,27	5,92	7,10	21,88	1,66	20,99	7,01	23,97	23,87	16,03	2,25
v_{10}	7,99	2,66	0,35	17,37	5,81	15,31	-1,36	14,56	13,65	11,62	4,29
v_{11}	0	12,96	13,02	8,69	12,01	7,81	10,37	13,36	15,57	3,66	13,8
v_{12}	12,96	0	-0,05	22,01	6,96	20,43	2,59	19,15	17,55	16,53	3,75
v_{13}	13,02	-0,05	0	20,21	7,84	20,52	2,64	19,76	18,51	16,58	5,19
v_{14}	8,69	22,01	20,21	0	20,68	2,07	17,71	6,29	10,99	7,02	22,17
v_{15}	12,01	6,96	7,84	20,68	0	19,64	7,18	23,29	23,52	14,60	3,65
v_{16}	7,81	20,43	20,52	2,07	19,64	0	17,90	11,97	16,09	5,25	21,61
v_{17}	10,37	2,59	2,64	17,71	7,18	17,90	0	17,85	17,13	13,95	5,76
v_{18}	13,36	19,15	19,76	6,29	23,29	11,97	17,85	0	4,77	14,30	23,41
v_{19}	15,57	17,55	18,51	10,99	23,52	16,09	17,13	4,77	0	17,40	22,89

Tabel 4.8 Data ke 3, Jarak Setelah Penambahan Bobot Negatif

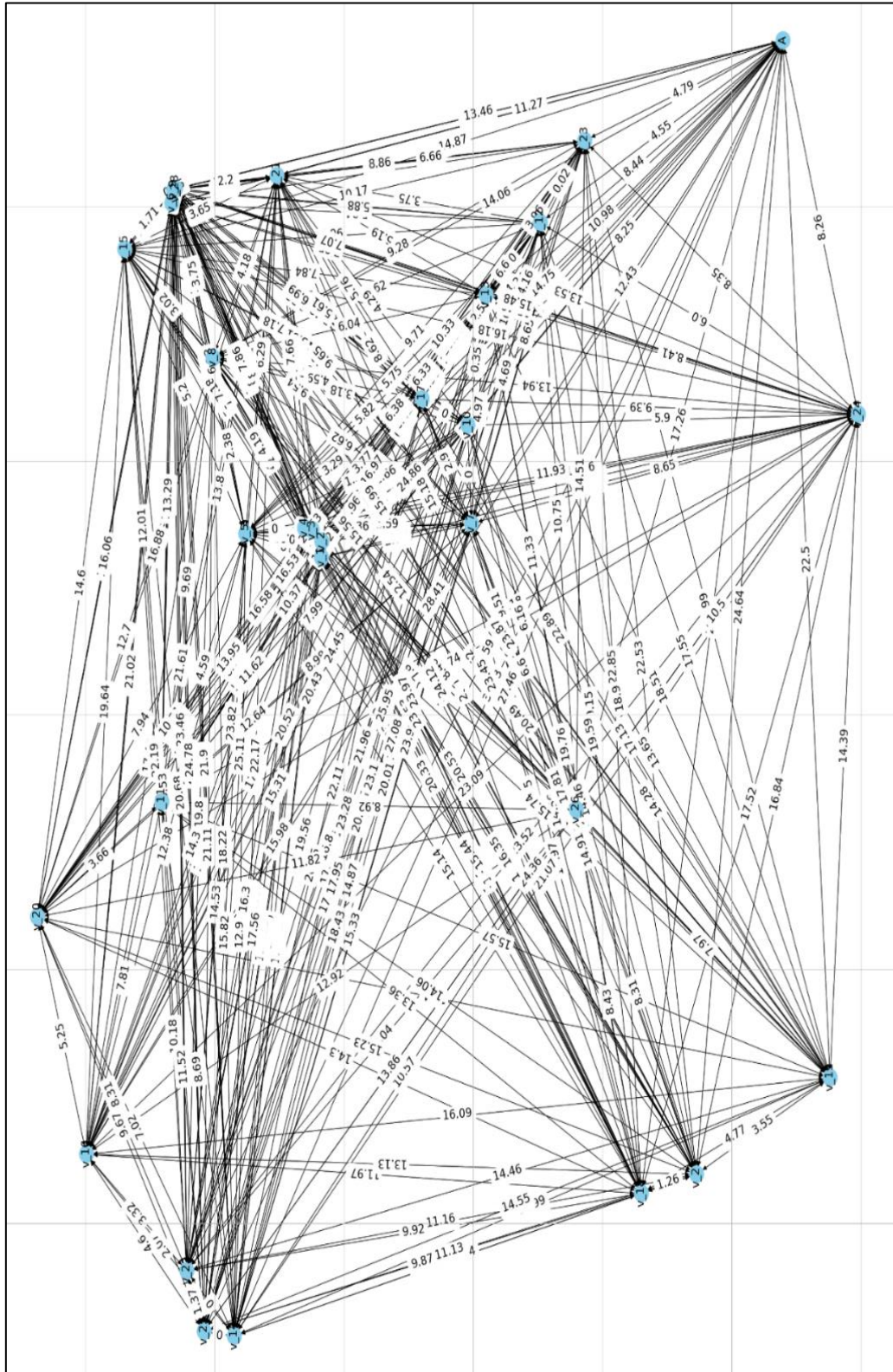
	v_{11}	v_{12}	v_{13}	v_{14}	v_{15}	v_{16}	v_{17}	v_{18}	v_{19}	v_{20}	v_{21}
v_{18}	13,36	19,15	19,76	6,29	23,29	11,97	17,85	0	4,77	14,30	23,41
v_{19}	15,57	17,55	18,51	10,99	23,52	16,09	17,13	4,77	0	17,40	22,89
v_{20}	3,66	16,53	16,58	7,02	14,60	5,25	13,95	14,30	17,40	0	16,88
v_{21}	13,80	3,75	5,19	22,17	3,65	21,61	5,76	23,41	22,89	16,88	0
v_{22}	10,18	21,96	22,11	-1,25	22,19	3,32	19,56	9,92	14,46	8,31	23,82
v_{23}	16,97	0,02	3,96	23,99	10,17	24,45	6,60	22,85	20,99	20,53	6,66
v_{24}	17,18	6,00	8,41	21,07	16,18	23,09	9,39	17,52	14,39	20,74	13,53
v_{25}	11,52	23,1	23,28	-2,34	23,53	4,60	20,75	9,87	14,55	9,67	25,11
v_{26}	8,92	10,75	11,33	10,57	15,56	12,92	9,51	8,43	7,97	11,82	15,18
v_{27}	14,06	18,90	19,59	7,54	23,53	13,13	17,81	1,26	3,55	15,23	23,45
v_{28}	13,29	5,88	7,07	21,90	1,71	21,02	6,99	23,97	23,87	16,06	2,20

Tabel 4.9 Data ke 4, Jarak Setelah Penambahan Bobot Negatif

	v_{22}	v_{23}	v_{24}	v_{25}	v_{26}	v_{27}	v_{28}
A	29,59	4,79	8,26	30,63	17,26	24,64	13,46
v_1	15,23	10,16	11,46	16,47	7,25	15,12	8,09
v_2	16,07	10,37	11,86	17,31	7,98	15,90	8,27
v_3	16,28	10,28	12,01	17,52	8,31	16,22	7,97
v_4	16,30	10,33	12,16	17,56	8,46	16,35	7,86
v_5	14,53	9,71	11,93	15,82	7,80	15,44	6,13
v_6	23,15	8,99	15,46	24,47	15,77	23,89	0,32
v_7	17,31	8,65	8,65	18,43	6,60	14,91	9,65
v_8	19,80	9,28	13,94	21,11	12,54	20,53	3,75
v_9	23,43	8,91	15,51	24,76	15,99	24,12	0,05
v_{10}	16,80	4,16	5,90	17,95	6,16	14,46	5,61
v_{11}	10,18	16,97	17,18	11,52	8,92	14,06	13,29
v_{12}	21,96	0,02	6,00	23,10	10,75	18,90	5,88
v_{13}	22,11	3,96	8,41	23,28	11,33	19,59	7,07
v_{14}	-1,25	23,99	21,07	-2,34	10,57	7,54	21,9
v_{15}	22,19	10,17	16,18	23,53	15,56	23,53	1,71
v_{16}	3,32	24,45	23,09	4,60	12,92	13,13	21,02
v_{17}	19,56	6,60	9,39	20,75	9,51	17,81	6,99
v_{18}	9,92	22,85	17,52	9,87	8,43	1,26	23,97
v_{19}	14,46	20,99	14,39	14,55	7,97	3,55	23,87
v_{20}	8,31	20,53	20,74	9,67	11,82	15,23	16,06
v_{21}	23,82	6,66	13,53	25,11	15,18	23,45	2,20
v_{22}	0	25,95	23,52	1,37	13,04	11,16	23,46
v_{23}	25,95	0	8,35	27,08	14,51	22,53	8,86
v_{24}	23,52	8,35	0	24,36	10,50	16,84	15,48
v_{25}	1,37	27,08	24,36	0	13,86	11,13	24,78
v_{26}	13,04	14,51	10,50	13,86	0	8,31	15,99
v_{27}	11,16	22,53	16,84	11,13	8,31	0	24,12
v_{28}	23,46	8,86	15,48	24,78	15,99	24,12	0

Adapun tahapan-tahapan yang dilakukan terkait pengolahan data guna menyelesaikan rute terpendek pada distribusi air mineral santri yakni:

1. Membuat graf berarah dan berbobot yang akan ditunjukkan pada gambar 4.1, yang mana bobot pada graf berupa jarak antar simpul yang sudah ditambah dengan bobot negatif pada beberapa titik prioritas, dengan satuan kilometer yang diketahui berdasarkan data tabel 4.6 sampai 4.9.
2. Menentukan nilai optimal untuk rute terpendek menggunakan Algoritma Bellman-Ford yang dimodifikasi dengan sifat *hamiltonian cycle*. Nilai optimal diperoleh dengan melakukan semua proses kerja Algoritma Bellman-Ford, seperti inisialisasi dan iterasi dan memeriksa siklus negatif pada hasil akhir. Proses inisialisasi terdapat pada Tabel 3.1. Tahapan iterasi dilalui guna mendapatkan titik terdekat dari titik awal, melewati semua sampai kembali ke titik awal lagi, dengan catatan masing-masing titik hanya boleh dilewati sekali, dan memastikan bahwa semua titik sudah dilewati tanpa ada siklus negatif.



Gambar 4.1 Graf yang menghubungkan semua toko di Pasuruan, dengan A adalah pabrik. Bobot menunjukkan ukuran jarak dalam satuan (km), graf ini tidak memuat *loop*.

Berikut iterasi yang perlu dilalui untuk memperoleh nilai optimal.

a. Iterasi 1

Iterasi 1 dimulai dari A (Titik awal) menuju titik terdekat.

Tabel 4.10 Iterasi 1

Rute	Perhitungan jarak minimum
$A - v_1$	$\min(\infty, 0 + 14,47) = 14,47 \text{ km}$
$A - v_2$	$\min(\infty, 0 + 14,71) = 14,71 \text{ km}$
$A - v_3$	$\min(\infty, 0 + 14,67) = 14,67 \text{ km}$
$A - v_4$	$\min(\infty, 0 + 14,75) = 14,75 \text{ km}$
$A - v_5$	$\min(\infty, 0 + 14,24) = 14,24 \text{ km}$
$A - v_6$	$\min(\infty, 0 + 13,61) = 13,61 \text{ km}$
$A - v_7$	$\min(\infty, 0 + 12,43) = 12,43 \text{ km}$
$A - v_8$	$\min(\infty, 0 + 14,06) = 14,06 \text{ km}$
$A - v_9$	$\min(\infty, 0 + 13,5) = 13,5 \text{ km}$
$A - v_{10}$	$\min(\infty, 0 + 8,25) = 8,25 \text{ km}$
$A - v_{11}$	$\min(\infty, 0 + 21,23) = 21,23 \text{ km}$
$A - v_{12}$	$\min(\infty, 0 + 4,55) = 4,55 \text{ km}$
$A - v_{13}$	$\min(\infty, 0 + 8,44) = 8,44 \text{ km}$
$A - v_{14}$	$\min(\infty, 0 + 27,46) = 27,46 \text{ km}$
$A - v_{15}$	$\min(\infty, 0 + 14,87) = 14,87 \text{ km}$
$A - v_{16}$	$\min(\infty, 0 + 28,41) = 28,41 \text{ km}$
$A - v_{17}$	$\min(\infty, 0 + 10,98) = 10,98 \text{ km}$
$A - v_{18}$	$\min(\infty, 0 + 25,17) = 25,17 \text{ km}$
$A - v_{19}$	$\min(\infty, 0 + 22,5) = 22,5 \text{ km}$
$A - v_{20}$	$\min(\infty, 0 + 24,86) = 24,86 \text{ km}$
$A - v_{21}$	$\min(\infty, 0 + 11,27) = 11,27 \text{ km}$
$A - v_{22}$	$\min(\infty, 0 + 29,59) = 29,59 \text{ km}$
$A - v_{23}$	$\min(\infty, 0 + 4,79) = 4,79 \text{ km}$
$A - v_{24}$	$\min(\infty, 0 + 8,26) = 8,26 \text{ km}$
$A - v_{25}$	$\min(\infty, 0 + 30,63) = 30,63 \text{ km}$
$A - v_{26}$	$\min(\infty, 0 + 17,26) = 17,26 \text{ km}$
$A - v_{27}$	$\min(\infty, 0 + 24,64) = 24,64 \text{ km}$
$A - v_{28}$	$\min(\infty, 0 + 13,46) = 13,46 \text{ km}$

Berdasarkan tabel di atas, diketahui bahwa titik v_{12} merupakan titik terdekat dari A , maka v_{12} ditandai agar tidak dilalui pada iterasi berikutnya.

b. Iterasi 2

Dari tabel 4.10 diketahui bahwasanya titik terdekat dari A adalah titik v_{12} maka langkah berikutnya mencari titik terdekat dari v_{12} . Iterasi

kedua ini menambahkan jarak A ke v_{12} dengan jarak v_{12} ke titik lainnya. Kemudian menentukan titik terdekat dengan v_{12} .

Tabel 4.11 Iterasi 2

Rute	Perhitungan jarak minimum
$v_{12} - v_1$	$\min(\infty, 4,55 + 6,65) = 11,2 \text{ km}$
$v_{12} - v_2$	$\min(\infty, 4,55 + 6,36) = 10,91 \text{ km}$
$v_{12} - v_3$	$\min(\infty, 4,55 + 6,28) = 10,83 \text{ km}$
$v_{12} - v_4$	$\min(\infty, 4,55 + 6,33) = 10,88 \text{ km}$
$v_{12} - v_5$	$\min(\infty, 4,55 + 7,25) = 11,8 \text{ km}$
$v_{12} - v_6$	$\min(\infty, 4,55 + 5,94) = 10,49 \text{ km}$
$v_{12} - v_7$	$\min(\infty, 4,55 + 4,69) = 9,24 \text{ km}$
$v_{12} - v_8$	$\min(\infty, 4,55 + 5,62) = 10,17 \text{ km}$
$v_{12} - v_9$	$\min(\infty, 4,55 + 5,92) = 10,47 \text{ km}$
$v_{12} - v_{10}$	$\min(\infty, 4,55 + 2,66) = 7,21 \text{ km}$
$v_{12} - v_{11}$	$\min(\infty, 4,55 + 12,96) = 17,51 \text{ km}$
$v_{12} - v_{13}$	$\min(\infty, 4,55 + -0,05) = 4,5 \text{ km}$
$v_{12} - v_{14}$	$\min(\infty, 4,55 + 22,01) = 26,56 \text{ km}$
$v_{12} - v_{15}$	$\min(\infty, 4,55 + 6,96) = 11,51 \text{ km}$
$v_{12} - v_{16}$	$\min(\infty, 4,55 + 20,43) = 24,98 \text{ km}$
$v_{12} - v_{17}$	$\min(\infty, 4,55 + 2,59) = 7,14 \text{ km}$
$v_{12} - v_{18}$	$\min(\infty, 4,55 + 19,15) = 23,7 \text{ km}$
$v_{12} - v_{19}$	$\min(\infty, 4,55 + 17,55) = 22,1 \text{ km}$
$v_{12} - v_{20}$	$\min(\infty, 4,55 + 16,53) = 21,08 \text{ km}$
$v_{12} - v_{21}$	$\min(\infty, 4,55 + 3,75) = 8,3 \text{ km}$
$v_{12} - v_{22}$	$\min(\infty, 4,55 + 21,96) = 26,51 \text{ km}$
$v_{12} - v_{23}$	$\min(\infty, 4,55 + 0,02) = 4,57 \text{ km}$
$v_{12} - v_{24}$	$\min(\infty, 4,55 + 6,0) = 10,55 \text{ km}$
$v_{12} - v_{25}$	$\min(\infty, 4,55 + 23,1) = 27,65 \text{ km}$
$v_{12} - v_{26}$	$\min(\infty, 4,55 + 10,75) = 15,3 \text{ km}$
$v_{12} - v_{27}$	$\min(\infty, 4,55 + 18,9) = 23,45 \text{ km}$
$v_{12} - v_{28}$	$\min(\infty, 4,55 + 5,88) = 10,43 \text{ km}$

Menurut tabel di atas, diketahui bahwa titik v_{13} merupakan titik terdekat dari v_{12} , maka v_{13} ditandai agar tidak dilalui pada iterasi berikutnya.

c. Iterasi 3

Berdasarkan tabel 4.11 diketahui titik terdekat dari v_{12} adalah v_{13} maka akan dicari titik terdekat dari v_{13} .

