JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 3, No. 12 Tahun 2026
ISSN 3025-0919 (media online)
Hal 3075-3084

Komparasi Algoritma Dijkstra dan A-Star untuk Optimasi Rute Kendaraan di Kota Medan

M. Faris Al Rafiq[1], Dea Anggraini[2], Farizky Aulia Lubis[3], Adidtya Perdana[4]

[1-4]Fakultas Matematika dan Ilmu Pengetahuan Alam, Program Studi Ilmu Komputer, Universitas Negeri Medan, Medan, Indonesia
Email: [1]alrafiqmfaris@gmail.com, [2]Anggrainidea2006@gmail.com, [3]farizkyaulia18@gmail.com, [4]adidtya@unimed.ac.id

Abstrak− Peningkatan kebutuhan akan efisiensi mobilitas di kawasan perkotaan menuntut adanya sistem navigasi cerdas yang mampu menentukan jalur perjalanan paling optimal. Penelitian ini mengkaji permasalahan pencarian rute terpendek (*shortest path problem*) dengan membandingkan performa komputasi algoritma Dijkstra dan A* (A-Star). Eksperimen dilakukan secara komprehensif pada representasi graf jaringan jalan nyata di Kota Medan (terdiri dari 42.692 *node* dan 99.736 *edge*) yang diekstraksi melalui *OpenStreetMap* (OSM). Pengujian menggunakan 100 *test case* yang terstratifikasi dalam kategori jarak dekat, sedang, dan jauh. Hasil penelitian menunjukkan bahwa kedua algoritma secara konsisten menghasilkan jarak rute yang identik, membuktikan bahwa penggunaan heuristik *haversine* pada A* tetap menjamin optimalitas jalur. Namun dari segi efisiensi waktu eksekusi, algoritma A* terbukti jauh lebih unggul dengan memenangkan 95% kasus uji dan mencatatkan rata-rata *speedup* sebesar 2,83 kali lipat dibandingkan Dijkstra. Analisis lebih lanjut mengungkap bahwa keunggulan waktu komputasi A* memuncak pada rute jarak dekat (3,77x) dan perlahan menurun pada rute jarak jauh (1,68x) akibat kompleksitas topologi jaringan jalan. Berdasarkan uji statistik *paired t-test*, perbedaan performa ini dinyatakan signifikan. Hasil analisis ini diharapkan dapat menjadi rujukan strategis bagi pengembangan sistem transportasi digital, khususnya penyesuaian algoritma perutean pada kondisi topologi lalu lintas riil di Indonesia.

Kata Kunci: Algoritma A-Star; Algoritma Dijkstra; Navigasi Kendaraan; OpenStreetMap; Rute Terpendek.

Abstract− *The increasing need for mobility efficiency in urban areas requires smart navigation systems capable of determining the most optimal travel routes. This study examines the shortest path problem by comparing the computational performance of the Dijkstra and A (A-Star) algorithms. Comprehensive experiments were conducted on a real road network graph of Medan City (consisting of 42,692 nodes and 99,736 edges) extracted through OpenStreetMap (OSM). The testing utilized 100 test cases stratified into short, medium, and long-distance categories. The results showed that both algorithms consistently produced identical route distances, proving that the use of the haversine heuristic in A* still guarantees path optimality. However, in terms of execution time efficiency, the A* algorithm proved to be far superior by winning 95% of the test cases and recording an average speedup of 2.83 times compared to Dijkstra. Further analysis revealed that the computational time advantage of A* peaks in short-distance routes (3.77x) and gradually decreases in long-distance routes (1.68x) due to the complexity of the road network topology. