

Optimasi rute transit penumpang trans metro deli antar-koridor menggunakan algoritma dijkstra pada pemodelan graf berarah

Alfin Syahri^{1*}, Muhammad Budi Akbar¹, Suvriadi Panggabean²

¹ Ilmu Komputer, FMIPA, Universitas Negeri Medan, Medan

² Matematika, FMIPA, Universitas Negeri Medan, Medan

alfinsyahri1025@mhs.unimed.ac.id

Diterima: 15-06-2026; Direvisi: 30-09-2026; Dipublikasi: 30-09-2026

Abstract

This study applies graph theory to model the Trans Metro Deli network and determine the minimum weighted path using Dijkstra's Algorithm. The transportation network is represented as a weighted directed graph $G=(V,E)$, where vertices represent bus stops, edges represent direct connections between bus stops, and the weight $w(u,v)$ represents the estimated travel time in minutes. Data was obtained from BusListrikMedan and Google Maps, then implemented using Python with the NetworkX library. The modeling results produce a weighted directed graph consisting of 236 vertices and 245 edges representing the five main corridors of Trans Metro Deli. Tests were conducted on five travel scenarios to analyze the path with the minimum total weight. The results show that Dijkstra's Algorithm successfully determines the minimum weighted path in all test scenarios. The smallest minimum weight was obtained on the Amplas Terminal Bus Stop–Juanda Bus Stop route at 22 minutes, while the largest minimum weight was obtained on the Bandar Khalipah Station Bus Stop–Pinang Baris Terminal Bus Stop route at 73 minutes. Validation against Google Maps resulted in a time difference of 2–4 minutes, which indicates that the graph model built is able to represent the connectivity of the Trans Metro Deli network well.

Keywords: Dijkstra's algorithm, weighted directed graph, route optimization, shortest path, Trans Metro Deli, public transportation, graph theory.

Abstrak

Penelitian ini menerapkan teori graf untuk memodelkan jaringan Trans Metro Deli dan menentukan lintasan berbobot minimum menggunakan Algoritma Dijkstra. Jaringan transportasi direpresentasikan sebagai graf berarah berbobot $G=(V,E)$, dengan vertex merepresentasikan halte, edge merepresentasikan hubungan langsung antarhalte, dan bobot $w(u,v)$ menyatakan estimasi waktu tempuh dalam satuan menit. Data diperoleh dari BusListrikMedan dan Google Maps, kemudian diimplementasikan menggunakan Python dengan library NetworkX. Hasil pemodelan menghasilkan graf berarah berbobot yang terdiri dari 236 vertex dan 245 edge yang merepresentasikan lima koridor utama Trans Metro Deli. Pengujian dilakukan pada lima skenario perjalanan untuk menganalisis lintasan dengan total bobot minimum. Hasil menunjukkan bahwa Algoritma Dijkstra berhasil menentukan lintasan berbobot minimum pada seluruh skenario pengujian. Bobot minimum terkecil diperoleh pada rute Halte Terminal Amplas–Halte Juanda sebesar 22 menit, sedangkan bobot minimum terbesar diperoleh pada rute Halte Stasiun Bandar Khalipah–Halte Terminal Pinang Baris sebesar 73 menit. Validasi terhadap Google Maps menghasilkan selisih waktu 2–4 menit, yang menunjukkan bahwa model graf yang dibangun mampu merepresentasikan konektivitas jaringan Trans Metro Deli dengan baik.

Kata Kunci: Algoritma Dijkstra, graf berarah berbobot, optimasi rute, shortest path, Trans Metro Deli, transportasi publik, teori graf.

1. PENDAHULUAN

Trans Metro Deli merupakan sistem transportasi publik berbasis bus listrik yang melayani wilayah Kota Medan dan sekitarnya melalui lima koridor utama, yaitu Koridor 1 (Amplas–Pinang Baris), Koridor 2 (J City–Lapangan Merdeka), Koridor 3 (Belawan–Lapangan Merdeka), Koridor 4 (Tuntungan–Lapangan Merdeka), dan Koridor 5 (Tembung–Lapangan Merdeka). Sistem ini didukung oleh total 60 unit bus dan beroperasi setiap hari dengan jam layanan yang berbeda pada setiap koridor, yaitu mulai pukul 05.00–06.00 WIB hingga pukul 19.15–21.00 WIB. Selain itu, Trans Metro Deli menyediakan fasilitas transit antar-koridor dengan waktu transit gratis selama 75 menit menggunakan kartu pembayaran. Jaringan halte yang saling terhubung melalui beberapa titik transit, seperti Halte Lapangan Merdeka dan Halte Stasiun Medan, menjadikan Trans Metro Deli sesuai untuk dimodelkan sebagai graf berarah berbobot dan dianalisis menggunakan Algoritma Dijkstra guna menentukan rute perjalanan dengan waktu tempuh minimum.

Algoritma Dijkstra merupakan algoritma yang digunakan untuk mencari jalur terpendek pada graf berbobot dengan nilai bobot positif. Algoritma ini bekerja dengan memilih simpul yang memiliki jarak terkecil dari titik awal, kemudian memperbarui jarak ke simpul lainnya secara bertahap hingga ditemukan rute paling lintasan dengan total bobot minimum menuju tujuan. Dalam penerapannya pada sistem transportasi dan navigasi, algoritma Dijkstra mampu menentukan rute perjalanan yang efisien berdasarkan jarak atau waktu tempuh. Selain itu, algoritma ini dikenal memiliki tingkat akurasi yang baik dalam menyelesaikan permasalahan pencarian jalur terpendek pada jaringan jalan maupun transportasi umum (Pasaribu et al., 2026), (Amelia et al., 2024).

Algoritma Dijkstra banyak diterapkan dalam sistem transportasi untuk menentukan rute tercepat secara efisien. Penelitian oleh Noval Rafael I dan rekan-rekan menunjukkan bahwa algoritma Dijkstra mampu mencari jalur lintasan dengan total bobot minimum pada jaringan transportasi lokal dengan hasil yang hampir sama dengan Google Maps. Implementasi dilakukan menggunakan Python dan data jarak dari Google Maps, sehingga algoritma ini dinilai efektif untuk mendukung navigasi dan pengelolaan transportasi (I et al., 2025).