Tabel 4.12 Iterasi 3

Rute	Perhitungan jarak minimum
$v_{13} - v_1$	$\min(\infty, 4,5 + 6,21) = 10,71 \text{ km}$
$v_{13} - v_2$	$\min(\infty, 4,5 + 6,41) = 10,91 \text{ km}$
$v_{13} - v_3$	$\min(\infty, 4,5 + 6,33) = 10,83 \text{ km}$
$v_{13} - v_4$	$\min(\infty, 4,5 + 6,38) = 10,88 \text{ km}$
$v_{13} - v_5$	$\min(\infty, 4,5 + 5,82) = 10,32 \text{ km}$
$v_{13} - v_6$	$\min(\infty, 4,5 + 7,05) = 11,55 \text{ km}$
$v_{13} - v_7$	$\min(\infty, 4,5 + 4,97) = 9,47 \text{ km}$
$v_{13} - v_8$	$\min(\infty, 4,5 + 6,04) = 10,54 \text{ km}$
$v_{13} - v_9$	$\min(\infty, 4,5 + 7,1) = 11,6 \text{ km}$
$v_{13} - v_{10}$	$\min(\infty, 4,5 + 0,35) = 4,85 \text{ km}$
$v_{13} - v_{11}$	$\min(\infty, 4,5 + 13,02) = 17,52 \text{ km}$
$v_{13} - v_{14}$	$\min(\infty, 4,5 + 20,21) = 24,71 \text{ km}$
$v_{13} - v_{15}$	$\min(\infty, 4,5 + 7,84) = 12,34 \text{ km}$
$v_{13} - v_{16}$	$\min(\infty, 4,5 + 20,52) = 25,02 \text{ km}$
$v_{13} - v_{17}$	$\min(\infty, 4,5 + 2,64) = 7,14 \text{ km}$
$v_{13} - v_{18}$	$\min(\infty, 4,5 + 19,76) = 24,26 \text{ km}$
$v_{13} - v_{19}$	$\min(\infty, 4,5 + 18,51) = 23,01 \text{ km}$
$v_{13} - v_{20}$	$\min(\infty, 4,5 + 16,58) = 21,08 \text{ km}$
$v_{13} - v_{21}$	$\min(\infty, 4,5 + 5,19) = 9,69 \text{ km}$
$v_{13} - v_{22}$	$\min(\infty, 4,5 + 22,11) = 26,61 \text{ km}$
$v_{13} - v_{23}$	$\min(\infty, 4,5 + 3,96) = 8,46 \text{ km}$
$v_{13} - v_{24}$	$\min(\infty, 4,5 + 8,41) = 12,91 \text{ km}$
$v_{13} - v_{25}$	$\min(\infty, 4,5 + 23,28) = 27,78 \text{ km}$
$v_{13} - v_{26}$	$\min(\infty, 4,5 + 11,33) = 15,83 \text{ km}$
$v_{13} - v_{27}$	$\min(\infty, 4,5 + 19,59) = 24,09 \text{ km}$
$v_{13} - v_{28}$	$\min(\infty, 4,5 + 7,07) = 11,57 \text{ km}$

Mengacu tabel di atas, diketahui bahwa titik v_{10} merupakan titik terdekat dari v_{13} , maka v_{10} ditandai agar tidak dilalui pada iterasi berikutnya.

d. Iterasi 4

Berdasarkan tabel 4.12 diketahui titik terdekat dari v_{13} adalah v_{10} maka akan dicari titik terdekat dari v_{10} .

Tabel 4.13 Iterasi 4

Rute	Perhitungan jarak minimum
$v_{10} - v_1$	$\min(\infty, 4,85 + 1,76) = 6,61 \text{ km}$
$v_{10} - v_2$	$\min(\infty, 4,85 + 1,53) = 6,38 \text{ km}$
$v_{10} - v_3$	$\min(\infty, 4,85 + 1,55) = 6,40 \text{ km}$
$v_{10} - v_4$	$\min(\infty, 4,85 + 1,66) = 6,51 \text{ km}$
$v_{10} - v_5$	$\min(\infty, 4,85 + 2,83) = 7,68 \text{ km}$
$v_{10} - v_6$	$\min(\infty, 4,85 + 5,49) = 10,34 \text{ km}$
$v_{10} - v_7$	$\min(\infty, 4,85 + -0,37) = 4,48 \text{ km}$

Rute	Perhitungan jarak minimum
$v_{10} - v_8$	$\min(\infty, 4,85 + 3,18) = 8,03 \text{ km}$
$v_{10} - v_9$	$\min(\infty, 4,85 + 5,63) = 10,48 \text{ km}$
$v_{10} - v_{11}$	$\min(\infty, 4,85 + 7,99) = 12,84 \text{ km}$
$v_{10} - v_{14}$	$\min(\infty, 4,85 + 17,37) = 22,22 \text{ km}$
$v_{10} - v_{15}$	$\min(\infty, 4,85 + 5,81) = 10,66 \text{ km}$
$v_{10} - v_{16}$	$\min(\infty, 4,85 + 15,31) = 20,16 \text{ km}$
$v_{10} - v_{17}$	$\min(\infty, 4,85 + -1,36) = 3,49 \text{ km}$
$v_{10} - v_{18}$	$\min(\infty, 4,85 + 14,56) = 19,41 \text{ km}$
$v_{10} - v_{19}$	$\min(\infty, 4,85 + 13,65) = 18,50 \text{ km}$
$v_{10} - v_{20}$	$\min(\infty, 4,85 + 11,62) = 16,47 \text{ km}$
$v_{10} - v_{21}$	$\min(\infty, 4,85 + 4,29) = 9,14 \text{ km}$
$v_{10} - v_{22}$	$\min(\infty, 4,85 + 16,80) = 21,65 \text{ km}$
$v_{10} - v_{23}$	$\min(\infty, 4,85 + 4,16) = 9,01 \text{ km}$
$v_{10} - v_{24}$	$\min(\infty, 4,85 + 5,90) = 10,75 \text{ km}$
$v_{10} - v_{25}$	$\min(\infty, 4,85 + 17,95) = 22,80 \text{ km}$
$v_{10} - v_{26}$	$\min(\infty, 4,85 + 6,16) = 11,01 \text{ km}$
$v_{10} - v_{27}$	$\min(\infty, 4,85 + 14,46) = 19,31 \text{ km}$
$v_{10} - v_{28}$	$\min(\infty, 4,85 + 5,61) = 10,46 \text{ km}$

Berdasarkan tabel di atas, diketahui bahwa titik v_{17} merupakan titik terdekat dari v_{10} , maka v_{17} ditandai agar tidak dilalui pada iterasi berikutnya.

e. Iterasi 5

Berdasarkan tabel 4.13 diketahui titik terdekat dari v_{10} adalah v_{17} maka akan dicari titik terdekat dari v_{17} .

Tabel 4.14 Iterasi 5

Rute	Perhitungan jarak minimum
$v_{17} - v_1$	$\min(\infty, 3,49 + 3,56) = 7,05 \text{ km}$
$v_{17} - v_2$	$\min(\infty, 3,49 + 3,77) = 7,26 \text{ km}$
$v_{17} - v_3$	$\min(\infty, 3,49 + 3,70) = 7,19 \text{ km}$
$v_{17} - v_4$	$\min(\infty, 3,49 + 3,77) = 7,26 \text{ km}$
$v_{17} - v_5$	$\min(\infty, 3,49 + 3,29) = 6,78 \text{ km}$
$v_{17} - v_6$	$\min(\infty, 3,49 + 6,86) = 10,35 \text{ km}$
$v_{17} - v_7$	$\min(\infty, 3,49 + 2,90) = 6,39 \text{ km}$
$v_{17} - v_8$	$\min(\infty, 3,49 + 4,59) = 8,08 \text{ km}$
$v_{17} - v_9$	$\min(\infty, 3,49 + 7,01) = 10,50 \text{ km}$
$v_{17} - v_{11}$	$\min(\infty, 3,49 + 10,37) = 13,86 \text{ km}$
$v_{17} - v_{14}$	$\min(\infty, 3,49 + 17,71) = 21,20 \text{ km}$
$v_{17} - v_{15}$	$\min(\infty, 3,49 + 7,18) = 10,67 \text{ km}$
$v_{17} - v_{16}$	$\min(\infty, 3,49 + 17,90) = 21,39 \text{ km}$
$v_{17} - v_{18}$	$\min(\infty, 3,49 + 17,85) = 21,34 \text{ km}$
$v_{17} - v_{19}$	$\min(\infty, 3,49 + 17,13) = 20,62 \text{ km}$
$v_{17} - v_{20}$	$\min(\infty, 3,49 + 13,95) = 17,44 \text{ km}$

Rute	Perhitungan jarak minimum
$v_{17} - v_{21}$	$\min(\infty, 3,49 + 5,76) = 9,25 \text{ km}$
$v_{17} - v_{22}$	$\min(\infty, 3,49 + 19,56) = 23,05 \text{ km}$
$v_{17} - v_{23}$	$\min(\infty, 3,49 + 6,60) = 10,09 \text{ km}$
$v_{17} - v_{24}$	$\min(\infty, 3,49 + 9,39) = 12,88 \text{ km}$
$v_{17} - v_{25}$	$\min(\infty, 3,49 + 20,75) = 24,24 \text{ km}$
$v_{17} - v_{26}$	$\min(\infty, 3,49 + 9,51) = 13,00 \text{ km}$
$v_{17} - v_{27}$	$\min(\infty, 3,49 + 17,81) = 21,30 \text{ km}$
$v_{17} - v_{28}$	$\min(\infty, 3,49 + 6,99) = 10,48 \text{ km}$

Menurut tabel di atas, diketahui bahwa titik v_7 merupakan titik terdekat dari v_{17} , maka v_7 ditandai agar tidak dilalui pada iterasi berikutnya.

f. Iterasi 6

Berdasarkan tabel 4.14 diketahui titik terdekat dari v_{17} adalah v_7 maka akan dicari titik terdekat dari v_7 .

Tabel 4.15 Iterasi 6

Rute	Perhitungan jarak minimum
$v_7 - v_1$	$\min(\infty, 6,39 + 2,81) = 9,20 \text{ km}$
$v_7 - v_2$	$\min(\infty, 6,39 + 3,24) = 9,63 \text{ km}$
$v_7 - v_3$	$\min(\infty, 6,39 + 3,43) = 9,82 \text{ km}$
$v_7 - v_4$	$\min(\infty, 6,39 + 3,59) = 9,98 \text{ km}$
$v_7 - v_5$	$\min(\infty, 6,39 + 3,36) = 9,75 \text{ km}$
$v_7 - v_6$	$\min(\infty, 6,39 + 9,48) = 15,87 \text{ km}$
$v_7 - v_8$	$\min(\infty, 6,39 + 6,62) = 13,01 \text{ km}$
$v_7 - v_9$	$\min(\infty, 6,39 + 9,66) = 16,05 \text{ km}$
$v_7 - v_{11}$	$\min(\infty, 6,39 + 8,98) = 15,37 \text{ km}$
$v_7 - v_{14}$	$\min(\infty, 6,39 + 15,33) = 21,72 \text{ km}$
$v_7 - v_{15}$	$\min(\infty, 6,39 + 9,54) = 15,93 \text{ km}$
$v_7 - v_{16}$	$\min(\infty, 6,39 + 15,98) = 22,37 \text{ km}$
$v_7 - v_{18}$	$\min(\infty, 6,39 + 14,97) = 21,36 \text{ km}$
$v_7 - v_{19}$	$\min(\infty, 6,39 + 14,28) = 20,67 \text{ km}$
$v_7 - v_{20}$	$\min(\infty, 6,39 + 12,64) = 19,03 \text{ km}$
$v_7 - v_{21}$	$\min(\infty, 6,39 + 8,62) = 15,01 \text{ km}$
$v_7 - v_{22}$	$\min(\infty, 6,39 + 17,31) = 23,70 \text{ km}$
$v_7 - v_{23}$	$\min(\infty, 6,39 + 8,65) = 15,04 \text{ km}$
$v_7 - v_{24}$	$\min(\infty, 6,39 + 8,65) = 15,04 \text{ km}$
$v_7 - v_{25}$	$\min(\infty, 6,39 + 18,43) = 24,82 \text{ km}$
$v_7 - v_{26}$	$\min(\infty, 6,39 + 6,60) = 12,99 \text{ km}$
$v_7 - v_{27}$	$\min(\infty, 6,39 + 14,91) = 21,30 \text{ km}$
$v_7 - v_{28}$	$\min(\infty, 6,39 + 9,65) = 16,04 \text{ km}$

Mengacu tabel di atas, diketahui bahwa titik v_1 merupakan titik terdekat dari v_7 , maka v_1 ditandai agar tidak dilalui pada iterasi berikutnya.

g. Iterasi 7

Berdasarkan tabel 4.15 diketahui titik terdekat dari v_7 adalah v_1 maka akan dicari titik terdekat dari v_1 .

Tabel 4.16 Iterasi 7

Rute	Perhitungan jarak minimum
$v_1 - v_2$	$\min(\infty, 9,2 + -0,16) = 9,04 \text{ km}$
$v_1 - v_3$	$\min(\infty, 9,2 + 0,12) = 9,32 \text{ km}$
$v_1 - v_4$	$\min(\infty, 9,2 + 0,24) = 9,44 \text{ km}$
$v_1 - v_5$	$\min(\infty, 9,2 + 0,21) = 9,41 \text{ km}$
$v_1 - v_6$	$\min(\infty, 9,2 + 7,83) = 17,03 \text{ km}$
$v_1 - v_8$	$\min(\infty, 9,2 + 4,42) = 13,62 \text{ km}$
$v_1 - v_9$	$\min(\infty, 9,2 + 8,08) = 17,28 \text{ km}$
$v_1 - v_{11}$	$\min(\infty, 9,2 + 5,81) = 15,01 \text{ km}$
$v_1 - v_{14}$	$\min(\infty, 9,2 + 13,98) = 23,18 \text{ km}$
$v_1 - v_{15}$	$\min(\infty, 9,2 + 7,41) = 16,61 \text{ km}$
$v_1 - v_{16}$	$\min(\infty, 9,2 + 13,39) = 22,59 \text{ km}$
$v_1 - v_{18}$	$\min(\infty, 9,2 + 14,91) = 24,11 \text{ km}$
$v_1 - v_{19}$	$\min(\infty, 9,2 + 15,22) = 24,42 \text{ km}$
$v_1 - v_{20}$	$\min(\infty, 9,2 + 9,4) = 18,60 \text{ km}$
$v_1 - v_{21}$	$\min(\infty, 9,2 + 7,83) = 17,03 \text{ km}$
$v_1 - v_{22}$	$\min(\infty, 9,2 + 15,23) = 24,43 \text{ km}$
$v_1 - v_{23}$	$\min(\infty, 9,2 + 10,16) = 19,36 \text{ km}$
$v_1 - v_{24}$	$\min(\infty, 9,2 + 11,46) = 20,66 \text{ km}$
$v_1 - v_{25}$	$\min(\infty, 9,2 + 16,47) = 25,67 \text{ km}$
$v_1 - v_{26}$	$\min(\infty, 9,2 + 7,25) = 16,45 \text{ km}$
$v_1 - v_{27}$	$\min(\infty, 9,2 + 15,12) = 24,32 \text{ km}$
$v_1 - v_{28}$	$\min(\infty, 9,2 + 8,09) = 17,29 \text{ km}$

Berdasarkan tabel di atas, diketahui bahwa titik v_2 merupakan titik terdekat dari v_1 , maka v_2 ditandai agar tidak dilalui pada iterasi berikutnya.

h. Iterasi 8

Berdasarkan tabel 4.16 diketahui titik terdekat dari v_1 adalah v_2 maka akan dicari titik terdekat dari v_2 .

Tabel 4.17 Iterasi 8

Rute	Perhitungan jarak minimum
$v_2 - v_3$	$\min(\infty, 9,04 + 0,32) = 9,36 \text{ km}$
$v_2 - v_4$	$\min(\infty, 9,04 + 0,48) = 9,52 \text{ km}$
$v_2 - v_5$	$\min(\infty, 9,04 + 0,15) = 9,19 \text{ km}$
$v_2 - v_6$	$\min(\infty, 9,04 + 8,02) = 17,06 \text{ km}$
$v_2 - v_8$	$\min(\infty, 9,04 + 4,63) = 13,67 \text{ km}$
$v_2 - v_9$	$\min(\infty, 9,04 + 8,27) = 17,31 \text{ km}$
$v_2 - v_{11}$	$\min(\infty, 9,04 + 6,61) = 15,65 \text{ km}$
$v_2 - v_{14}$	$\min(\infty, 9,04 + 14,32) = 23,36 \text{ km}$
$v_2 - v_{15}$	$\min(\infty, 9,04 + 7,63) = 16,67 \text{ km}$
$v_2 - v_{16}$	$\min(\infty, 9,04 + 14,21) = 23,25 \text{ km}$
$v_2 - v_{18}$	$\min(\infty, 9,04 + 15,71) = 24,75 \text{ km}$
$v_2 - v_{19}$	$\min(\infty, 9,04 + 15,96) = 25,00 \text{ km}$
$v_2 - v_{20}$	$\min(\infty, 9,04 + 10,18) = 19,22 \text{ km}$
$v_2 - v_{21}$	$\min(\infty, 9,04 + 7,99) = 17,03 \text{ km}$
$v_2 - v_{22}$	$\min(\infty, 9,04 + 16,07) = 25,11 \text{ km}$
$v_2 - v_{23}$	$\min(\infty, 9,04 + 10,37) = 19,41 \text{ km}$
$v_2 - v_{24}$	$\min(\infty, 9,04 + 11,86) = 20,90 \text{ km}$
$v_2 - v_{25}$	$\min(\infty, 9,04 + 17,31) = 26,35 \text{ km}$
$v_2 - v_{26}$	$\min(\infty, 9,04 + 7,98) = 17,02 \text{ km}$
$v_2 - v_{27}$	$\min(\infty, 9,04 + 15,9) = 24,94 \text{ km}$
$v_2 - v_{28}$	$\min(\infty, 9,04 + 8,27) = 17,31 \text{ km}$

Berdasarkan tabel di atas, diketahui bahwa titik v_5 merupakan titik terdekat dari v_2 , maka v_5 ditandai agar tidak dilalui pada iterasi berikutnya.

i. Iterasi 9

Berdasarkan tabel 4.17 diketahui titik terdekat dari v_2 adalah v_5 maka akan dicari titik terdekat dari v_5 .

Tabel 4.18 Iterasi 9

Rute	Perhitungan jarak minimum
$v_5 - v_3$	$\min(\infty, 9,19 + -0,07) = 9,12 \text{ km}$
$v_5 - v_4$	$\min(\infty, 9,19 + -0,23) = 8,96 \text{ km}$
$v_5 - v_6$	$\min(\infty, 9,19 + 5,84) = 15,03 \text{ km}$
$v_5 - v_8$	$\min(\infty, 9,19 + 2,38) = 11,57 \text{ km}$
$v_5 - v_9$	$\min(\infty, 9,19 + 6,11) = 15,30 \text{ km}$
$v_5 - v_{11}$	$\min(\infty, 9,19 + 4,59) = 13,78 \text{ km}$
$v_5 - v_{14}$	$\min(\infty, 9,19 + 14,40) = 23,59 \text{ km}$
$v_5 - v_{15}$	$\min(\infty, 9,19 + 5,20) = 14,39 \text{ km}$
$v_5 - v_{16}$	$\min(\infty, 9,19 + 12,38) = 21,57 \text{ km}$
$v_5 - v_{18}$	$\min(\infty, 9,19 + 15,14) = 24,33 \text{ km}$
$v_5 - v_{19}$	$\min(\infty, 9,19 + 15,74) = 24,93 \text{ km}$
$v_5 - v_{20}$	$\min(\infty, 9,19 + 7,94) = 17,13 \text{ km}$

Rute	Perhitungan jarak minimum
$v_5 - v_{21}$	$\min(\infty, 9,19 + 6,29) = 15,48 \text{ km}$
$v_5 - v_{22}$	$\min(\infty, 9,19 + 14,53) = 23,72 \text{ km}$
$v_5 - v_{23}$	$\min(\infty, 9,19 + 9,71) = 18,90 \text{ km}$
$v_5 - v_{24}$	$\min(\infty, 9,19 + 11,93) = 21,12 \text{ km}$
$v_5 - v_{25}$	$\min(\infty, 9,19 + 15,82) = 25,01 \text{ km}$
$v_5 - v_{26}$	$\min(\infty, 9,19 + 7,80) = 16,99 \text{ km}$
$v_5 - v_{27}$	$\min(\infty, 9,19 + 15,44) = 24,63 \text{ km}$
$v_5 - v_{28}$	$\min(\infty, 9,19 + 6,13) = 15,32 \text{ km}$

Berdasarkan tabel di atas, diketahui bahwa titik v_4 merupakan titik terdekat dari v_5 , maka v_4 ditandai agar tidak dilalui pada iterasi berikutnya.

j. Iterasi 10

Berdasarkan tabel 4.18 diketahui titik terdekat dari v_5 adalah v_4 maka akan dicari titik terdekat dari v_4 .

Tabel 4.19 Iterasi 10

Rute	Perhitungan jarak minimum
$v_4 - v_3$	$\min(\infty, 8,96 + 0,17) = 9,13 \text{ km}$
$v_4 - v_6$	$\min(\infty, 8,96 + 7,60) = 16,56 \text{ km}$
$v_4 - v_8$	$\min(\infty, 8,96 + 4,19) = 13,15 \text{ km}$
$v_4 - v_9$	$\min(\infty, 8,96 + 7,85) = 16,81 \text{ km}$
$v_4 - v_{11}$	$\min(\infty, 8,96 + 6,69) = 15,65 \text{ km}$
$v_4 - v_{14}$	$\min(\infty, 8,96 + 14,59) = 23,55 \text{ km}$
$v_4 - v_{15}$	$\min(\infty, 8,96 + 7,18) = 16,14 \text{ km}$
$v_4 - v_{16}$	$\min(\infty, 8,96 + 14,37) = 23,33 \text{ km}$
$v_4 - v_{18}$	$\min(\infty, 8,96 + 16,14) = 25,10 \text{ km}$
$v_4 - v_{19}$	$\min(\infty, 8,96 + 16,43) = 25,39 \text{ km}$
$v_4 - v_{20}$	$\min(\infty, 8,96 + 10,21) = 19,17 \text{ km}$
$v_4 - v_{21}$	$\min(\infty, 8,96 + 7,66) = 16,62 \text{ km}$
$v_4 - v_{22}$	$\min(\infty, 8,96 + 16,30) = 25,26 \text{ km}$
$v_4 - v_{23}$	$\min(\infty, 8,96 + 10,33) = 19,29 \text{ km}$
$v_4 - v_{24}$	$\min(\infty, 8,96 + 12,16) = 21,12 \text{ km}$
$v_4 - v_{25}$	$\min(\infty, 8,96 + 17,56) = 26,52 \text{ km}$
$v_4 - v_{26}$	$\min(\infty, 8,96 + 8,46) = 17,42 \text{ km}$
$v_4 - v_{27}$	$\min(\infty, 8,96 + 16,35) = 25,31 \text{ km}$
$v_4 - v_{28}$	$\min(\infty, 8,96 + 7,86) = 16,82 \text{ km}$

Berdasarkan tabel di atas, diketahui bahwa titik v_3 merupakan titik terdekat dari v_4 , maka v_3 ditandai agar tidak dilalui pada iterasi berikutnya.

k. Iterasi 11

Berdasarkan tabel 4.19 diketahui titik terdekat dari v_4 adalah v_3 maka akan dicari titik terdekat dari v_3 .

Tabel 4.20 Iterasi 11

Rute	Perhitungan jarak minimum
$v_3 - v_6$	$\min(\infty, 9,13 + 7,71) = 16,84 \text{ km}$
$v_3 - v_8$	$\min(\infty, 9,13 + 4,31) = 13,44 \text{ km}$
$v_3 - v_9$	$\min(\infty, 9,13 + 7,96) = 17,09 \text{ km}$
$v_3 - v_{11}$	$\min(\infty, 9,13 + 6,72) = 15,85 \text{ km}$
$v_3 - v_{14}$	$\min(\infty, 9,13 + 14,55) = 23,68 \text{ km}$
$v_3 - v_{15}$	$\min(\infty, 9,13 + 7,31) = 16,44 \text{ km}$
$v_3 - v_{16}$	$\min(\infty, 9,13 + 14,37) = 23,50 \text{ km}$
$v_3 - v_{18}$	$\min(\infty, 9,13 + 16,02) = 25,15 \text{ km}$
$v_3 - v_{19}$	$\min(\infty, 9,13 + 16,28) = 25,41 \text{ km}$
$v_3 - v_{20}$	$\min(\infty, 9,13 + 10,25) = 19,38 \text{ km}$
$v_3 - v_{21}$	$\min(\infty, 9,13 + 7,73) = 16,86 \text{ km}$
$v_3 - v_{22}$	$\min(\infty, 9,13 + 16,28) = 25,41 \text{ km}$
$v_3 - v_{23}$	$\min(\infty, 9,13 + 10,28) = 19,41 \text{ km}$
$v_3 - v_{24}$	$\min(\infty, 9,13 + 12,01) = 21,14 \text{ km}$
$v_3 - v_{25}$	$\min(\infty, 9,13 + 17,52) = 26,65 \text{ km}$
$v_3 - v_{26}$	$\min(\infty, 9,13 + 8,31) = 17,44 \text{ km}$
$v_3 - v_{27}$	$\min(\infty, 9,13 + 16,22) = 25,35 \text{ km}$
$v_3 - v_{28}$	$\min(\infty, 9,13 + 7,97) = 17,10 \text{ km}$

Berdasarkan tabel di atas, diketahui bahwa titik v_8 merupakan titik terdekat dari v_3 , maka v_8 ditandai agar tidak dilalui pada iterasi berikutnya.

l. Iterasi 12

Berdasarkan tabel 4.20 diketahui titik terdekat dari v_3 adalah v_8 maka akan dicari titik terdekat dari v_8 .

Tabel 4.21 Iterasi 12

Rute	Perhitungan jarak minimum
$v_8 - v_6$	$\min(\infty, 13,44 + 3,47) = 16,91 \text{ km}$
$v_8 - v_9$	$\min(\infty, 13,44 + 3,73) = 17,17 \text{ km}$
$v_8 - v_{11}$	$\min(\infty, 13,44 + 9,69) = 23,13 \text{ km}$
$v_8 - v_{14}$	$\min(\infty, 13,44 + 18,22) = 31,66 \text{ km}$
$v_8 - v_{15}$	$\min(\infty, 13,44 + 3,02) = 16,46 \text{ km}$
$v_8 - v_{16}$	$\min(\infty, 13,44 + 17,48) = 30,92 \text{ km}$
$v_8 - v_{18}$	$\min(\infty, 13,44 + 20,33) = 33,77 \text{ km}$
$v_8 - v_{19}$	$\min(\infty, 13,44 + 20,49) = 33,93 \text{ km}$
$v_8 - v_{20}$	$\min(\infty, 13,44 + 12,7) = 26,14 \text{ km}$

Rute	Perhitungan jarak minimum
$v_8 - v_{21}$	$\min(\infty, 13,44 + 4,18) = 17,62 \text{ km}$
$v_8 - v_{22}$	$\min(\infty, 13,44 + 19,8) = 33,24 \text{ km}$
$v_8 - v_{23}$	$\min(\infty, 13,44 + 9,28) = 22,72 \text{ km}$
$v_8 - v_{24}$	$\min(\infty, 13,44 + 13,94) = 27,38 \text{ km}$
$v_8 - v_{25}$	$\min(\infty, 13,44 + 21,11) = 34,55 \text{ km}$
$v_8 - v_{26}$	$\min(\infty, 13,44 + 12,54) = 25,98 \text{ km}$
$v_8 - v_{27}$	$\min(\infty, 13,44 + 20,53) = 33,97 \text{ km}$
$v_8 - v_{28}$	$\min(\infty, 13,44 + 3,75) = 17,19 \text{ km}$

Berdasarkan tabel di atas, diketahui bahwa titik v_{15} merupakan titik terdekat dari v_8 , maka v_{15} ditandai agar tidak dilalui pada iterasi berikutnya.

m. Iterasi 13

Berdasarkan tabel 4.21 diketahui titik terdekat dari v_8 adalah v_{15} maka akan dicari titik terdekat dari v_{15} .

Tabel 4.22 Iterasi 13

Rute	Perhitungan jarak minimum
$v_{15} - v_6$	$\min(\infty, 16,46 + 1,42) = 17,88 \text{ km}$
$v_{15} - v_9$	$\min(\infty, 16,46 + 1,66) = 18,12 \text{ km}$
$v_{15} - v_{11}$	$\min(\infty, 16,46 + 12,01) = 28,47 \text{ km}$
$v_{15} - v_{14}$	$\min(\infty, 16,46 + 20,68) = 37,14 \text{ km}$
$v_{15} - v_{16}$	$\min(\infty, 16,46 + 19,64) = 36,10 \text{ km}$
$v_{15} - v_{18}$	$\min(\infty, 16,46 + 23,29) = 39,75 \text{ km}$
$v_{15} - v_{19}$	$\min(\infty, 16,46 + 23,52) = 39,98 \text{ km}$
$v_{15} - v_{20}$	$\min(\infty, 16,46 + 14,60) = 31,06 \text{ km}$
$v_{15} - v_{21}$	$\min(\infty, 16,46 + 3,65) = 20,11 \text{ km}$
$v_{15} - v_{22}$	$\min(\infty, 16,46 + 22,19) = 38,65 \text{ km}$
$v_{15} - v_{23}$	$\min(\infty, 16,46 + 10,17) = 26,63 \text{ km}$
$v_{15} - v_{24}$	$\min(\infty, 16,46 + 16,18) = 32,64 \text{ km}$
$v_{15} - v_{25}$	$\min(\infty, 16,46 + 23,53) = 39,99 \text{ km}$
$v_{15} - v_{26}$	$\min(\infty, 16,46 + 15,56) = 32,02 \text{ km}$
$v_{15} - v_{27}$	$\min(\infty, 16,46 + 23,53) = 39,99 \text{ km}$
$v_{15} - v_{28}$	$\min(\infty, 16,46 + 1,71) = 18,17 \text{ km}$

Berdasarkan tabel di atas, diketahui bahwa titik v_6 merupakan titik terdekat dari v_{15} , maka v_6 ditandai agar tidak dilalui pada iterasi berikutnya.

n. Iterasi 14

Berdasarkan tabel 4.22 diketahui titik terdekat dari v_{15} adalah v_6 maka akan dicari titik terdekat dari v_6 .