Pertumbuhan kota dan peningkatan mobilitas masyarakat menuntut sistem transportasi yang lebih efisien, terencana, dan berbasis analisis ilmiah. Permasalahan kemacetan, jarak tempuh yang tidak efisien, serta sulitnya menentukan jalur transit terbaik menjadi tantangan nyata di kota besar seperti Medan. Pendekatan matematis melalui teori graf memungkinkan jaringan transportasi dimodelkan sebagai simpul (halte/persimpangan) dan sisi (jalur penghubung) sehingga dapat dianalisis secara sistematis untuk menemukan jalur lintasan dengan total bobot minimum. Teori ini terbukti mampu mengidentifikasi jalur alternatif, titik kemacetan, dan kebutuhan peningkatan kapasitas dalam perencanaan transportasi kota (Agusnur, 2025).

Dalam penerapan praktisnya, graf yang digunakan adalah graf berarah berbobot, di mana setiap sisi memiliki bobot berupa jarak atau waktu tempuh. Model ini sangat sesuai untuk merepresentasikan rute kendaraan yang memiliki arah perjalanan tertentu. Algoritma yang paling banyak digunakan dalam pencarian jalur terpendek pada graf jenis ini adalah algoritma Dijkstra, yang bekerja dengan strategi greedy memilih sisi berbobot terkecil secara bertahap hingga mencapai tujuan akhir. Konsep ini telah banyak diterapkan dalam studi navigasi dan pemetaan rute karena efektivitasnya dalam menyelesaikan permasalahan jalur terpendek (Wahyudi et al., 2024).

Keandalan algoritma Dijkstra telah dibuktikan dalam berbagai studi kasus nyata di Kota Medan, seperti pencarian rute terpendek dari Fakultas Kedokteran UNIMED menuju gerbang kampus dan dari Universitas Negeri Medan menuju Museum Negeri Sumatera Utara. Hasil simulasi menunjukkan bahwa algoritma ini mampu menghasilkan rute paling lintasan dengan total bobot minimum berdasarkan perhitungan bobot jarak pada graf, bahkan disertai pseudocode, implementasi Python, serta visualisasi graf berbobot. Hal ini menegaskan bahwa Dijkstra tidak hanya kuat secara teori, tetapi juga sangat aplikatif dalam sistem navigasi berbasis peta (Simbolon et al., 2025).

Penelitian lain juga menunjukkan penerapan algoritma Dijkstra dalam menentukan rute terpendek dari Stasiun Kereta Api Medan menuju enam kampus di Medan Estate. Studi tersebut memodelkan jaringan jalan ke dalam adjacency list, menerapkan rumus $\text{distance}(v) = \min(\text{distance}(v), \text{distance}(u) + \text{weight}(u, v))$, dan membuktikan efisiensi algoritma dalam meminimalkan jarak serta waktu tempuh. Temuan ini memperkuat bahwa pemodelan graf berarah dengan Dijkstra sangat relevan untuk diterapkan pada sistem transit antar koridor seperti Trans Metro Deli, yang memiliki banyak simpul halte dan jalur penghubung (Arsyad et al., 2025).

Selain itu, penelitian oleh Sunardi, Anton Yudhana, dan Ahmad Azhar Kadim menerapkan algoritma Dijkstra pada aplikasi TransJogja berbasis Android. Hasil penelitian menunjukkan bahwa algoritma ini dapat menentukan rute bus dan halte dengan waktu proses yang sangat cepat serta lebih baik dibandingkan algoritma Ant Colony. Aplikasi juga dilengkapi teknologi GIS untuk menampilkan posisi halte dan bus secara real-time sehingga memudahkan pengguna dalam menggunakan transportasi umum (Yudhana et al., 2019).

Penelitian terdahulu menunjukkan bahwa algoritma Dijkstra efektif digunakan untuk mencari rute terpendek pada sistem transportasi. Penelitian oleh Purba dan rekan-rekan berhasil menerapkan algoritma Dijkstra untuk menentukan jalur tercepat dari SMA Negeri 17 Medan menuju Universitas Katolik Santo Thomas dengan hasil yang hampir sama dengan Google Maps. Hal ini membuktikan bahwa algoritma Dijkstra mampu memberikan rute yang akurat dan efisien pada sistem navigasi berbasis graf (Pasaribu et al., 2026). Selain itu, penelitian oleh Hidayah, Samsudin, dan Triase menerapkan algoritma Dijkstra pada aplikasi transportasi online di Kota Medan dan menunjukkan

bahwa algoritma ini dapat membantu menentukan rute perjalanan secara lintasan dengan total bobot minimum melalui perhitungan graf berbobot (Hidayah, 2021).

Penelitian lain oleh Meliala dan rekan-rekan membandingkan algoritma Dijkstra dan A* dalam optimasi rute transportasi perkotaan. Hasil penelitian menunjukkan bahwa algoritma Dijkstra lebih cocok digunakan pada kondisi jaringan yang stabil, sedangkan algoritma A* lebih efektif untuk kondisi lalu lintas yang berubah secara real-time. Penelitian ini membuktikan bahwa algoritma graf dapat meningkatkan efisiensi sistem transportasi dan membantu pengguna memperoleh rute perjalanan yang lebih cepat dan efektif (Amelia et al., 2024).

Implementasi algoritma Dijkstra pada studi kasus nyata menunjukkan bahwa algoritma ini efektif dalam menentukan rute terpendek pada berbagai bidang. Penelitian Helena Devi Ariyani dkk. menerapkan algoritma Dijkstra pada jaringan jalan antar Kampus Polimarin dan menghasilkan rute paling lintasan dengan total bobot minimum dengan jarak 25,80 km (Devi et al., 2026). Sementara itu, penelitian Anastasya Putri Wibowo dkk. menggunakan algoritma Dijkstra untuk menentukan rute tercepat dari RS Bina Kasih menuju RS Adam Malik guna mempercepat proses rujukan pasien. Kedua penelitian tersebut membuktikan bahwa algoritma Dijkstra mampu memilih jalur dengan bobot minimum secara efektif dan efisien (Anastasya Putri Wibowo, 2025).