Tabel 4.23 Iterasi 14

Rute	Perhitungan jarak minimum
$v_6 - v_9$	$\min(\infty, 17,88 + 0,29) = 18,17 \text{ km}$
$v_6 - v_{11}$	$\min(\infty, 17,88 + 12,98) = 30,86 \text{ km}$
$v_6 - v_{14}$	$\min(\infty, 17,88 + 21,6) = 39,48 \text{ km}$
$v_6 - v_{16}$	$\min(\infty, 17,88 + 20,7) = 38,58 \text{ km}$
$v_6 - v_{18}$	$\min(\infty, 17,88 + 23,73) = 41,61 \text{ km}$
$v_6 - v_{19}$	$\min(\infty, 17,88 + 23,67) = 41,55 \text{ km}$
$v_6 - v_{20}$	$\min(\infty, 17,88 + 15,74) = 33,62 \text{ km}$
$v_6 - v_{21}$	$\min(\infty, 17,88 + 2,34) = 20,22 \text{ km}$
$v_6 - v_{22}$	$\min(\infty, 17,88 + 23,15) = 41,03 \text{ km}$
$v_6 - v_{23}$	$\min(\infty, 17,88 + 8,99) = 26,87 \text{ km}$
$v_6 - v_{24}$	$\min(\infty, 17,88 + 15,46) = 33,34 \text{ km}$
$v_6 - v_{25}$	$\min(\infty, 17,88 + 24,47) = 42,35 \text{ km}$
$v_6 - v_{26}$	$\min(\infty, 17,88 + 15,77) = 33,65 \text{ km}$
$v_6 - v_{27}$	$\min(\infty, 17,88 + 23,89) = 41,77 \text{ km}$
$v_6 - v_{28}$	$\min(\infty, 17,88 + 0,32) = 18,20 \text{ km}$

Berdasarkan tabel di atas, diketahui bahwa titik v_9 merupakan titik terdekat dari v_6 , maka v_9 ditandai agar tidak dilalui pada iterasi berikutnya.

o. Iterasi 15

Berdasarkan tabel 4.23 diketahui titik terdekat dari v_6 adalah v_9 maka akan dicari titik terdekat dari v_9 .

Tabel 4.24 Iterasi 15

Rute	Perhitungan jarak minimum
$v_9 - v_{11}$	$\min(\infty, 18,17 + 13,27) = 31,44 \text{ km}$
$v_9 - v_{14}$	$\min(\infty, 18,17 + 21,88) = 40,05 \text{ km}$
$v_9 - v_{16}$	$\min(\infty, 18,17 + 20,99) = 39,16 \text{ km}$
$v_9 - v_{18}$	$\min(\infty, 18,17 + 23,97) = 42,14 \text{ km}$
$v_9 - v_{19}$	$\min(\infty, 18,17 + 23,87) = 42,04 \text{ km}$
$v_9 - v_{20}$	$\min(\infty, 18,17 + 16,03) = 34,20 \text{ km}$
$v_9 - v_{21}$	$\min(\infty, 18,17 + 2,25) = 20,42 \text{ km}$
$v_9 - v_{22}$	$\min(\infty, 18,17 + 23,43) = 41,60 \text{ km}$
$v_9 - v_{23}$	$\min(\infty, 18,17 + 8,91) = 27,08 \text{ km}$
$v_9 - v_{24}$	$\min(\infty, 18,17 + 15,51) = 33,68 \text{ km}$
$v_9 - v_{25}$	$\min(\infty, 18,17 + 24,76) = 42,93 \text{ km}$

Rute	Perhitungan jarak minimum
$v_9 - v_{26}$	$\min(\infty, 18,17 + 15,99) = 34,16 \text{ km}$
$v_9 - v_{27}$	$\min(\infty, 18,17 + 24,12) = 42,29 \text{ km}$
$v_9 - v_{28}$	$\min(\infty, 18,17 + 0,05) = 18,22 \text{ km}$

Berdasarkan tabel di atas, diketahui bahwa titik v_{28} merupakan titik terdekat dari v_9 , maka v_{28} ditandai agar tidak dilalui pada iterasi berikutnya.

p. Iterasi 16

Berdasarkan tabel 4.24 diketahui titik terdekat dari v_9 adalah v_{28} maka akan dicari titik terdekat dari v_{28} .

Tabel 4.25 Iterasi 16

Rute	Perhitungan jarak minimum
$v_{28} - v_{11}$	$\min(\infty, 18,22 + 13,29) = 31,51 \text{ km}$
$v_{28} - v_{14}$	$\min(\infty, 18,22 + 21,9) = 40,12 \text{ km}$
$v_{28} - v_{16}$	$\min(\infty, 18,22 + 21,02) = 39,24 \text{ km}$
$v_{28} - v_{18}$	$\min(\infty, 18,22 + 23,97) = 42,19 \text{ km}$
$v_{28} - v_{19}$	$\min(\infty, 18,22 + 23,87) = 42,09 \text{ km}$
$v_{28} - v_{20}$	$\min(\infty, 18,22 + 16,06) = 34,28 \text{ km}$
$v_{28} - v_{21}$	$\min(\infty, 18,22 + 2,2) = 20,42 \text{ km}$
$v_{28} - v_{22}$	$\min(\infty, 18,22 + 23,46) = 41,68 \text{ km}$
$v_{28} - v_{23}$	$\min(\infty, 18,22 + 8,86) = 27,08 \text{ km}$
$v_{28} - v_{24}$	$\min(\infty, 18,22 + 15,48) = 33,7 \text{ km}$
$v_{28} - v_{25}$	$\min(\infty, 18,22 + 24,78) = 43,00 \text{ km}$
$v_{28} - v_{26}$	$\min(\infty, 18,22 + 15,99) = 34,21 \text{ km}$
$v_{28} - v_{27}$	$\min(\infty, 18,22 + 24,12) = 42,34 \text{ km}$

Berdasarkan tabel di atas, diketahui bahwa titik v_{21} merupakan titik terdekat dari v_{28} , maka v_{21} ditandai agar tidak dilalui pada iterasi berikutnya.

q. Iterasi 17

Berdasarkan tabel 4.25 diketahui titik terdekat dari v_{28} adalah v_{21} maka akan dicari titik terdekat dari v_{21} .

Tabel 4.26 Iterasi 17

Rute	Perhitungan jarak minimum
$v_{21} - v_{11}$	$\min(\infty, 20,42 + 13,8) = 34,22 \text{ km}$
$v_{21} - v_{14}$	$\min(\infty, 20,42 + 22,17) = 42,59 \text{ km}$
$v_{21} - v_{16}$	$\min(\infty, 20,42 + 21,61) = 42,03 \text{ km}$
$v_{21} - v_{18}$	$\min(\infty, 20,42 + 23,41) = 43,83 \text{ km}$
$v_{21} - v_{19}$	$\min(\infty, 20,42 + 22,89) = 43,31 \text{ km}$
$v_{21} - v_{20}$	$\min(\infty, 20,42 + 16,88) = 37,30 \text{ km}$
$v_{21} - v_{22}$	$\min(\infty, 20,42 + 23,82) = 44,24 \text{ km}$
$v_{21} - v_{23}$	$\min(\infty, 20,42 + 6,66) = 27,08 \text{ km}$
$v_{21} - v_{24}$	$\min(\infty, 20,42 + 13,53) = 33,95 \text{ km}$
$v_{21} - v_{25}$	$\min(\infty, 20,42 + 25,11) = 45,53 \text{ km}$
$v_{21} - v_{26}$	$\min(\infty, 20,42 + 15,18) = 35,60 \text{ km}$
$v_{21} - v_{27}$	$\min(\infty, 20,42 + 23,45) = 43,87 \text{ km}$

Berdasarkan tabel di atas, diketahui bahwa titik v_{23} merupakan titik terdekat dari v_{21} , maka v_{23} ditandai agar tidak dilalui pada iterasi berikutnya.

r. Iterasi 18

Berdasarkan tabel 4.26 diketahui titik terdekat dari v_{21} adalah v_{23} maka akan dicari titik terdekat dari v_{23} .

Tabel 4.27 Iterasi 18

Rute	Perhitungan jarak minimum
$v_{23} - v_{11}$	$\min(\infty, 27,08 + 16,97) = 44,05 \text{ km}$
$v_{23} - v_{14}$	$\min(\infty, 27,08 + 23,99) = 51,07 \text{ km}$
$v_{23} - v_{16}$	$\min(\infty, 27,08 + 24,45) = 51,53 \text{ km}$
$v_{23} - v_{18}$	$\min(\infty, 27,08 + 22,85) = 49,93 \text{ km}$
$v_{23} - v_{19}$	$\min(\infty, 27,08 + 20,99) = 48,07 \text{ km}$
$v_{23} - v_{20}$	$\min(\infty, 27,08 + 20,53) = 47,61 \text{ km}$
$v_{23} - v_{22}$	$\min(\infty, 27,08 + 25,95) = 53,03 \text{ km}$
$v_{23} - v_{24}$	$\min(\infty, 27,08 + 8,35) = 35,43 \text{ km}$
$v_{23} - v_{25}$	$\min(\infty, 27,08 + 27,08) = 54,16 \text{ km}$
$v_{23} - v_{26}$	$\min(\infty, 27,08 + 14,51) = 41,59 \text{ km}$
$v_{23} - v_{27}$	$\min(\infty, 27,08 + 22,53) = 49,61 \text{ km}$

Berdasarkan tabel di atas, diketahui bahwa titik v_{24} merupakan titik terdekat dari v_{23} , maka v_{24} ditandai agar tidak dilalui pada iterasi berikutnya.

s. Iterasi 19

Berdasarkan tabel 4.27 diketahui titik terdekat dari v_{23} adalah v_{24} maka akan dicari titik terdekat dari v_{24} .

Tabel 4.28 Iterasi 19

Rute	Perhitungan jarak minimum
$v_{24} - v_{11}$	$\min(\infty, 35,43 + 17,18) = 52,61 \text{ km}$
$v_{24} - v_{14}$	$\min(\infty, 35,43 + 21,07) = 56,5 \text{ km}$
$v_{24} - v_{16}$	$\min(\infty, 35,43 + 23,09) = 58,52 \text{ km}$
$v_{24} - v_{18}$	$\min(\infty, 35,43 + 17,52) = 52,95 \text{ km}$
$v_{24} - v_{19}$	$\min(\infty, 35,43 + 14,39) = 49,82 \text{ km}$
$v_{24} - v_{20}$	$\min(\infty, 35,43 + 20,74) = 56,17 \text{ km}$
$v_{24} - v_{22}$	$\min(\infty, 35,43 + 23,52) = 58,95 \text{ km}$
$v_{24} - v_{25}$	$\min(\infty, 35,43 + 24,36) = 59,79 \text{ km}$
$v_{24} - v_{26}$	$\min(\infty, 35,43 + 10,5) = 45,93 \text{ km}$
$v_{24} - v_{27}$	$\min(\infty, 35,43 + 16,84) = 52,27 \text{ km}$

Berdasarkan tabel di atas, diketahui bahwa titik v_{26} merupakan titik terdekat dari v_{24} , maka v_{26} ditandai agar tidak dilalui pada iterasi berikutnya.

t. Iterasi 20

Berdasarkan tabel 4.28 diketahui titik terdekat dari v_{24} adalah v_{26} maka akan dicari titik terdekat dari v_{26} .

Tabel 4.29 Iterasi 20

Rute	Perhitungan jarak minimum
$v_{26} - v_{11}$	$\min(\infty, 45,93 + 8,92) = 54,85 \text{ km}$
$v_{26} - v_{14}$	$\min(\infty, 45,93 + 10,57) = 56,5 \text{ km}$
$v_{26} - v_{16}$	$\min(\infty, 45,93 + 12,92) = 58,85 \text{ km}$
$v_{26} - v_{18}$	$\min(\infty, 45,93 + 8,43) = 54,36 \text{ km}$
$v_{26} - v_{19}$	$\min(\infty, 45,93 + 7,97) = 53,90 \text{ km}$
$v_{26} - v_{20}$	$\min(\infty, 45,93 + 11,82) = 57,75 \text{ km}$
$v_{26} - v_{22}$	$\min(\infty, 45,93 + 13,04) = 58,97 \text{ km}$
$v_{26} - v_{25}$	$\min(\infty, 45,93 + 13,86) = 59,79 \text{ km}$
$v_{26} - v_{27}$	$\min(\infty, 45,93 + 8,31) = 54,24 \text{ km}$

Berdasarkan tabel di atas, diketahui bahwa titik v_{19} merupakan titik terdekat dari v_{26} , maka v_{19} ditandai agar tidak dilalui pada iterasi berikutnya.

u. Iterasi 21

Berdasarkan tabel 4.29 diketahui titik terdekat dari v_{26} adalah v_{19} maka akan dicari titik terdekat dari v_{19} .

Tabel 4.30 Iterasi 21

Rute	Perhitungan jarak minimum
$v_{19} - v_{11}$	$\min(\infty, 53,9 + 15,57) = 69,47 \text{ km}$
$v_{19} - v_{14}$	$\min(\infty, 53,9 + 10,99) = 64,89 \text{ km}$
$v_{19} - v_{16}$	$\min(\infty, 53,9 + 16,09) = 69,99 \text{ km}$
$v_{19} - v_{18}$	$\min(\infty, 53,9 + 4,77) = 58,67 \text{ km}$
$v_{19} - v_{20}$	$\min(\infty, 53,9 + 17,4) = 71,3 \text{ km}$
$v_{19} - v_{22}$	$\min(\infty, 53,9 + 14,46) = 68,36 \text{ km}$
$v_{19} - v_{25}$	$\min(\infty, 53,9 + 14,55) = 68,45 \text{ km}$
$v_{19} - v_{27}$	$\min(\infty, 53,9 + 3,55) = 57,45 \text{ km}$

Berdasarkan tabel di atas, diketahui bahwa titik v_{27} merupakan titik terdekat dari v_{19} , maka v_{27} ditandai agar tidak dilalui pada iterasi berikutnya.

v. Iterasi 22

Berdasarkan tabel 4.30 diketahui titik terdekat dari v_{19} adalah v_{27} maka akan dicari titik terdekat dari v_{27} .

Tabel 4.31 Iterasi 22

Rute	Perhitungan jarak minimum
$v_{27} - v_{11}$	$\min(\infty, 57,45 + 14,06) = 71,51 \text{ km}$
$v_{27} - v_{14}$	$\min(\infty, 57,45 + 7,54) = 64,99 \text{ km}$
$v_{27} - v_{16}$	$\min(\infty, 57,45 + 13,13) = 70,58 \text{ km}$
$v_{27} - v_{18}$	$\min(\infty, 57,45 + 1,26) = 58,71 \text{ km}$
$v_{27} - v_{20}$	$\min(\infty, 57,45 + 15,23) = 72,68 \text{ km}$
$v_{27} - v_{22}$	$\min(\infty, 57,45 + 11,16) = 68,61 \text{ km}$
$v_{27} - v_{25}$	$\min(\infty, 57,45 + 11,13) = 68,58 \text{ km}$

Berdasarkan tabel di atas, diketahui bahwa titik v_{18} merupakan titik terdekat dari v_{27} , maka v_{18} ditandai agar tidak dilalui pada iterasi berikutnya.

w. Iterasi 23

Berdasarkan tabel 4.31 diketahui titik terdekat dari v_{27} adalah v_{18} ,

maka akan dicari titik terdekat dari v_{18} .

Tabel 4.32 Iterasi 23

Rute	Perhitungan jarak minimum
$v_{18} - v_{11}$	$\min(\infty, 58,71 + 13,36) = 72,07 \text{ km}$
$v_{18} - v_{14}$	$\min(\infty, 58,71 + 6,29) = 65,0 \text{ km}$
$v_{18} - v_{16}$	$\min(\infty, 58,71 + 11,97) = 70,68 \text{ km}$
$v_{18} - v_{20}$	$\min(\infty, 58,71 + 14,3) = 73,01 \text{ km}$
$v_{18} - v_{22}$	$\min(\infty, 58,71 + 9,92) = 68,63 \text{ km}$
$v_{18} - v_{25}$	$\min(\infty, 58,71 + 9,87) = 68,58 \text{ km}$

Berdasarkan tabel di atas, diketahui bahwa titik v_{14} merupakan titik terdekat dari v_{18} , maka v_{14} ditandai agar tidak dilalui pada iterasi berikutnya.

x. Iterasi 24

Berdasarkan tabel 4.32 diketahui titik terdekat dari v_{18} adalah v_{14} maka akan dicari titik terdekat dari v_{14} .

Tabel 4.33 Iterasi 24

Rute	Perhitungan jarak minimum
$v_{14} - v_{11}$	$\min(\infty, 65,0 + 8,69) = 73,69 \text{ km}$
$v_{14} - v_{16}$	$\min(\infty, 65,0 + 2,07) = 67,07 \text{ km}$
$v_{14} - v_{20}$	$\min(\infty, 65,0 + 7,02) = 72,02 \text{ km}$
$v_{14} - v_{22}$	$\min(\infty, 65,0 + -1,25) = 63,75 \text{ km}$
$v_{14} - v_{25}$	$\min(\infty, 65,0 + -2,34) = 62,66 \text{ km}$

Berdasarkan tabel di atas, diketahui bahwa titik v_{25} merupakan titik terdekat dari v_{14} , maka v_{25} ditandai agar tidak dilalui pada iterasi berikutnya.

y. Iterasi 25

Berdasarkan tabel 4.33 diketahui titik terdekat dari v_{14} adalah v_{25} maka akan dicari titik terdekat dari v_{25} .

Tabel 4.34 Iterasi 25

Rute	Perhitungan jarak minimum
$v_{25} - v_{11}$	$\min(\infty, 62,66 + 11,52) = 74,18 \text{ km}$
$v_{25} - v_{16}$	$\min(\infty, 62,66 + 4,6) = 67,26 \text{ km}$
$v_{25} - v_{20}$	$\min(\infty, 62,66 + 9,67) = 72,33 \text{ km}$
$v_{25} - v_{22}$	$\min(\infty, 62,66 + 1,37) = 64,03 \text{ km}$

Berdasarkan tabel di atas, diketahui bahwa titik v_{22} merupakan titik terdekat dari v_{25} , maka v_{22} ditandai agar tidak dilalui pada iterasi berikutnya.

z. Iterasi 26

Berdasarkan tabel 4.34 diketahui titik terdekat dari v_{25} adalah v_{22} maka akan dicari titik terdekat dari v_{22} .

Tabel 4.35 Iterasi 26

Rute	Perhitungan jarak minimum
$v_{22} - v_{11}$	$\min(\infty, 64,03 + 10,18) = 74,21 \text{ km}$
$v_{22} - v_{16}$	$\min(\infty, 64,03 + 3,32) = 67,35 \text{ km}$
$v_{22} - v_{20}$	$\min(\infty, 64,03 + 8,31) = 72,34 \text{ km}$

Berdasarkan tabel di atas, diketahui bahwa titik v_{16} merupakan titik terdekat dari v_{22} , maka v_{16} ditandai agar tidak dilalui pada iterasi berikutnya.

aa. Iterasi 27

Berdasarkan tabel 4.35 diketahui titik terdekat dari v_{22} adalah v_{16} maka akan dicari titik terdekat dari v_{16} .

Tabel 4.36 Iterasi 27

Rute	Perhitungan jarak minimum
$v_{16} - v_{11}$	$\min(\infty, 67,35 + 7,81) = 75,16 \text{ km}$
$v_{16} - v_{20}$	$\min(\infty, 67,35 + 5,25) = 72,6 \text{ km}$

Berdasarkan tabel di atas, diketahui bahwa titik v_{20} merupakan titik terdekat dari v_{16} , maka v_{20} ditandai agar tidak dilalui pada iterasi berikutnya.

bb. Iterasi 28

Berdasarkan tabel 4.36 diketahui titik terdekat dari v_{16} adalah v_{20} maka akan dicari titik terdekat dari v_{20} .

Tabel 4.37 Iterasi 28

Rute	Perhitungan jarak minimum
$v_{20} - v_{11}$	$\min(\infty, 72,6 + 3,66) = 76,26 \text{ km}$

Iterasi 28 menunjukkan bahwa titik terakhir ialah v_{11} , dengan demikian menunjukkan bahwa semua titik telah dilalui. Langkah selanjutnya ialah menambah satu iterasi untuk mencari nilai optimal titik menuju titik awal sekaligus memeriksa siklus negatif.

cc. Iterasi 29

Tabel 4.38 Iterasi 29

Rute	Perhitungan jarak minimum
$v_{11} - A$	$\min(\infty, 76,25 + 21,23) = 97,48 \text{ km}$

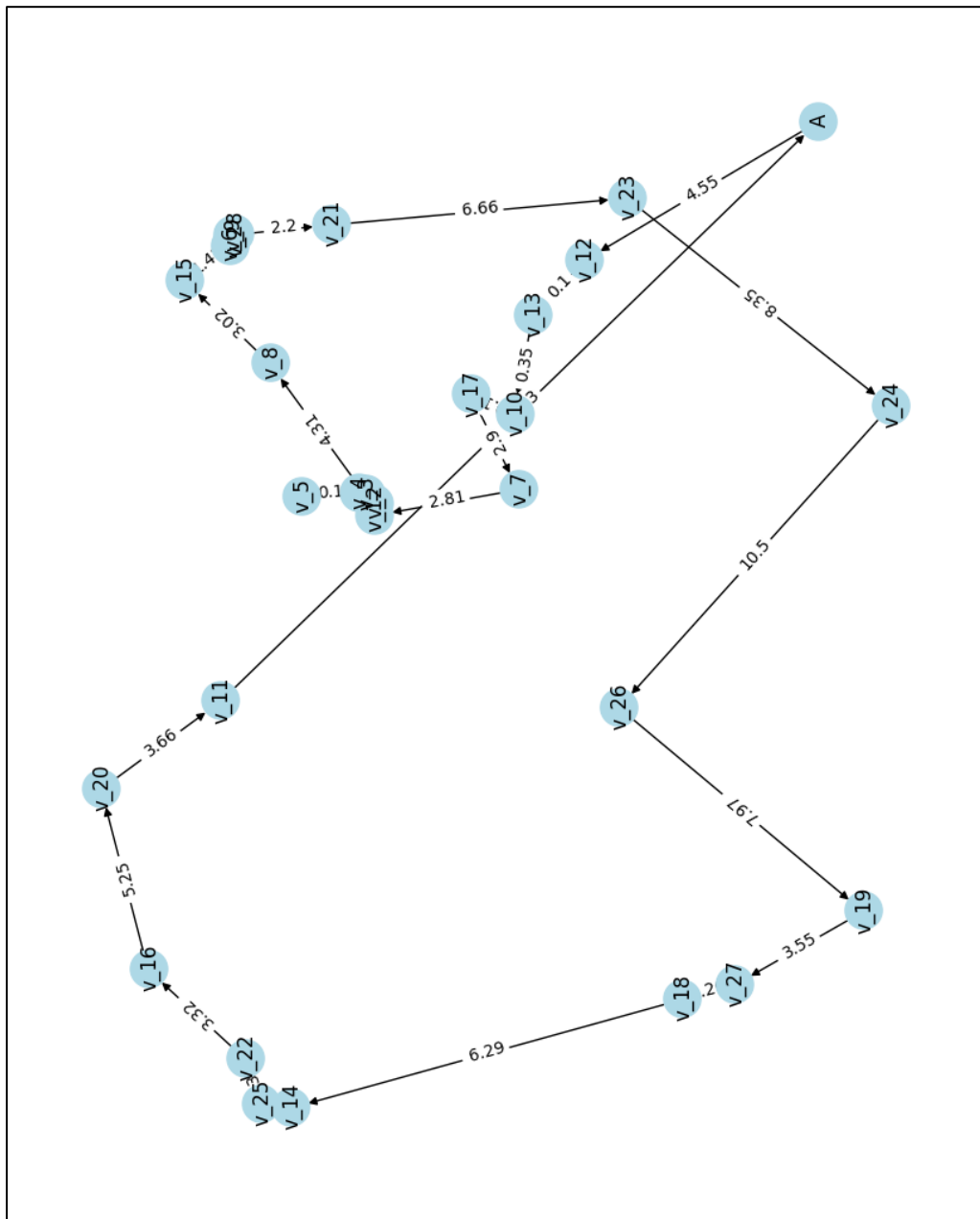
Total jarak dari titik awal melewati semua titik kemudian kembali ke titik awal lagi adalah 97,48 km hal ini menunjukkan bahwa

$$d(v) = 76,26 \text{ km} < d(u) + w(u, v) = 97,48 \text{ km}$$

Artinya, tidak ada siklus negatif. Sehingga nilai optimal untuk rute terpendek distribusi adalah 97,48 km dengan rute sebagai berikut.

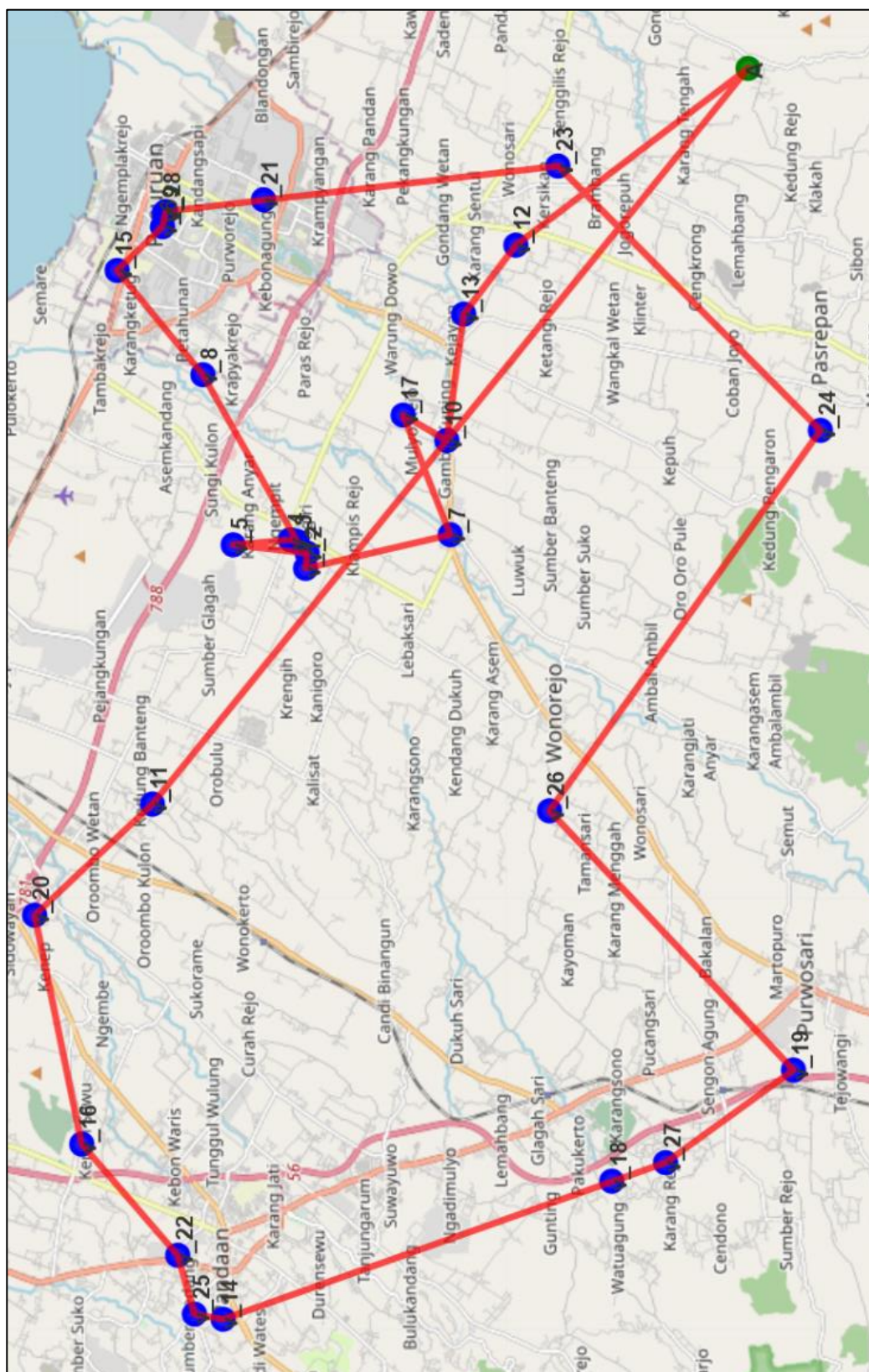
$$\begin{aligned} A &\rightarrow v_{12} \rightarrow v_{13} \rightarrow v_{10} \rightarrow v_{17} \rightarrow v_7 \rightarrow v_1 \rightarrow v_2 \rightarrow v_5 \rightarrow v_4 \rightarrow v_3 \rightarrow v_8 \\ &\rightarrow v_{15} \rightarrow v_6 \rightarrow v_9 \rightarrow v_{28} \rightarrow v_{21} \rightarrow v_{23} \rightarrow v_{24} \rightarrow v_{26} \rightarrow v_{19} \rightarrow v_{27} \\ &\rightarrow v_{18} \rightarrow v_{14} \rightarrow v_{25} \rightarrow v_{22} \rightarrow v_{16} \rightarrow v_{20} \rightarrow v_{11} \rightarrow A. \end{aligned}$$

3. Membuat graf Akhir Setelah menemukan nilai optimal yang menentukan rute terpendek distribusi air mineral Santri, langkah berikutnya adalah membuat graf yang merepresentasikan rute tersebut. Berikut ini adalah graf yang menunjukkan rute terpendek yang melewati semua titik distribusi dan kembali ke titik awal.



Gambar 4.2 Graf Rute Terpendek dengan Sifat *Hamiltonian Cycle*

Berikut ini adalah visualisasi graf dalam bentuk peta untuk memberikan gambaran lokasi yang lebih nyata.



Gambar 4.3 Visualisasi Graf dengan Sifat *Hamiltonian Cycle* pada Peta

4.2 Implementasi Algoritma Bellman-Ford dalam Menentukan Rute Terpendek Distribusi Air Mineral Santri.

Implementasi Algoritma Bellman-Ford tanpa siklus hamilton tidak dikerjakan secara manual, sebab data terlalu besar dan kombinasi titik untuk setiap iterasi yang mencapai ribuan sehingga tidak memungkinkan dikerjakan secara manual, karena akan memakan waktu yang lama dan meningkatkan risiko kesalahan. Penggunaan bahasa pemrograman diperlukan untuk mencari rute terpendek pada kasus ini, sebab akan menghasilkan perhitungan yang lebih cepat dan akurat. Penulis menggunakan bahasa pemrograman *python* sebagai alat bantu untuk mencari rute terpendek dengan Algoritma Bellman-Ford. Berikut tahapan implementasi Algoritma Bellman-Ford dengan bantuan *python*.

1. *Import library*.
 - a. *Pandas*, untuk membaca dan memproses data.
 - b. *Networkx*, untuk membuat dan memproses graf.
2. Membaca matriks jarak dari tabel 3.4 yang disimpan dalam file *excel*.
3. Membuat graf berarah berbobot
4. Menjalankan Algoritma Bellman-Ford untuk mencari rute terpendek dari node A ke semua node lain.
5. Menampilkan hasil jarak dan rute terpendek dari A ke setiap node tujuan.

Berikut hasil rute terpendek dengan Algoritma Bellman-Ford dengan dengan Pabrik AMDK Santri sebagai titik awal.