Meskipun algoritma Dijkstra telah banyak diterapkan pada berbagai permasalahan pencarian lintasan terpendek, penelitian yang secara khusus memodelkan jaringan halte Trans Metro Deli dalam bentuk graf berarah berbobot masih terbatas. Selain itu, belum banyak kajian yang memanfaatkan estimasi waktu tempuh antarhalte sebagai bobot untuk menganalisis rute perjalanan pada jaringan transportasi publik di Kota Medan. Oleh karena itu, penelitian ini bertujuan membangun model graf berarah berbobot berdasarkan jaringan halte Trans Metro Deli dan menerapkan Algoritma Dijkstra untuk menentukan lintasan berbobot minimum berdasarkan estimasi waktu tempuh perjalanan antarhalte.

2. METODE PENELITIAN

Metodologi penelitian implementasi algoritma Dijkstra meliputi pengumpulan data, pemodelan graf berarah, penerapan algoritma, pengujian, dan analisis hasil. (Rufus et al., 2024) menggunakan graf berbobot untuk rute pengiriman terpendek. (Yusnita et al., 2021) menerapkan observasi, wawancara, dan metode prototype. (Arta & Victoria, 2023) mengembangkan pemetaan UMKM berbasis Android. (Ginasta, 2024) menggabungkan Dijkstra dan Best First Search untuk e-parking.

3.1 Jenis Penelitian

Penelitian terapan kuantitatif dengan tahapan: pengumpulan data, pemodelan graf berarah, implementasi algoritma Dijkstra, pengujian, dan analisis hasil mengembangkan sistem berbasis web untuk rute agen rongsokan dengan pemodelan graf berbobot jarak. (Taneo et al., 2025) melakukan eksperimen membandingkan Dijkstra dan Greedy BFS

pada graf 10 simpul menggunakan Python. (Arthalia & Pristi, 2022) menyusun graf dari Google Maps (9 simpul) dan menghasilkan lintasan terpendek 6,2 km.



Gambar 1. Flowchart Penelitian

3.2 Teknik Pengumpulan Data

Data penelitian diperoleh dari dua sumber utama, yaitu BusListrikMedan dan Google Maps. Data halte dan koridor Trans Metro Deli diperoleh melalui situs BusListrikMedan (<https://buslistrikmedan.com>) yang diakses pada bulan Mei 2026. Data yang dikumpulkan meliputi nama halte, urutan halte pada setiap koridor, arah perjalanan, serta informasi perpindahan antar koridor.

Estimasi waktu tempuh antarhalte diperoleh menggunakan Google Maps (<https://maps.google.com>) pada tanggal 10–15 Mei 2026. Pengambilan data dilakukan secara manual dengan memilih rute kendaraan bermotor pada kondisi lalu lintas normal. Estimasi waktu tempuh tersebut digunakan sebagai bobot pada setiap edge dalam graf. Seluruh data kemudian disusun ke dalam bentuk edge list yang terdiri atas koridor, vertex asal, vertex tujuan, dan bobot waktu tempuh. Data ini selanjutnya digunakan sebagai masukan dalam pembentukan graf berarah berbobot dan penerapan Algoritma Dijkstra.

3.3 Analisis Sistem

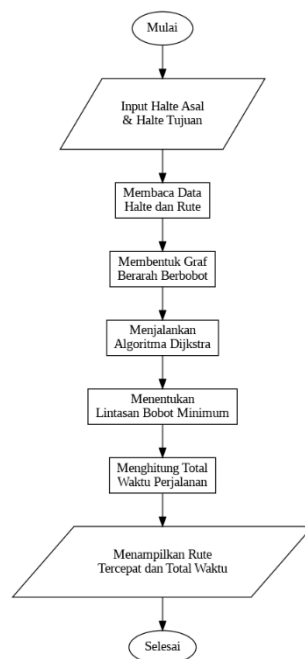
Analisis sistem dilakukan dengan memodelkan jaringan halte dan rute Trans Metro Deli ke dalam bentuk graf berarah berbobot (weighted directed graph). Pada penelitian ini, halte direpresentasikan sebagai vertex (simpul), sedangkan jalur langsung antarhalte direpresentasikan sebagai edge (sisi) berarah yang memiliki bobot berupa estimasi waktu tempuh perjalanan dalam satuan menit. (Syahputra & Program, 2017), (Hidayah, 2021),

(Syahroni et al., 2025) menerapkan algoritma Dijkstra untuk mencari rute terpendek dari titik awal ke titik tujuan. Graf berarah digunakan karena pada beberapa koridor Trans Metro Deli terdapat perbedaan jalur antara rute pergi dan rute pulang. Oleh karena itu, hubungan antarhalte tidak selalu bersifat dua arah. Jika terdapat jalur dari halte A menuju halte B, belum tentu tersedia jalur langsung dari halte B menuju halte A.

Data halte dan rute diperoleh dari BusListrikMedan, sedangkan bobot waktu perjalanan diperoleh dari estimasi Google Maps. Seluruh data kemudian diolah menjadi struktur graf yang digunakan sebagai masukan bagi algoritma Dijkstra untuk menentukan lintasan dengan total bobot minimum.

3.4 Alur Kerja Sistem

Alur kerja sistem menggambarkan proses pencarian rute tercepat pada jaringan Trans Metro Deli menggunakan Algoritma Dijkstra. Proses dimulai dengan memasukkan halte asal dan halte tujuan, kemudian sistem membentuk graf berarah berbobot berdasarkan data halte dan waktu tempuh antarhalte. Selanjutnya, Algoritma Dijkstra digunakan untuk menentukan lintasan dengan total bobot minimum. Hasil yang ditampilkan berupa rute tercepat dan total waktu perjalanan. Alur kerja sistem ditunjukkan pada Gambar 2.



Gambar 2. Flowchart Sistem

3.5 Pemodelan Graf Berarah

Komarullah (2024) menggambarkan destinasi wisata sebagai simpul dan jalur penghubung sebagai ruas berbobot jarak dari Google Maps, dengan titik awal dari terminal (Komarullah, 2024). Pada proyek ini, jaringan Trans Metro Deli dimodelkan sebagai graf berarah berbobot yang dinyatakan sebagai:

$$G = (V, E)$$

dengan:

- V merupakan himpunan vertex yang merepresentasikan halte Trans Metro Deli.
- E merupakan himpunan edge yang merepresentasikan jalur langsung antarhalte.