Tabel 4.39 Rute terpendek dari titik A ke titik lainnya

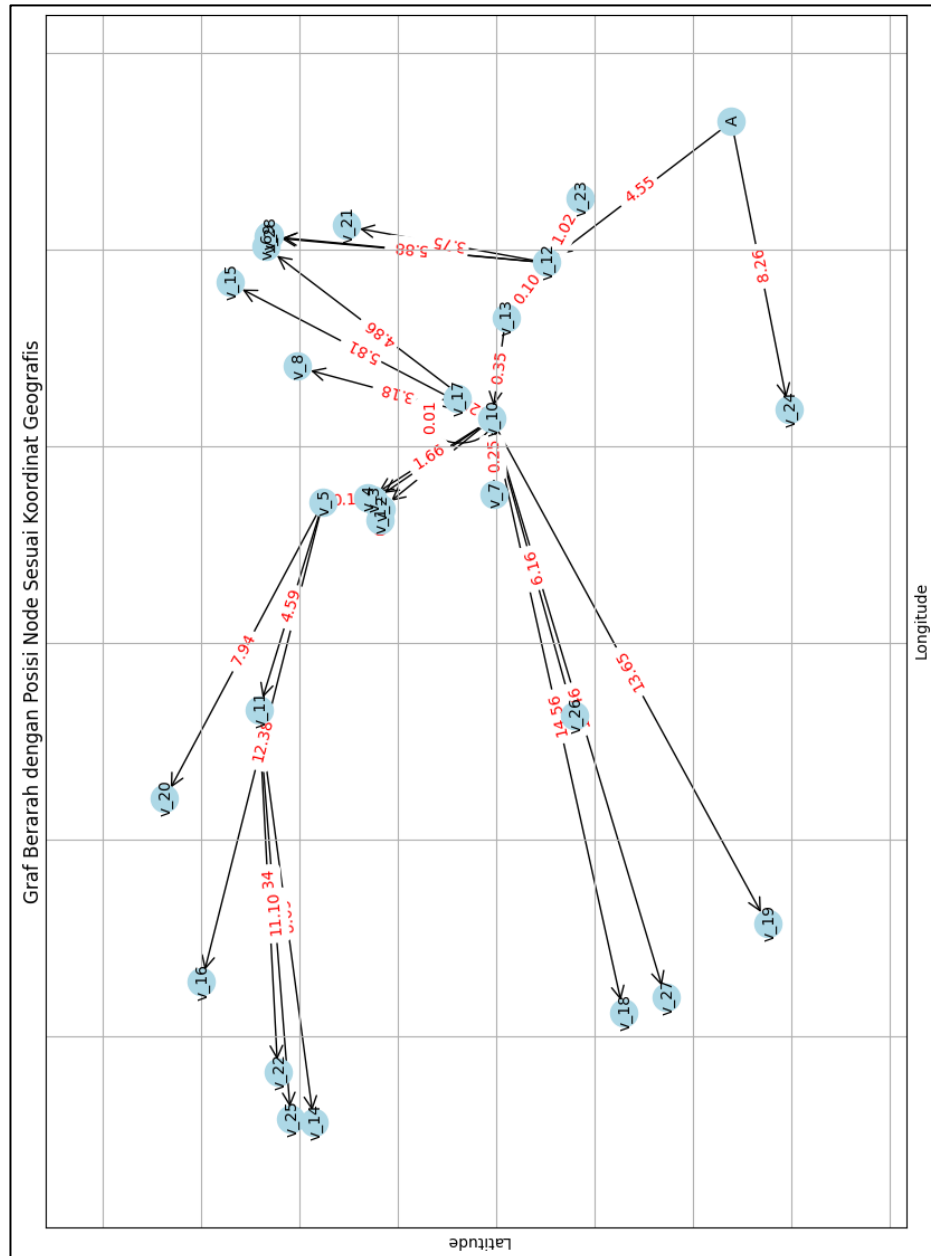
Titik tujuan	Total jarak	Rute
v_1	6,22	$A \rightarrow v_{12} \rightarrow v_{13} \rightarrow v_{10} \rightarrow v_2 \rightarrow v_1$
v_2	6,38	$A \rightarrow v_{12} \rightarrow v_{13} \rightarrow v_{10} \rightarrow v_2$
v_3	6,40	$A \rightarrow v_{12} \rightarrow v_{13} \rightarrow v_{10} \rightarrow v_3$
v_4	6,51	$A \rightarrow v_{12} \rightarrow v_{13} \rightarrow v_{10} \rightarrow v_4$
v_5	6,33	$A \rightarrow v_{12} \rightarrow v_{13} \rightarrow v_{10} \rightarrow v_3 \rightarrow v_5$
v_6	10,34	$A \rightarrow v_{12} \rightarrow v_{13} \rightarrow v_{10} \rightarrow v_6$
v_7	4,48	$A \rightarrow v_{12} \rightarrow v_{13} \rightarrow v_{10} \rightarrow v_7$
v_8	8,03	$A \rightarrow v_{12} \rightarrow v_{13} \rightarrow v_{10} \rightarrow v_8$
v_9	10,47	$A \rightarrow v_{12} \rightarrow v_9$
v_{10}	4,85	$A \rightarrow v_{12} \rightarrow v_{13} \rightarrow v_{10}$
v_{11}	10,92	$A \rightarrow v_{12} \rightarrow v_{13} \rightarrow v_{10} \rightarrow v_3 \rightarrow v_5 \rightarrow v_{11}$
v_{12}	4,55	$A \rightarrow v_{12}$
v_{13}	4,50	$A \rightarrow v_{12} \rightarrow v_{13}$
v_{14}	19,61	$A \rightarrow v_{12} \rightarrow v_{13} \rightarrow v_{10} \rightarrow v_3 \rightarrow v_5 \rightarrow v_{11} \rightarrow v_{14}$
v_{15}	10,66	$A \rightarrow v_{12} \rightarrow v_{13} \rightarrow v_{10} \rightarrow v_{15}$
v_{16}	18,71	$A \rightarrow v_{12} \rightarrow v_{13} \rightarrow v_{10} \rightarrow v_3 \rightarrow v_5 \rightarrow v_{16}$
v_{17}	3,49	$A \rightarrow v_{12} \rightarrow v_{13} \rightarrow v_{10} \rightarrow v_{17}$
v_{18}	19,41	$A \rightarrow v_{12} \rightarrow v_{13} \rightarrow v_{10} \rightarrow v_{18}$
v_{19}	18,5	$A \rightarrow v_{12} \rightarrow v_{13} \rightarrow v_{10} \rightarrow v_{19}$
v_{20}	14,27	$A \rightarrow v_{12} \rightarrow v_{13} \rightarrow v_{10} \rightarrow v_3 \rightarrow v_5 \rightarrow v_{20}$
v_{21}	8,30	$A \rightarrow v_{12} \rightarrow v_{21}$
v_{22}	18,36	$A \rightarrow v_{12} \rightarrow v_{13} \rightarrow v_{10} \rightarrow v_3 \rightarrow v_5 \rightarrow v_{11} \rightarrow v_{14} \rightarrow v_{22}$
v_{23}	4,57	$A \rightarrow v_{12} \rightarrow v_{23}$
v_{24}	8,26	$A \rightarrow v_{24}$
v_{25}	17,27	$A \rightarrow v_{12} \rightarrow v_{13} \rightarrow v_{10} \rightarrow v_3 \rightarrow v_5 \rightarrow v_{11} \rightarrow v_{14} \rightarrow v_{25}$
v_{26}	11,01	$A \rightarrow v_{12} \rightarrow v_{13} \rightarrow v_{10} \rightarrow v_{26}$
v_{27}	19,31	$A \rightarrow v_{12} \rightarrow v_{13} \rightarrow v_{10} \rightarrow v_{27}$
v_{28}	10,43	$A \rightarrow v_{12} \rightarrow v_{28}$

Berdasarkan tabel 4.39, ditemukan bahwa rute dengan jarak terpendek adalah dari titik awal A menuju v_{17} dengan melewati titik v_{12}, v_{13}, v_{10} .

6. Membuat graf akhir

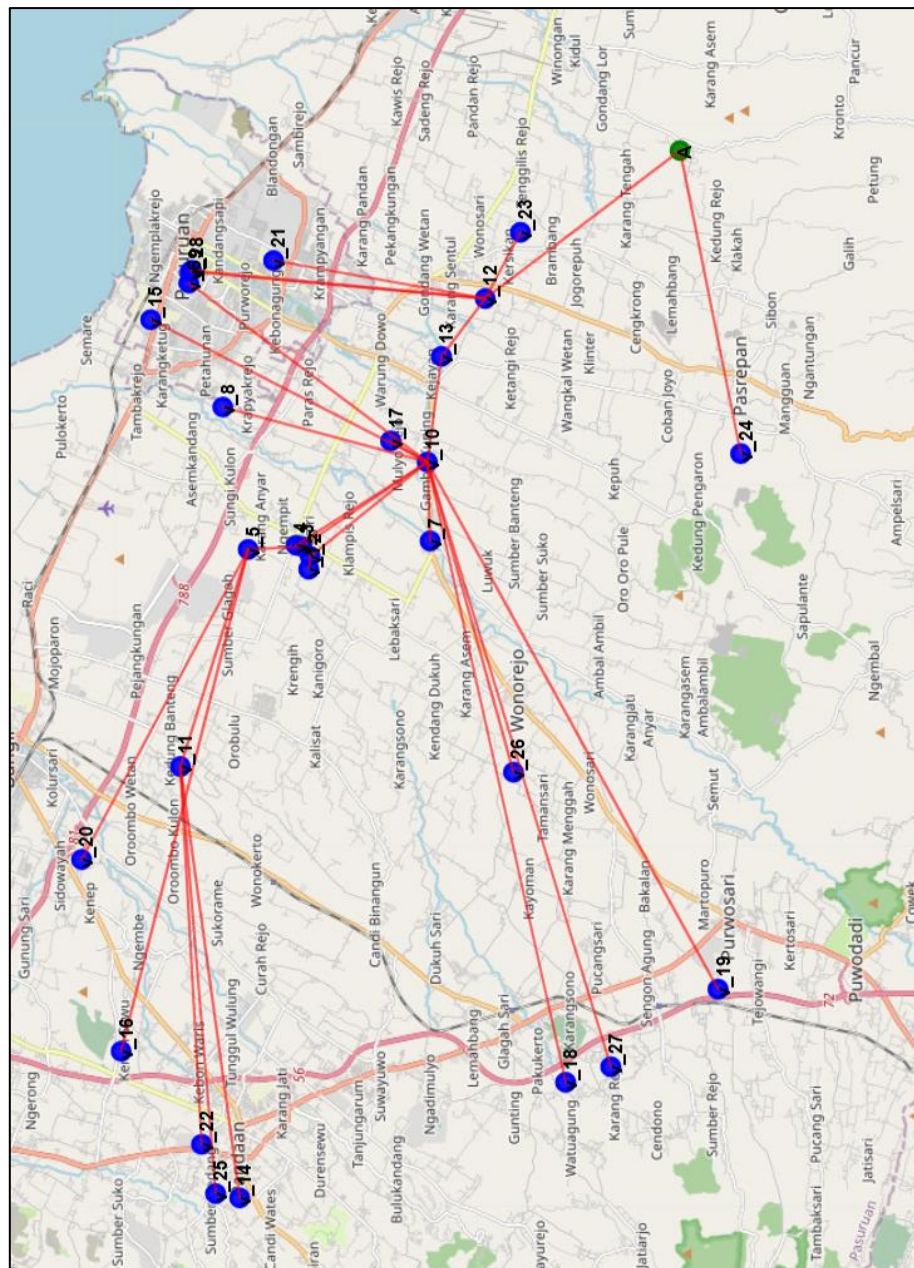
Setelah mengetahui rute terpendek semua titik, langkah berikutnya ialah membuat graf akhir yang menunjukkan rute terpendek dari titik awal kemudaaian kembali ke titik lainnya seperti $v_1, v_2, \dots, v_{28}$. Berikut

representasi graf yang di peroleh berdasarkan proses relaksasi dan memeriksa siklus negatif.



Gambar 4.4 Graf Rute Terpendek Tanpa Siklus *Hamiltonian Cycle*

Berikut ini adalah visualisasi graf dalam bentuk peta untuk memberikan gambaran lokasi yang lebih nyata.



Gambar 4.5 Visualisasi Graf Rute Terpendek Tanpa Siklus *Hamiltonian Cycle* pada Peta

4.3 Evaluasi Penelitian

Berdasarkan hasil penelitian mengenai penerapan siklus hamiltonian cycle pada Algoritma Bellman-Ford dan implementasi Algoritma Bellman-Ford dalam menentukan rute terpendek distribusi Air Mineral Santri di Pasuruan, berikut adalah beberapa aspek evaluasi terhadap penerapan *hamiltonian cycle* pada Algoritma Bellman-Ford dan implementasi Algoritma Bellman-Ford.

1. Memvalidasi Hasil Rute

Setelah dilakukan analisis terhadap hasil penerapan algoritma, validasi hasil rute menunjukkan bahwa Bellman-Ford dengan *Hamiltonian Cycle* berhasil melewati semua simpul dengan total jarak 97,48 KM, sedangkan Bellman-Ford tanpa *Hamiltonian* hanya menghasilkan jarak 3,49 KM tetapi tidak mencakup seluruh simpul. Hal ini membuktikan bahwa pendekatan tanpa *Hamiltonian Cycle* tidak sesuai untuk kebutuhan distribusi yang mengharuskan semua titik terlewati dalam satu perjalanan. Sehingga, kombinasi Bellman-Ford dengan *Hamiltonian Cycle* lebih sesuai untuk menyelesaikan masalah ini, meskipun berdampak pada peningkatan jarak tempuh.

2. Efisiensi Algoritma

Algoritma Bellman-Ford dengan *Hamiltonian Cycle* efektif untuk kasus yang membutuhkan perjalanan ke semua titik tanpa pengecualian. Sedangkan Algoritma tanpa siklus hamilton mencakup beberapa titik saja. Baik memakai siklus hamilton maupun tidak, keduanya mampu menangani bobot negatif tanpa siklus negatif sehingga memperoleh hasil yang optimal.

3. Kelemahan dan Kendala yang Ditemui

Algoritma Bellman-Ford tanpa *Hamiltonian Cycle* hanya mencari jalur terpendek kembali ke titik awal, sehingga tidak cocok untuk permasalahan yang membutuhkan kunjungan ke seluruh simpul. Pencarian *Hamiltonian Cycle* sendiri merupakan permasalahan NP-hard, yang berarti sulit diselesaikan dalam skala besar tanpa optimasi tambahan. Selain itu, perubahan dalam jaringan distribusi, seperti bertambahnya simpul

dikarenakan bertambahnya cabang baru pada Toko Basmalah atau perubahan bobot sisi, dapat mengharuskan perhitungan ulang secara manual, yang dapat menjadi kendala dalam implementasi di lapangan.

4. Potensi Perbaikan

Setelah dilakukan pendekatan hamiltonian cycle pada Algoritma Belman-Ford diperoleh rute terpendek melewati semua titik distribusi. Akan tetapi perlu waktu lebih lambat karena banyak proses iterasi. Untuk meningkatkan efisiensi algoritma tersebut, perlu dilakukan penelitian lebih lanjut terkait data yang sama dengan pendekatan lain seperti *dynamic programming* (pemecahan masalah dengan cara membagi masalah besar menjadi submasalah yang lebih kecil, menyimpan hasilnya, dan menggunakannya kembali untuk menghindari perhitungan ulang) atau menggunakan *parallel processing* (teknik komputasi di mana beberapa bagian dari suatu algoritma dijalankan secara bersamaan pada banyak prosesor atau core komputer, sehingga prosesnya menjadi lebih cepat) yang berpotensi meningkatkan kinerja algoritma dalam menangani graf berskala besar.

5. Penarikan Kesimpulan

Berdasarkan hasil evaluasi, Bellman-Ford dengan Hamiltonian Cycle merupakan metode yang paling sesuai untuk memastikan seluruh simpul dilewati dalam satu perjalanan, meskipun memiliki tantangan dalam efisiensi waktu. Bellman-Ford tanpa Hamiltonian tidak dapat digunakan untuk menjangkau siklus yg memuat semua titik karena hanya mencari rute terpendek tanpa mempertimbangkan semua simpul.

4.4 Integrasi Al-Qur'an dalam Penentuan Rute Terpendek Distribusi

Distribusi air mineral Santri merupakan bagian dari optimalisasi logistik yang bertujuan untuk mencapai efisiensi dalam pengiriman produk dari pabrik ke toko-toko Basmalah di Kota Pasuruan. Efisiensi ini selaras dengan prinsip yang diajarkan dalam QS. Al-A'raf ayat 31. Prinsip ini juga dapat diinterpretasikan dalam konteks distribusi barang, di mana penggunaan sumber daya seperti bahan bakar dan waktu harus dioptimalkan agar tidak terjadi pemborosan.

QS. An-Nahl ayat 90 menunjukkan bahwa Allah memerintahkan manusia untuk berlaku adil dan berbuat kebajikan, termasuk dalam sistem distribusi yang memastikan bahwa semua pihak mendapatkan manfaat yang adil. Penerapan Algoritma Bellman-Ford dalam pencarian rute terpendek distribusi air mineral Santri mencerminkan upaya untuk mencapai keadilan dalam distribusi, di mana setiap toko mendapatkan pasokan air dengan efisien, tanpa ada yang terabaikan atau mengalami ketidakseimbangan dalam distribusi.

Distribusi air mineral Santri dilakukan dengan tahapan-tahapan analisis data guna menentukan rute terpendek. Tahapan ini meliputi:

1. Pembuatan graf berarah berbobot berdasarkan jarak antar simpul.
2. Penentuan nilai optimal dengan Algoritma Bellman-Ford yang dimodifikasi dengan sifat Hamiltonian cycle.
3. Iterasi untuk memastikan semua titik dilewati satu kali sebelum kembali ke titik awal.
4. Memeriksa siklus negatif.

Menurut hasil perhitungan Algoritma Bellman-Ford dengan siklus *hamiltonian cycle*, rute optimal untuk distribusi air mineral Santri adalah

$$A \rightarrow v_{12} \rightarrow v_{13} \rightarrow v_{10} \rightarrow v_{17} \rightarrow v_7 \rightarrow v_1 \rightarrow v_2 \rightarrow v_5 \rightarrow v_4 \rightarrow v_3 \rightarrow v_8 \rightarrow v_{15} \rightarrow v_6$$

$$\rightarrow v_9 \rightarrow v_{28} \rightarrow v_{21} \rightarrow v_{23} \rightarrow v_{24} \rightarrow v_{26} \rightarrow v_{19} \rightarrow v_{27} \rightarrow v_{18} \rightarrow v_{14} \rightarrow v_{25} \rightarrow v_{22} \rightarrow$$

$$v_{16} \rightarrow v_{20} \rightarrow v_{11} \rightarrow A.$$

dengan total jarak 97,48 km. Sementara itu, jika siklus Hamiltonian tidak diterapkan maka rute dari awal A menuju v_{17} dengan melewati titik v_{12}, v_{13}, v_{10} . Dengan total jarak 3,49 km adalah rute terpendek.

Hadis ke-36 dalam kitab Arbai'in Nawawi juga menegaskan pentingnya mempermudah urusan orang lain. Penerapan algoritma pencarian rute terpendek dalam konteks distribusi adalah salah satu bentuk kemudahan bagi pihak pengelola dan konsumen. Sistem yang lebih efisien, menjadikan biaya operasional dapat ditekan, waktu pengiriman menjadi lebih cepat, dan pasokan air dapat sampai ke toko-toko Basmalah dengan lebih baik. Hadis tersebut mengajarkan bahwa memberikan kemudahan bagi sesama akan mendatangkan kemudahan pula dari Allah, baik di dunia maupun di akhirat.

Algoritma Bellman-Ford dalam distribusi air mineral Santri tidak hanya mencerminkan upaya efisiensi dalam dunia bisnis, tetapi juga memiliki nilai moral dan ajaran Islam yang kuat. Prinsip tidak berlebihan, keadilan dalam distribusi, serta memberikan kemudahan bagi orang lain menjadi landasan yang memperkuat praktik optimasi ini dalam kerangka yang lebih luas.

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian mengenai penerapan Algoritma Bellman-Ford dalam menentukan rute terpendek distribusi Air Mineral Santri di Pasuruan, dilakukan perbandingan antara implementasi algoritma dengan dan tanpa siklus Hamiltonian.

1. Hasil penelitian menunjukkan bahwa penerapan siklus Hamiltonian pada Algoritma Bellman-Ford memungkinkan distribusi melewati seluruh Toko Basmalah di Pasuruan dengan rute $A \rightarrow v_{12} \rightarrow v_{13} \rightarrow v_{10} \rightarrow v_{17} \rightarrow v_7 \rightarrow v_1 \rightarrow v_2 \rightarrow v_5 \rightarrow v_4 \rightarrow v_3 \rightarrow v_8 \rightarrow v_{15} \rightarrow v_6 \rightarrow v_9 \rightarrow v_{28} \rightarrow v_{21} \rightarrow v_{23} \rightarrow v_{24} \rightarrow v_{26} \rightarrow v_{19} \rightarrow v_{27} \rightarrow v_{18} \rightarrow v_{14} \rightarrow v_{25} \rightarrow v_{22} \rightarrow v_{16} \rightarrow v_{20} \rightarrow v_{11} \rightarrow A$. dengan total jarak tempuh 97,48 km. Algoritma Bellman-Ford dengan siklus Hamiltonian lebih sesuai, akurat, dan hasilnya lebih optimal dalam memenuhi kebutuhan distribusi secara menyeluruh. Hal ini berkaitan dengan permasalahan distribusi membutuhkan jalur yang melewati seluruh titik tujuan.
2. Algoritma Bellman-Ford yang diterapkan tanpa mempertimbangkan siklus Hamiltonian, akan menghasilkan rute terpendek yang hanya melewati dua. Berdasarkan Gambar 4.1, rute yang dihasilkan adalah awal A menuju v_{17} dengan melewati titik v_{12}, v_{13}, v_{10} dengan total jarak 3,49 km . Hal ini menunjukkan bahwa pada kasus ini banyak simpul atau titik distribusi yang tidak dapat dijangkau apabila pencarian rute terpendek memakai Algoritma Bellman-Ford tanpa siklus hamiltonian.

5.2 Saran

Peneliti mengharapkan agar hasil yang diperoleh dapat memberikan manfaat dalam optimalisasi sistem distribusi dan logistik, khususnya dalam menentukan rute terpendek untuk efisiensi transportasi di berbagai sektor industri.

Untuk penelitian selanjutnya, diharapkan dapat melakukan percobaan dengan algoritma shortest path lainnya, seperti A* atau Floyd-Warshall, guna mendapatkan perbandingan lebih komprehensif terkait keakuratan dan efisiensi perhitungan.

DAFTAR RUJUKAN

- Al-Qur'an dan Terjemahnya*. (2019). Kementerian Agama RI
- Bondy, J. A. (1972). Variations on the hamiltonian theme. *Canadian Mathematical Bulletin*, 15(1), 57–62.
- Br Ketaren, E. S. U., & Marpaung, F. (2023). *Penerapan Algoritma Bellman-Ford untuk Menentukan Lintasan Terpendek dalam Pendistribusian Barang pada PT. Global Jet Cargo (J&T Cargo)*.
<https://digilib.unimed.ac.id/id/eprint/58613/>
- Buhaerah, B., Busrah, Z., & Sanjaya, H. (2022). *Teori Graf dan Aplikasinya*. Living Spiritual Quotient. <http://repository.iainpare.ac.id/id/eprint/3489/>
- Chartrand, G., Lesniak, L., & Zhang, P. (2016). *Graphs & Digraphs: Sixth Edition*. CRC Press.
- Cormen, T. H. (Ed.). (2009). *Introduction to algorithms* (3rd ed). MIT Press.
- Darmalaksana, W. (2020). Metode penelitian kualitatif studi pustaka dan studi lapangan. *Pre-Print Digital Library UIN Sunan Gunung Djati Bandung*. <https://etheses.uinsgd.ac.id/id/eprint/32855>
- Fazri, M., & Puspita, R. (2015). Perencanaan Jumlah Distribusi Pemasaran Sebagai Pendukung Peningkatan Penjualan Produk Sumpit PT. Candi Kekal Jaya Co. Ltd. *Industrial Engineering Journal*, 4(1).
<https://journal.unimal.ac.id/miej/article/view/43>
- Fuadah, N., & Fauzi, R. M. Q. (2019). Eksternalitas pada Perusahaan Air Minum Santri Sidogiri Perspektif Ekonomi Sumber Daya Alam Islam. *Jurnal Ekonomi Syariah Teori Dan Terapan*, 6(5), 899–912.
- Hamdi, S., & Prihandoko, P. (2018). Analisis Algoritma Dijkstra dan Algoritma Bellman-Ford Sebagai Penentuan Jalur Terpendek Menuju Lokasi Kebakaran (Studi Kasus: Kecamatan Praya Kota). *Energy: Jurnal Ilmiah Ilmu-Ilmu Teknik*, 8(1), 26–32.
- Hasugian, P. M. (2015). Analisa dan implementasi algoritma bellman ford dalam mnentukan jalur terpendek pengantaran barang dalam kota. *Jurnal Mantik Penusa*, 18(2).
<http://download.garuda.kemdikbud.go.id/article.php?article=746812&val=11774&title=Analisa%20Dan%20Implementasi%20Algoritma%20Bellman%20Ford%20Dalam%20Mnentukan%20Jalur%20Terpendek%20Pengantaran%20Barang%20Dalam%20Kota>
- Hutasoit, E. T. H. (2019). Pencarian Rute Terpendek Menggunakan Algoritma Bellman-Ford (Studi Kasus: PT. JNE Medan). *Jurnal Sistem Komputer Dan Informatika (JSON)*, 1(1), 20–25.
- Khair, H. (2022). Alquran dan Hadits Sebagai Dasar Pendidikan Islam. *Darul Ulum: Jurnal Ilmiah Keagamaan, Pendidikan Dan Kemasyarakatan*, 13(1), 1–16.
- Morihito, R. V., Chungdinata, S. E., Nazareth, T. A., Pulukadang, M. I., Makalew,

- R. A., & Pinontoan, B. (2017). Identifikasi perubahan struktur dna terhadap pembentukan sel kanker menggunakan dekomposisi graf. *Jurnal Ilmiah Sains*, 153–160.
- Munib, A. (2018). Hukum Islam dan Muamalah (Asas-asas hukum Islam dalam bidang muamalah). *Al-Ulum Jurnal Pemikiran Dan Penelitian Ke Islaman*, 5(1), 72–80.
- Neshia, U., Baizal, Z. A., & Ummah, I. (2016). Pemrosesan Graf Berskala Besar Secara Paralel. *eProceedings of Engineering*, 3(1).
<https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/3652>
- Pambudi, R. B., & Hendratmi, A. (2020). Pengembangan bisnis oleh pondok pesantren Sidogiri Pasuruan. *Jurnal Ekonomi Syariah Teori Dan Terapan*. Vol. 7 No. 1 Januari 2020:, 7(1), 87–99.
- Retanto, Y. (2009). Algoritma Dijkstra dan Bellman-Ford dalam Pencarian Jalur Terpendek. *Bandung: ITB*.
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2009-2010/Makalah0910/MakalahStrukdis0910-082.pdf>
- SM, A. I., Ekawati, R., & Febriana, N. (2017). Optimalisasi rute distribusi air minum quelle dengan algoritma clarke & wright saving dan model vehicle routing problem. *Prosiding SENIATI*, 3(2), C1-1.

LAMPIRAN

Lampiran 1 Kode *Google Collab* untuk Membuat Graf Berarah dan Berbobot.

```
import pandas as pd
import networkx as nx
import matplotlib.pyplot as plt

# 1. Baca data matriks jarak dari file Excel
df = pd.read_excel('matriks_jarak_disesuaikan28.xlsx', index_col=0)

# 2. Daftar koordinat (longitude, latitude)
coords = {
    'A': (-7.7598876, 112.9327587),
    'v_1': (-7.6705829, 112.8310441),
    'v_2': (-7.6706883, 112.8341549),
    'v_3': (-7.6688276, 112.8363952),
    'v_4': (-7.6673642, 112.8368768),
    'v_5': (-7.6559664, 112.8357438),
    'v_6': (-7.6416277, 112.9007693),
    'v_7': (-7.6996056, 112.8377162),
    'v_8': (-7.6496078, 112.8703608),
    'v_9': (-7.6420066, 112.9033673),
    'v_10': (-7.6989441, 112.8570664),
    'v_11': (-7.6397052, 112.7829982),
    'v_12': (-7.7129144, 112.8969039),
    'v_13': (-7.7024837, 112.8827123),
    'v_14': (-7.653754, 112.6778982),
    'v_15': (-7.632513, 112.8916976),
    'v_16': (-7.6251352, 112.7136629),
    'v_17': (-7.6900767, 112.8622375),
    'v_18': (-7.7324973, 112.7059571),
    'v_19': (-7.7690225, 112.7287059),
    'v_20': (-7.6157069, 112.7603345),
    'v_21': (-7.6620254, 112.9061699),
    'v_22': (-7.6445605, 112.6907465),
    'v_23': (-7.721524, 112.9130406),
    'v_24': (-7.7744448, 112.8592286),
    'v_25': (-7.6478844, 112.6787975),
    'v_26': (-7.7199181, 112.7814007),
    'v_27': (-7.7431726, 112.7097226),
    'v_28': (-7.6423855, 112.9035835),
}

# 3. Konversi koordinat ke format (x, y) untuk plotting
# (longitude, latitude) -> (x, y)
pos = {k: (v[1], v[0]) for k, v in coords.items()}

# 4. Buat graf berbobot
G = nx.Graph()

# Tambahkan node
for node in df.index:
    G.add_node(node)

# Tambahkan edge berbobot dari matriks
for i, source in enumerate(df.index):
    for j, target in enumerate(df.columns):
        if i < j: # agar tidak double (graf tak berarah)
            weight = df.loc[source, target]
            if weight > 0:
                G.add_edge(source, target, weight=weight)

# 5. Visualisasi graf
plt.figure(figsize=(20, 14))

# Gambar node
nx.draw_networkx_nodes(G, pos, node_size=600, node_color='skyblue')
nx.draw_networkx_labels(G, pos, font_size=9)

# Gambar edge
nx.draw_networkx_edges(G, pos, width=1, alpha=0.7)

# Label bobot pada edge
edge_labels = {(u, v): f"{d['weight']:.2f}" for u, v, d in G.edges(data=True)}
nx.draw_networkx_edge_labels(G, pos, edge_labels=edge_labels, font_size=7)

plt.title("Graf Berbobot Berdasarkan Titik Koordinat dan Matriks Jarak", fontsize=16)
plt.axis('off')
plt.tight_layout()
plt.show()
```

Lampiran 2 Kode *Google Collab* untuk Membuat Graf Rute Terpendek dengan Sifat *Hamiltonian Cycle*.

a. Graf rute terpendek dengan siklus *hamiltonian cycle*

```
import matplotlib.pyplot as plt
import math

# Data koordinat lengkap 29 titik
data = {
    "A": (-7.7598876, 112.9327587),
    "v_1": (-7.6705829, 112.8310441),
    "v_2": (-7.6706883, 112.8341549),
    "v_3": (-7.6688276, 112.8363952),
    "v_4": (-7.6673642, 112.8368768),
    "v_5": (-7.6559664, 112.8357438),
    "v_6": (-7.6416277, 112.9007693),
    "v_7": (-7.6996056, 112.8377162),
    "v_8": (-7.6496078, 112.8703608),
    "v_9": (-7.6420066, 112.9033673),
    "v_10": (-7.6989441, 112.8570664),
    "v_11": (-7.6397052, 112.7829982),
    "v_12": (-7.7129144, 112.8969039),
    "v_13": (-7.7024837, 112.8827123),
    "v_14": (-7.653754, 112.6778982),
    "v_15": (-7.632513, 112.8916976),
    "v_16": (-7.6251352, 112.7136629),
    "v_17": (-7.6900767, 112.8622375),
    "v_18": (-7.7324973, 112.7059571),
    "v_19": (-7.7690225, 112.7287059),
    "v_20": (-7.6157069, 112.7603345),
    "v_21": (-7.6620254, 112.9061699),
    "v_22": (-7.6445605, 112.6907465),
    "v_23": (-7.721524, 112.9130406),
    "v_24": (-7.7744448, 112.8592286),
    "v_25": (-7.6478844, 112.6787975),
    "v_26": (-7.7199181, 112.7814007),
    "v_27": (-7.7431726, 112.7097226),
    "v_28": (-7.6423855, 112.9035835),
}
```

```
priority_adjustments = {
    "v_1": 0.5,
    "v_5": 1.5,
    "v_10": 2.5,
    "v_14": 3,
    "v_12": 2,
}

# Fungsi jarak haversine dengan pengaturan pengurangan bobot
def haversine(coord1, coord2, u, v):
    R = 6371 # Radius bumi dalam km
    lat1, lon1 = coord1
    lat2, lon2 = coord2
    dlat = math.radians(lat2 - lat1)
    dlon = math.radians(lon2 - lon1)
    a = math.sin(dlat/2)**2 + math.cos(math.radians(lat1)) * math.cos(math.radians(lat2)) * math.sin(dlon/2)**2
    distance = R * (2 * math.atan2(math.sqrt(a), math.sqrt(1 - a)))

    # Sesuaikan jarak jika titik dalam priority_adjustments
    if u in priority_adjustments:
        distance -= priority_adjustments[u]
    if v in priority_adjustments:
        distance -= priority_adjustments[v]

    return max(distance, 0.1) # Hindari jarak negatif

# List rute
route = ["A", "v_12", "v_13", "v_10", "v_17", "v_7", "v_1", "v_2", "v_5", "v_4", "v_3", "v_8", "v_15", "v_6",
        "v_9", "v_28", "v_21", "v_23", "v_24", "v_26", "v_19", "v_27", "v_18", "v_14", "v_25", "v_22", "v_16",
        "v_20", "v_11", "A"]

# Buat directed graph
G = nx.DiGraph()

# Tambah node dengan posisi koordinat
for name, coord in data.items():
    G.add_node(name, pos=(coord[1], coord[0])) # (lon, lat) supaya tampil benar
```

```

# Tambah edges + bobot jarak
for i in range(len(route)-1):
    u, v = route[i], route[i+1]
    weight = haversine(data[u], data[v], u, v) # Menghitung jarak antar titik dengan pengurangan bobot
    G.add_edge(u, v, weight=round(weight, 2))

# Visualisasi
pos = nx.get_node_attributes(G, 'pos')
edge_labels = nx.get_edge_attributes(G, 'weight')

plt.figure(figsize=(10,8))
nx.draw(G, pos, with_labels=True, node_size=500, node_color='lightblue', arrows=True)
nx.draw_networkx_edge_labels(G, pos, edge_labels=edge_labels)
plt.show()