Setiap edge memiliki bobot waktu tempuh yang dinyatakan sebagai:

$w(u,v)$

dengan:

- u = halte asal,
- v = halte tujuan,
- $w(u,v)$ = waktu tempuh dari halte u menuju halte v dalam satuan menit

Karena graf bersifat berarah, maka berlaku:

$$(u,v) \neq (v,u)$$

yang menunjukkan bahwa jalur dari halte u ke halte v tidak selalu sama dengan jalur dari halte v ke halte u .

Berdasarkan data yang diperoleh, graf yang dibangun terdiri dari 236 vertex (halte) dan 245 edge (jalur antarhalte).

HALTE TERMINAL AMPLAS
→ HALTE AMPLAS (3)
HALTE AMPLAS
→ HALTE SMK PARULIAN 3 (1)
HALTE SMK PARULIAN 3
→ HALTE ALS (1)
HALTE ALS
→ HALTE TRITURA (2)

Gambar 3. Contoh Adjacency List

Data halte dan rute direpresentasikan dalam bentuk adjacency list sebelum dibangun menjadi graf menggunakan library NetworkX. Setiap vertex menyimpan daftar vertex tujuan yang terhubung langsung beserta bobot waktu tempuhnya.

3.6 Penerapan Algoritma Dijkstra

Setelah graf terbentuk, proses pencarian rute dilakukan menggunakan Algoritma Dijkstra. Algoritma ini digunakan untuk menentukan lintasan dengan total bobot minimum dari halte asal menuju halte tujuan. (Hutabarat et al., 2022) menggunakan algoritma Dijkstra pada graf berarah untuk menghitung jarak terpendek antar simpul (Sulistyo, 2023). (Indah et al., 2024) menerapkan algoritma tersebut dengan merepresentasikan lokasi sebagai titik, jalur sebagai ruas penghubung, dan jarak sebagai nilai bobot. Tahapan algoritma Dijkstra pada penelitian ini adalah sebagai berikut:

1. Menentukan halte asal dan halte tujuan.
2. Memberikan nilai 0 pada vertex asal dan nilai tak hingga (∞) pada vertex lainnya.
3. Memilih vertex dengan nilai jarak terkecil yang belum dikunjungi.

4. Melakukan proses relaksasi terhadap seluruh vertex tetangga.
5. Memperbarui nilai jarak apabila ditemukan lintasan yang lebih kecil.
6. Mengulangi proses hingga vertex tujuan ditemukan.
7. Menyusun kembali lintasan terpendek berdasarkan predecessor yang terbentuk.

Proses pembaruan jarak dilakukan menggunakan persamaan:

$$d(v) = \min(d(v), d(u) + w(u, v))$$

di mana:

- $d(v)$ adalah jarak minimum menuju vertex v ,
- $d(u)$ adalah jarak minimum menuju vertex sebelumnya,
- $w(u, v)$ adalah bobot edge dari u menuju v

Implementasi algoritma dilakukan menggunakan bahasa pemrograman Python dengan library NetworkX.

Kompleksitas waktu Algoritma Dijkstra menggunakan priority queue adalah $O((|V| + |E|) \log |V|)$. Dengan jumlah 236 vertex dan 245 edge, algoritma masih mampu melakukan pencarian lintasan secara efisien pada jaringan Trans Metro Deli.

3.7 Pengujian Sistem

Menguji algoritma Dijkstra dengan membandingkan empat rute alternatif menggunakan graf 12 simpul berbobot jarak dari Google Maps. Hasil pengujian menunjukkan rute terpendek A-H-I-E-F-G-L sejauh 26,4 km, lebih akurat dari perhitungan manual 26,5 km (Kurniawan et al., 2025). Pada penelitian ini digunakan lima skenario pengujian yang terdiri dari perjalanan dalam satu koridor maupun perjalanan yang melibatkan perpindahan antar koridor.

Pada setiap skenario, sistem menghasilkan:

1. lintasan tercepat yang dilalui,
2. total bobot waktu perjalanan,
3. jumlah halte yang dilewati,

3.8 Teknik Analisis Hasil

Membandingkan jarak tiga rute alternatif dari Google Maps untuk menentukan rute terpendek. Rumini & Lesmana (2020) membandingkan jarak hasil algoritma Dijkstra dengan metode Haversine, diperoleh selisih 0,648% (Lesmana, 2020). Pada penelitian ini, teknik analisis hasil dilakukan dengan membandingkan hasil lintasan dan estimasi waktu yang diperoleh dari algoritma Dijkstra terhadap estimasi waktu perjalanan yang tersedia pada Google Maps.

Proses validasi bertujuan untuk mengetahui apakah model graf yang dibangun mampu merepresentasikan jaringan halte Trans Metro Deli dengan baik. Apabila waktu hasil perhitungan algoritma mendekati estimasi Google Maps, maka model dianggap cukup representatif untuk digunakan dalam pencarian rute tercepat.

Namun demikian, penelitian ini memiliki keterbatasan karena bobot yang digunakan bersifat statis dan belum mempertimbangkan:

- waktu tunggu bus,
- jadwal keberangkatan,
- headway antar bus,
- kondisi lalu lintas real-time,
- kepadatan jalan.

Oleh karena itu, hasil yang diperoleh merupakan rute tercepat berdasarkan model graf yang dibangun dan bukan kondisi operasional aktual secara real-time.

4. HASIL DAN PEMBAHASAN

4.1 Implementasi Data Penelitian

Pada penelitian ini, data yang digunakan berupa data halte dan rute Trans Metro Deli yang terdiri dari 5 koridor utama. Data rute dan halte diperoleh dari BusListrikMedan sebagai sumber utama informasi koridor dan halte Trans Metro Deli.



Gambar 4. Peta Jaringan Trans Metro Deli

Sumber : [Wikipedia Bahasa Indonesia – Bus Listrik Medan](#)

Selain itu, estimasi waktu tempuh antarhalte diperoleh menggunakan Google Maps dengan satuan menit. Estimasi waktu tersebut digunakan sebagai bobot pada setiap edge dalam graf untuk merepresentasikan waktu perjalanan antarhalte. Bobot waktu dipilih karena lebih sesuai untuk menggambarkan efisiensi perjalanan transportasi publik dibandingkan menggunakan jarak tempuh.

Data penelitian kemudian disusun ke dalam bentuk tabel yang terdiri dari:

- koridor,
- arah perjalanan,
- vertex awal,
- vertex tujuan,
- dan bobot waktu tempuh.