```

b. Visualisasi Graf rute terpendek dengan siklus *hamiltonial cycle* pada peta

```

import folium
import networkx as nx
# Gunakan rute hasil dari algoritma
route = ["A", "v_12", "v_13", "v_10", "v_17", "v_7", "v_1", "v_2", "v_5", "v_4", "v_3",
        "v_8", "v_15", "v_6", "v_9", "v_28", "v_21", "v_23", "v_24", "v_26", "v_19",
        "v_27", "v_18", "v_14", "v_25", "v_22", "v_16", "v_20", "v_11", "A"]

# Titik tengah peta (bisa dari titik awal 'A')
map_center = data["A"]

# Buat peta
m = folium.Map(location=map_center, zoom_start=11)

# Tambahkan marker ke peta
for name, coord in data.items():
    color = 'green' if name == 'A' else 'blue'
    folium.CircleMarker(
        location=coord,
        radius=6,
        color=color,
        fill=True,
        fill_color=color,
        fill_opacity=0.9
    ).add_to(m)

    folium.Marker(
        location=coord,
        icon=folium.DivIcon(html=f"<div style='font-size:10pt; font-weight:bold'>{name}</div>")
    ).add_to(m)

# Tambahkan garis rute
route_coords = [data[node] for node in route]
folium.PolyLine(
    locations=route_coords,
    color='red',
    weight=4,
    opacity=0.7
).add_to(m)

```

```

# Simpan ke HTML
m.save("graf_rute_berarah.html")

from google.colab import files
files.download("graf_rute_berarah.html")

```

Lampiran 3 Kode *Google Collab* Membuat Graf Rute Terpendek dari **A** ke titik lainnya

a. Graf rute terpendek dengan algoritma Bellman-Ford

```

import networkx as nx
import matplotlib.pyplot as plt

# Data koordinat node (latitude, longitude)
pos = {
    'A': (-7.7598876, 112.9327587),
    'v_1': (-7.6705829, 112.8310441),
    'v_2': (-7.6706883, 112.8341549),
    'v_3': (-7.6688276, 112.8363952),
    'v_4': (-7.6673642, 112.8368768),
    'v_5': (-7.6559664, 112.8357438),
    'v_6': (-7.6416277, 112.9007693),
    'v_7': (-7.6996056, 112.8377162),
    'v_8': (-7.6496078, 112.8703608),
    'v_9': (-7.6420066, 112.9033673),
    'v_10': (-7.6989441, 112.8570664),
    'v_11': (-7.6397052, 112.7829982),
    'v_12': (-7.7129144, 112.8969039),
    'v_13': (-7.7024837, 112.8827123),
    'v_14': (-7.653754, 112.6778982),
    'v_15': (-7.632513, 112.8916976),
    'v_16': (-7.6251352, 112.7136629),
    'v_17': (-7.6900767, 112.8622375),
    'v_18': (-7.7324973, 112.7059571),
    'v_19': (-7.7690225, 112.7287059),
    'v_20': (-7.6157069, 112.7603345),
    'v_21': (-7.6620254, 112.9061699),
    'v_22': (-7.6445605, 112.6907465),
    'v_23': (-7.721524, 112.9130406),
    'v_24': (-7.7744448, 112.8592286),
    'v_25': (-7.6478844, 112.6787975),
    'v_26': (-7.7199181, 112.7814007),
    'v_27': (-7.7431726, 112.7097226),
    'v_28': (-7.6423855, 112.9035835)
}

# Buat graf berarah
G = nx.DiGraph()
G.add_nodes_from(pos.keys())

# Tambahkan edge dengan bobot
edges = [
    ('A', 'v_12', 4.55),
    ('v_12', 'v_10', 2.66),
    ('v_10', 'v_3', 1.55),
    ('v_3', 'v_1', 0.04),
    ('v_10', 'v_2', 1.53),
    ('v_10', 'v_4', 1.66),
    ('v_2', 'v_5', 0.15),
    ('v_5', 'v_11', 4.59),
    ('v_11', 'v_14', 8.69),
    ('v_5', 'v_16', 12.38),
    ('v_10', 'v_18', 14.56),
    ('v_10', 'v_19', 13.65),
    ('v_5', 'v_20', 7.94),
    ('v_5', 'v_22', 14.53),
    ('v_5', 'v_25', 15.82),
    ('v_10', 'v_26', 6.16),
    ('v_10', 'v_27', 14.46),
    ('v_12', 'v_6', 5.94),
    ('v_12', 'v_7', 4.69),
    ('v_12', 'v_8', 5.62),
    ('v_12', 'v_9', 5.92),
    ('v_12', 'v_13', max(0.01, 4.50 - 4.55)), # bobot negatif disesuaikan
    ('v_12', 'v_15', 6.96),
    ('v_12', 'v_17', 2.59),
    ('v_12', 'v_21', 3.75),
    ('v_12', 'v_23', 1.02),
    ('A', 'v_24', 8.26),
    ('v_12', 'v_28', 5.88)
]

# Koreksi bobot negatif menjadi nilai kecil positif
edges = [(u, v, max(w, 0.01)) for u, v, w in edges]
G.add_weighted_edges_from(edges)

# Buat posisi untuk plotting (longitude = X, latitude = Y)
pos_plot = {node: (lon, lat) for node, (lat, lon) in pos.items()}

```

```

# Plot graf sesuai posisi geografis
plt.figure(figsize=(14, 10))
nx.draw_networkx_nodes(G, pos_plot, node_size=300, node_color='lightblue')
nx.draw_networkx_labels(G, pos_plot, font_size=10)

# Gambar edge dengan panah
nx.draw_networkx_edges(G, pos_plot, edgelist=G.edges(), arrowstyle='->', arrowsize=20, edge_color='black')

# Tampilkan bobot di atas edge
edge_labels = nx.get_edge_attributes(G, 'weight')
nx.draw_networkx_edge_labels(G, pos_plot, edge_labels={e: f'{w:.2f}' for e, w in edge_labels.items()}, font_color='red', font_size=10)

# Set tampilan
plt.xlabel("Longitude")
plt.ylabel("Latitude")
plt.title("Graf Berarah dengan Posisi Node Sesuai Koordinat Geografis")
plt.grid(True)
plt.gca().set_aspect('equal', adjustable='datalim')
plt.show()

```

b. Visualisasi graf rute terpendek dengan algoritma Bellman-Ford pada peta

```

import folium
import networkx as nx

# Data koordinat node (latitude, longitude)
pos = {
    'A': (-7.7598876, 112.9327587),
    'v_1': (-7.6705829, 112.8310441),
    'v_2': (-7.6706883, 112.8341549),
    'v_3': (-7.6688276, 112.8363952),
    'v_4': (-7.6673642, 112.8368768),
    'v_5': (-7.6559664, 112.8357438),
    'v_6': (-7.6416277, 112.9007693),
    'v_7': (-7.6996056, 112.8377162),
    'v_8': (-7.6496078, 112.8703608),
    'v_9': (-7.6420066, 112.9033673),
    'v_10': (-7.6989441, 112.8570664),
    'v_11': (-7.6397052, 112.7829982),
    'v_12': (-7.7129144, 112.8969039),
    'v_13': (-7.7024837, 112.8827123),
    'v_14': (-7.653754, 112.6778982),
    'v_15': (-7.632513, 112.8916976),
    'v_16': (-7.6251352, 112.7136629),
    'v_17': (-7.6900767, 112.8622375),
    'v_18': (-7.7324973, 112.7059571),
    'v_19': (-7.7690225, 112.7287059),
    'v_20': (-7.6157069, 112.7603345),
    'v_21': (-7.6620254, 112.9061699),
    'v_22': (-7.6445605, 112.6907465),
    'v_23': (-7.721524, 112.9130406),
    'v_24': (-7.7744448, 112.8592286),
    'v_25': (-7.6478844, 112.6787975),
    'v_26': (-7.7199181, 112.7814007),
    'v_27': (-7.7431726, 112.7097226),
    'v_28': (-7.6423855, 112.9035835)
}

```

```

# Edge data
edges = [
    ('A', 'v_12', 4.55),
    ('v_12', 'v_10', 2.66),
    ('v_10', 'v_3', 1.55),
    ('v_3', 'v_1', 0.04),
    ('v_10', 'v_2', 1.53),
    ('v_10', 'v_4', 1.66),
    ('v_2', 'v_5', 0.15),
    ('v_5', 'v_11', 4.59),
    ('v_11', 'v_14', 8.69),
    ('v_5', 'v_16', 12.38),
    ('v_10', 'v_18', 14.56),
    ('v_10', 'v_19', 13.65),
    ('v_5', 'v_20', 7.94),
    ('v_5', 'v_22', 14.53),
    ('v_5', 'v_25', 15.82),
    ('v_10', 'v_26', 6.16),
    ('v_10', 'v_27', 14.46),
    ('v_12', 'v_6', 5.94),
    ('v_12', 'v_7', 4.69),
    ('v_12', 'v_8', 5.62),
    ('v_12', 'v_9', 5.92),
    ('v_12', 'v_13', 0.01), # Koreksi bobot negatif
    ('v_12', 'v_15', 6.96),
    ('v_12', 'v_17', 2.59),
    ('v_12', 'v_21', 3.75),
    ('v_12', 'v_23', 1.02),
    ('A', 'v_24', 8.26),
    ('v_12', 'v_28', 5.88)
]

# Buat graf
G = nx.DiGraph()
for u, v, w in edges:
    G.add_edge(u, v, weight=w)

```



```

# Peta pusat di titik A
center_lat, center_lon = pos['A']
m = folium.Map(location=[center_lat, center_lon], zoom_start=12, control_scale=True)

# Tambahkan node + label
for name, coord in pos.items():
    folium.CircleMarker(
        location=coord,
        radius=6,
        color='green' if name == 'A' else 'blue',
        fill=True,
        fill_color='green' if name == 'A' else 'blue',
        fill_opacity=0.9
    ).add_to(m)
# Tambahkan label langsung tampil
folium.Marker(
    location=coord,
    icon=folium.DivIcon(
        html=f"""<div style="font-size:10pt; font-weight:bold; color:black">{name}</div>"""
    )
).add_to(m)

# Tambahkan edge (panah dengan bobot)
for u, v, data in G.edges(data=True):
    lat_u, lon_u = pos[u]
    lat_v, lon_v = pos[v]
    weight = data['weight']
    folium.PolyLine(
        locations=[(lat_u, lon_u), (lat_v, lon_v)],
        color='red',
        weight=2,
        opacity=0.6,
        tooltip=f"{u} → {v}: {weight:.2f}"
    ).add_to(m)

# Simpan ke HTML
m.save("graf_dengan_label.html")

```

Lampiran 4 Mencari rute terpendek dengan Algoritma Bellman-Ford memakai bahasa pemrograman *python*

```

import pandas as pd
import networkx as nx

# 1. Baca matriks jarak
df = pd.read_excel('matriks_jarak_disesuaikan28.xlsx', index_col=0)

# 2. Buat graf berarah berbobot
G = nx.DiGraph()
for source in df.index:
    for target in df.columns:
        if source != target:
            weight = df.loc[source, target]
            if pd.notna(weight): # pastikan bukan NaN
                # Jika bobot negatif, ubah menjadi 0.1
                adjusted_weight = weight if weight > 0 else 0.1
                G.add_edge(source, target, weight=adjusted_weight)

# 3. Jalankan algoritma Bellman-Ford dari node 'A'
start_node = 'A'
lengths, paths = nx.single_source_bellman_ford(G, start_node, weight='weight')

# 4. Tampilkan hasil
print(f"Rute terpendek dari {start_node} ke semua node lain:")
for target in df.columns:
    if target != start_node:
        if target in lengths:
            print(f"- Ke {target}:")
            print(f"    Jarak: {lengths[target]:.2f}")
            print(f"    Jalur: {' -> '.join(paths[target])}")
        else:
            print(f"- Ke {target}: Tidak dapat dijangkau.")

```

RIWAYAT HIDUP



Sulastri, lahir di Kabupaten Kediri pada 9 Maret 2003. Penulis Merupakan anak terakhir dari enam bersaudara, dari pasangan Bapak Suyani dan Ibu Maryam. Penulis dibesarkan di Dusun Sumberjati, Desa Bedali, Kecamatan Ngancar, Kabupaten Kediri.

Penulis menempuh pendidikan di RA Baitul Hikmah Kecamatan Ngancar, Kabupaten Kediri dan lulus pada tahun 2009, selanjutnya di MI Baitul Hikmah dan lulus pada tahun 2015, setelah itu penulis melanjutkan pendidikan di Mts SA Jabal Nuur Kecamatan Wates, Kabupaten Kediri dan lulus pada tahun 2021, MA Ma'arif Kecamatan Udanawu, Kabupaten Blitar dan lulus pada tahun 2021. Kemudian, penulis melanjutkan pendidikan di Universitas Islam Negeri Maulana Malik Ibrahim Malang dan diterima di Program Studi Matematika Fakultas Sains dan Teknologi pada tahun 2021 melalui jalur SNMPTN.

Selama menjalani masa kuliah, penulis memilih untuk memfokuskan diri pada kegiatan akademik, baik di lingkungan kampus maupun di pondok pesantren. Hari-hari penulis diisi dengan aktivitas yang berpusat di dua tempat tersebut. Di kampus, penulis tidak aktif dalam organisasi kemahasiswaan, namun tetap berkomitmen penuh pada perkuliahan dan tugas-tugas akademik. Sementara itu, di pondok pesantren, penulis dipercaya menjadi salah satu pengurus di divisi taklim, yang berkaitan kegiatan keilmuan santri.



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 : Sulastri
NIM : 210601110002
Fakultas / Jurusan : Sains dan Teknologi / Matematika
Judul Skripsi : Penerapan Algoritma Bellman-Ford Pada Rute Terpendek Distribusi Air Mineral Santri di Pasuruan
Pembimbing I : Prof. Dr. H. Turmudi, M.Si., Ph. D
Pembimbing II : Juhari, M.Si

No	Tanggal	Hal	Tanda Tangan
1.	3 Oktober 2024	Konsultasi Bab I	1.
2.	10 Oktober 2024	Konsultasi Bab I, II, dan III	2.
3.	14 Oktober 2024	Konsultasi Bab I, II, dan III	3.
4.	17 Oktober 2024	ACC Bab I, II, dan III	4.
5.	18 Oktober 2024	ACC Kajian Agama Bab I dan II	5.
6.	24 Oktober 2024	ACC Seminar Proposal	6.
7.	26 Februari 2025	Konsultasi Revisi Seminar Proposal	7.
8.	27 Maret 2025	Konsultasi Kajian Agama Bab IV	8.
9.	7 Maret 2025	Konsultasi Bab IV dan V	9.
10.	8 Maret 2025	ACC Kajian Agama Bab IV	10.
11.	10 Maret 2025	ACC Bab IV dan V	11.
12.	10 Maret 2025	ACC Seminar Hasil	12.
13.	2 Mei 2025	Konsultasi Revisi Seminar Hasil	13.
14.	7 Mei 2025	Konsultasi Revisi Seminar Hasil	14.
15.	15 Mei 2025	ACC Sidang Skripsi	15.



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

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

16.	10 Juni 2025	ACC Keseluruhan	16.
-----	--------------	-----------------	-----

Malang, 10 Juni 2025

Mengetahui,

Ketua Program Studi Matematika

Dr. Elly Susanti, M.Sc.

NIP. 19741129 200012 2 005