Setiap halte direpresentasikan sebagai vertex (simpul), sedangkan hubungan antarhalte direpresentasikan sebagai directed edge (sisi berarah) yang memiliki bobot berupa estimasi waktu perjalanan. Pemodelan dilakukan menggunakan Directed Graph karena beberapa koridor memiliki arah perjalanan berbeda antara jalur keberangkatan dan kepulangan.

Contoh struktur data penelitian ditunjukkan pada tabel berikut.

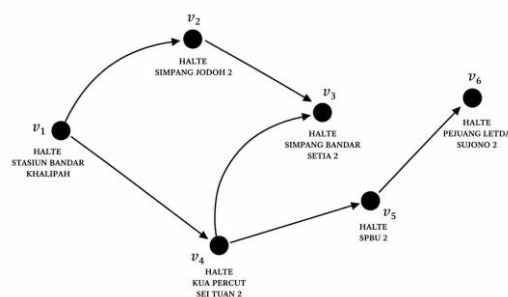
Tabel 1. Struktur Data

Koridor	Arah	Vertex Awal	Vertex Tujuan	Bobot (Menit)
K1	Pergi	HALTE TERMINAL AMPLAS	HALTE AMPLAS	2
K1	Pergi	HALTE AMPLAS	HALTE SMK PARULIAN 3	1
K1	Pergi	HALTE SMK PARULIAN 3	HALTE ALS	1
K1	Pergi	HALTE ALS	HALTE TRITURA	2

Data tersebut kemudian digunakan sebagai dasar pembentukan graf berbobot dan implementasi Dijkstra's Algorithm untuk menentukan rute transit tercepat antar koridor.

4.2 Pembentukan Graf Berarah

Model Pada penelitian ini, sistem rute Trans Metro Deli dimodelkan menggunakan Directed Graph untuk merepresentasikan hubungan antarhalte pada setiap koridor. Pemodelan graf berarah digunakan karena jalur keberangkatan dan kepulangan pada beberapa koridor memiliki lintasan yang berbeda sehingga hubungan antarhalte tidak selalu bersifat dua arah.



Gambar 5. Contoh Representasi Graf Berarah

Jaringan Trans Metro Deli dimodelkan sebagai graf berarah berbobot, dengan vertex merepresentasikan halte dan edge merepresentasikan hubungan langsung antarhalte. Bobot pada setiap edge menyatakan estimasi waktu tempuh dalam satuan menit.

Pemodelan berarah digunakan karena terdapat perbedaan lintasan pada rute pergi dan pulang.

Representasi graf pada penelitian ini dapat dituliskan sebagai:

$$G = (V, E)$$

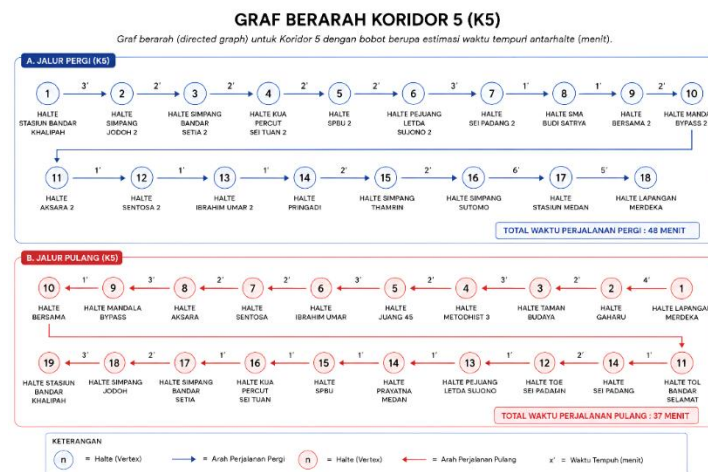
Keterangan:

- V merupakan himpunan vertex atau halte,
- E merupakan himpunan edge atau jalur antarhalte.

Karena menggunakan graf berarah, maka hubungan antar vertex memenuhi kondisi:

$$(u, v) \neq (v, u)$$

yang berarti lintasan dari vertex u menuju vertex v tidak selalu sama dengan lintasan dari vertex v menuju vertex u.



Gambar 6. Jalur Pergi dan Pulang Koridor 5

Perbedaan lintasan pergi dan pulang menyebabkan graf dimodelkan sebagai graf berarah berbobot (weighted directed graph). Bobot pada setiap edge dinyatakan sebagai $w(u,v)$, yaitu estimasi waktu tempuh dari vertex u ke vertex v. Selain itu, terdapat beberapa node transit yang terhubung dengan lebih dari satu koridor dan berfungsi sebagai titik perpindahan antarjalur. Contoh node transit ditunjukkan pada Tabel 5.

Tabel 2. Node Transit

Node Transit	Koridor
HALTE LAPANGAN MERDEKA	K1, K2, K3, K4, K5
HALTE STASIUN MEDAN	K1, K3, K4, K5
HALTE GEDUNG JUANG 45	K1, K4

Node transit direpresentasikan sebagai vertex yang terhubung dengan lebih dari satu koridor. Dalam graf yang dibangun, Halte Lapangan Merdeka, Halte Stasiun Medan, dan Halte Gedung Juang 45 memiliki derajat konektivitas yang lebih tinggi dibandingkan vertex lainnya sehingga berperan sebagai simpul transit utama.

4.3 Implementasi Algoritma Dijkstra

Algoritma Dijkstra diterapkan pada graf berarah berbobot untuk menentukan lintasan dengan total bobot minimum antara vertex asal dan vertex tujuan. Bobot yang digunakan berupa estimasi waktu tempuh antarhalte dalam satuan menit. Proses pembaruan jarak dilakukan menggunakan relasi :

$$d(v) = \min(d(v), d(u) + w(u, v))$$

Keterangan:

- $d(v)$ = jarak minimum menuju vertex v ,
- $d(u)$ = jarak minimum menuju vertex sebelumnya,
- $w(u, v)$ = bobot edge dari vertex u ke vertex v .

hingga diperoleh lintasan dengan bobot minimum. Pada implementasinya, penelitian ini menggunakan library NetworkX pada Python untuk mempermudah proses pembentukan graf dan pencarian lintasan terpendek.

4.4 Hasil Pengujian

4.4.1 Skenario Pengujian

Tabel 3. Skenario Pengujian

No	Halte Asal	Halte Tujuan	Jenis Pengujian
1	HALTE TERMINAL AMPLAS	HALTE JUANDA	Dalam satu koridor
2	HALTE PASAR BELAWAN	HALTE STASIUN MEDAN	Dalam satu koridor
3	HALTE RS MITRA MEDIKA	HALTE DELI PARK	Antar-koridor
4	HALTE J CITY	HALTE PLAZA MEDAN FAIR	Antar-koridor
5	HALTE STASIUN BANDAR KHALIPAH	HALTE TERMINAL PINANG BARIS	Antar-koridor

Pengujian dilakukan pada lima skenario yang mewakili berbagai kondisi perjalanan pada jaringan Trans Metro Deli. Skenario pertama dan kedua digunakan untuk menguji pencarian lintasan pada perjalanan dalam satu koridor. Skenario ketiga hingga kelima digunakan untuk menguji kemampuan algoritma dalam menentukan lintasan yang melibatkan perpindahan antar koridor melalui node transit.

4.4.2 Hasil Pengujian

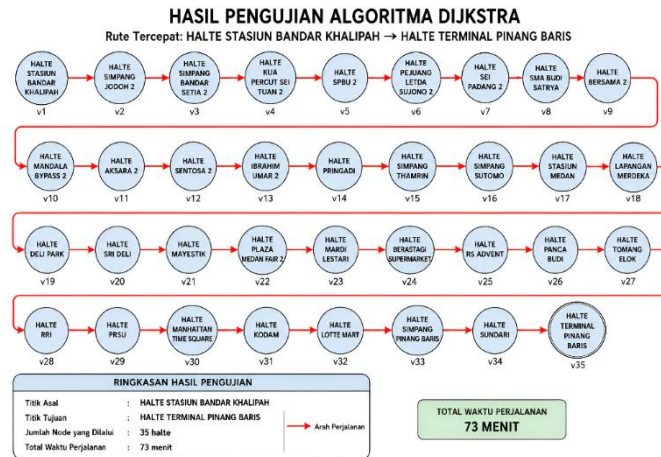
Tabel 4. Hasil Pengujian

No	Asal	Tujuan	Jumlah Halte	Total Waktu
1	Terminal Amplas	Juanda	15	22
2	Pasar Belawan	Stasiun Medan	29	65
3	RS Mitra Medika	Deli Park	22	40
4	J City	Plaza Medan Fair	25	46
5	Bandar Khalipah	Terminal Pinang Baris	35	73

Berdasarkan hasil pengujian, seluruh skenario berhasil menghasilkan lintasan dengan total bobot minimum sesuai perhitungan Algoritma Dijkstra. Waktu perjalanan tercepat diperoleh pada skenario HALTE TERMINAL AMPLAS menuju HALTE JUANDA dengan total waktu 22 menit. Sebaliknya, waktu perjalanan terlama diperoleh pada skenario HALTE STASIUN BANDAR KHALIPAH menuju HALTE TERMINAL

PINANG BARIS dengan total waktu 73 menit karena melibatkan lintasan yang lebih panjang dan beberapa node transit.

4.4.3 Detail Rute Hasil Dijkstra



Gambar 7. Hasil Pengujian Algoritma Dijkstra

Berdasarkan hasil pengujian menggunakan Algoritma Dijkstra, diperoleh lintasan tercepat dari HALTE STASIUN BANDAR KHALIPAH menuju HALTE TERMINAL PINANG BARIS dengan total waktu perjalanan 73 menit. Detail lintasan dan bobot kumulatif ditunjukkan pada Tabel 5.

Tabel 5. Detail Lintasan HALTE STASIUN BANDAR KHALIPAH – HALTE TERMINAL PINANG BARIS

No	Koridor	Vertex Awal	Vertex Tujuan	Bobot (Menit)
1	K5	HALTE STASIUN BANDAR KHALIPAH	HALTE SIMPANG JODOH 2	3
2	K5	HALTE SIMPANG JODOH 2	HALTE SIMPANG BANDAR SETIA 2	2
3	K5	HALTE SIMPANG BANDAR SETIA 2	HALTE KUA PERCUT SEI TUAN 2	2
4	K5	HALTE KUA PERCUT SEI TUAN 2	HALTE SPBU 2	2
5	K5	HALTE SPBU 2	HALTE PEJUANG LETDA SUJONO 2	2
6	K5	HALTE PEJUANG LETDA SUJONO 2	HALTE SEI PADANG 2	3
7	K5	HALTE SEI PADANG 2	HALTE SMA BUDI SATRYA	1
8	K5	HALTE SMA BUDI SATRYA	HALTE BERSAMA 2	1
9	K5	HALTE BERSAMA 2	HALTE MANDALA BYPASS 2	2
10	K5	HALTE MANDALA BYPASS 2	HALTE AKSARA 2	4
11	K5	HALTE AKSARA 2	HALTE SENTOSA 2	1
12	K5	HALTE SENTOSA 2	HALTE IBRAHIM UMAR 2	1
13	K5	HALTE IBRAHIM UMAR 2	HALTE PRINGADI	1
14	K5	HALTE PRINGADI	HALTE SIMPANG THAMRIN	2
15	K5	HALTE SIMPANG THAMRIN	HALTE SIMPANG SUTOMO	2
16	K5	HALTE SIMPANG SUTOMO	HALTE STASIUN MEDAN	6
17	K5	HALTE STASIUN MEDAN	HALTE LAPANGAN MERDEKA	5
18	Transit K5 → K1	HALTE LAPANGAN MERDEKA	HALTE DELI PARK	3

19	K1	HALTE DELI PARK	HALTE SRI DELI	1
20	K1	HALTE SRI DELI	HALTE MAYESTIK	2
21	K1	HALTE MAYESTIK	HALTE PLAZA MEDAN FAIR 2	2
22	K1	HALTE PLAZA MEDAN FAIR 2	HALTE MARDI LESTARI	2
23	K1	HALTE MARDI LESTARI	HALTE BERASTAGI SUPERMARKET	1
24	K1	HALTE BERASTAGI SUPERMARKET	HALTE RS ADVENT	1
25	K1	HALTE RS ADVENT	HALTE PANCA BUDI	1
26	K1	HALTE PANCA BUDI	HALTE TOMANG ELOK	6
27	K1	HALTE TOMANG ELOK	HALTE RRI	2
28	K1	HALTE RRI	HALTE PRSU	1
29	K1	HALTE PRSU	HALTE MANHATTAN TIME SQUARE	3
30	K1	HALTE MANHATTAN TIME SQUARE	HALTE KODAM	2
31	K1	HALTE KODAM	HALTE LOTTE MART	1
32	K1	HALTE LOTTE MART	HALTE SIMPANG PINANG BARIS	1
33	K1	HALTE SIMPANG PINANG BARIS	HALTE SUNDARI	2
34	K1	HALTE SUNDARI	HALTE TERMINAL PINANG BARIS	2
Total				73 Menit

Validasi manual dilakukan dengan menjumlahkan seluruh bobot edge pada lintasan yang dilalui. Berdasarkan Tabel 8 diperoleh:

$$3 + 2 + 2 + 2 + 2 + 3 + 1 + 1 + 2 + 4 + 1 + 1 + 1 + 2 + 2 + 6 + 5 + 3 + 1 + 2 + 2 + 2 + 1 + 1 + 1 + 6 + 2 + 1 + 3 + 2 + 1 + 1 + 2 + 2 = 73 \text{ menit}$$

Hasil penjumlahan tersebut sesuai dengan keluaran Algoritma Dijkstra yang menghasilkan total waktu perjalanan sebesar 73 menit.

4.5 Validasi Hasil

Tabel 6. Validasi Hasil Algoritma Dijkstra

No	Asal	Tujuan	Waktu Dijkstra (menit)	Waktu Google Maps (menit)	Selisih
1	Terminal Amplas	Juanda	22	24	2
2	Pasar Belawan	Stasiun Medan	65	69	4
3	RS Mitra Medika	Deli Park	40	42	2
4	J City	Plaza Medan Fair	46	49	3
5	Bandar Khalipah	Terminal Pinang Baris	73	76	3

Berdasarkan Tabel 6, hasil pencarian lintasan menggunakan Algoritma Dijkstra menunjukkan waktu perjalanan yang mendekati estimasi Google Maps dengan selisih antara 2–4 menit pada seluruh skenario pengujian. Selisih yang relatif kecil menunjukkan bahwa graf berarah berbobot yang dibangun mampu merepresentasikan jaringan Trans Metro Deli dengan baik. Perbedaan hasil yang masih muncul disebabkan oleh penggunaan bobot waktu statis yang belum mempertimbangkan kondisi lalu lintas real-time, waktu tunggu bus, dan faktor operasional lainnya.

4.6 Analisis Hasil

Berdasarkan lima skenario pengujian, Algoritma Dijkstra berhasil menentukan lintasan dengan total bobot minimum pada seluruh pasangan vertex asal dan tujuan. Bobot

minimum terkecil diperoleh pada rute Halte Terminal Amplas–Halte Juanda sebesar 22 menit, sedangkan bobot minimum terbesar diperoleh pada rute Halte Stasiun Bandar Khalipah–Halte Terminal Pinang Baris sebesar 73 menit. Perbedaan nilai bobot dipengaruhi oleh panjang lintasan, jumlah edge yang dilalui, dan keterlibatan node transit antar koridor.

Hasil pengujian menunjukkan bahwa node transit, seperti Halte Lapangan Merdeka dan Halte Stasiun Medan, memiliki peran penting dalam meningkatkan konektivitas graf. Selain itu, kesesuaian antara hasil perhitungan bobot dan keluaran algoritma menunjukkan bahwa implementasi Algoritma Dijkstra pada graf berarah berbobot telah berjalan dengan benar. Meskipun demikian, model yang dibangun masih menggunakan bobot waktu statis sehingga belum mempertimbangkan waktu tunggu bus, headway, jadwal keberangkatan, dan kondisi lalu lintas secara real-time. Oleh karena itu, hasil yang diperoleh merepresentasikan lintasan dengan total bobot minimum pada model graf yang digunakan.

5. KESIMPULAN DAN SARAN

Penelitian ini berhasil membangun model jaringan Trans Metro Deli dalam bentuk graf berarah berbobot yang terdiri dari 236 vertex dan 245 edge. Dalam model tersebut, halte direpresentasikan sebagai vertex, jalur antarhalte direpresentasikan sebagai edge berarah, dan estimasi waktu tempuh antarhalte digunakan sebagai bobot. Pemodelan graf berarah terbukti mampu merepresentasikan karakteristik operasional Trans Metro Deli yang memiliki perbedaan lintasan pada rute pergi dan rute pulang di beberapa koridor.

Berdasarkan model graf yang dibangun, Algoritma Dijkstra berhasil menentukan lintasan dengan total bobot minimum pada seluruh skenario pengujian. Hasil pengujian menunjukkan bahwa bobot minimum terkecil diperoleh pada rute Halte Terminal Amplas menuju Halte Juanda sebesar 22 menit, sedangkan bobot minimum terbesar diperoleh pada rute Halte Stasiun Bandar Khalipah menuju Halte Terminal Pinang Baris sebesar 73 menit. Temuan ini menunjukkan bahwa nilai bobot minimum tidak hanya dipengaruhi oleh jarak antarhalte, tetapi juga oleh struktur graf, jumlah edge yang dilalui, dan keterhubungan antar-koridor melalui node transit.

Hasil penelitian juga menunjukkan bahwa vertex transit, seperti Halte Lapangan Merdeka dan Halte Stasiun Medan, memiliki peranan penting dalam meningkatkan konektivitas graf karena menghubungkan beberapa koridor Trans Metro Deli. Keberadaan vertex tersebut memungkinkan terbentuknya lintasan antar-koridor yang dapat dipilih oleh Algoritma Dijkstra dalam proses pencarian lintasan berbobot minimum. Dengan demikian, struktur graf yang dibangun tidak hanya merepresentasikan hubungan antarhalte, tetapi juga mampu menggambarkan pola konektivitas jaringan transportasi publik secara matematis.

Validasi terhadap Google Maps menghasilkan selisih waktu sebesar 2–4 menit pada seluruh skenario pengujian. Hasil tersebut menunjukkan bahwa model graf berarah berbobot yang dibangun memiliki tingkat representasi yang baik terhadap jaringan Trans Metro Deli. Oleh karena itu, teori graf dan Algoritma Dijkstra dapat diterapkan sebagai pendekatan matematis untuk menganalisis dan menentukan lintasan dengan total bobot minimum pada sistem transportasi publik yang memiliki banyak simpul, jalur, dan titik transit.

6. REFERENSI

- Agusnur, A. (2025). Penggunaan teori graf dalam optimasi jaringan transportasi kota. *Jurnal Matematika Terapan*, 1(1), 29–35.
- Amelia, R., Meliala, V. S., Harahap, S. N., Al-majid, M. H., & Harliana, P. (2024). Optimalisasi rute transportasi menggunakan algoritma graf (studi kasus: Jaringan transportasi perkotaan). *Jurnal Mahasiswa Teknik Informatika*, 8(6), 12625–12632.
- Anastasya Putri Wibowo, K. A. (2025). Penerapan algoritma Dijkstra untuk menentukan rute terpendek dari RS Bina Kasih menuju RS Adam Malik. *Jurnal Ilmiah Matematika*, 13(1), 157–167.
- Arsyad, T. D., Dwi, M., Nababan, C., Imburi, R. K., & Harliana, P. (2025). Implementasi algoritma Dijkstra dalam mencari rute terpendek dari Universitas Negeri Medan. [Nama jurnal belum tersedia], 9(1), 235–242.
- Arta, T., & Victoria, D. (2023). Penerapan algoritma Dijkstra dalam pemetaan UMKM berbasis Android. *Bulletin of Computer Science Research*, 3(6), 420–426.
- Arthalia, I., & Pristi, W. (2022). Implementasi algoritma Dijkstra untuk menentukan rute terpendek menuju pelayanan kesehatan. *Jurnal Ilmiah Sistem Informasi*, 1(1), 30–37.
- Devi, H., Ariyani, H. D., Nafillah, K., Nindita, K., & Rahayu, S. T. (2026). Pemodelan graf berarah berbobot untuk optimasi penentuan rute terpendek antar kampus Polimarin berbasis algoritma Dijkstra. *Jurnal Ilmiah Matematika, Sains dan Teknologi*, 14(1), 151–162.
- Ginasta, N. G. (2024). Implementation of the best route search to find out the location of parking places in the e-parking system using the Dijkstra algorithm and best first search. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 4, 607–613.
- Hidayah, A. A. (2021). Penerapan algoritma Dijkstra pada aplikasi jasa transportasi. *Al Ulum Sains dan Teknologi*, 7, 9–13.
- Hutabarat, C. (2022). *Aplikasi konsep graf berbobot untuk menentukan lokasi fasilitas umum*. [Informasi sumber belum tersedia].
- I, N. R., Hilmi, M. Z., Proborini, E., & Husain, F. (2025). Pencarian rute terpendek dengan algoritma Dijkstra. *Jurnal EDU Elektromatika (JEE)*, 6(1), 50–57.
- Indah, R., Pasaribu, V., & Yurinanda, S. (2024). Optimasi rute terpendek pemeliharaan listrik dengan algoritma Dijkstra di PLN UIP Sumbagsel. *Journal of Islamic Science and Technology*, 9, 240–248.
- Komarullah, H. (2024). Penggunaan algoritma Dijkstra dalam [judul belum lengkap]. *As-Sunnayah*, 4(2), 96–101.
- Kurniawan, E. K., & Atmojo, W. T. (2025). Penerapan algoritma Dijkstra dalam penentuan rute terpendek kunjungan [judul perlu diperiksa kelengkapannya]. *Jurnal Sistem Informasi*, 7(2).

- Lesmana, D. (2020). Algoritma Dijkstra untuk menentukan jalur tercepat pada pendistribusian barang berbasis mobile. *Jurnal Sistem dan Teknologi Informasi*, 8(4), 362–368.
- Pasaribu, W., Purba, C., & Gunawan, I. (2026). Implementasi algoritma Dijkstraa berbasis teori graf untuk penentuan rute terpendek dari STIKOM Tunas Bangsa ke Suzuya Merdeka Mall Kota Pematangsiantar. *Jurnal Teknik Informatika dan Teknologi Informasi*, 6.
- Rufus, E. C., Riyadi, R. R., Hasibuan, D. N., Christian, E., & Pranatawijaya, V. H. (2024). Penerapan algoritma Dijkstra dalam menentukan rute terpendek untuk jasa pengiriman barang di Palangka Raya. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(3), 3387–3391.
- Simbolon, A. B., Artika, D. A., Sitanggang, Y. C., & Harliana, P. (2025). Implementasi algoritma Dijkstra dalam menganalisis rute terpendek dan efisiensi jarak tempuh dari Stasiun Kereta Api Medan menuju 6 kampus di area Medan Estate. [Nama jurnal belum tersedia], 9(1), 162–168.
- Sulistyo, H. (2023). Strategies in the 2022 Russia-Ukraine crisis. *Jurnal Keamanan Nasional*, 9(2).
- Syahputra, S. (2017). Penentuan rute terpendek pendistribusian naskah ujian nasional menggunakan algoritma Dijkstra. *Jurnal Teknik Informatika Kaputama*, 1(1).
- Syahroni, M. (2025). Implementasi algoritma Dijkstra untuk pencarian rute terpendek ke museum. *Jurnal Penelitian Matematika dan Pendidikan Matematika*, 8, 244–253.
- Taneo, R. E., Ndun, R., Fallo, D. Y. A., & Do, F. (2025). Optimasi jalur terpendek menggunakan algoritma Dijkstra dan greedy pada sistem informasi geografis. *Jurnal Kridatama Sains dan Teknologi*, 7(1), 572–582.
- Wahyudi, R., Alfin, M., Henrydunan, J. B., & Harliana, P. (2024). Penerapan algoritma Dijkstra untuk optimasi ke empat gerbang kampus menggunakan Python. [Nama jurnal belum tersedia], 8(6), 12073–12078.
- Yudhana, A., & Azhar, A. (2019). Implementasi algoritma Dijkstra untuk analisis rute transportasi umum Transjogja berbasis Android. *Jurnal Sistem Informasi Bisnis*, 32–38. [Volume dan nomor terbitan perlu diperiksa].
- Yusnita, A., Lailiyah, S., & Merta, I. G. A. (2021). Application of Dijkstra's algorithm in searching the shortest path of coal production locations. *JTKSI*, 4(1), 9–13.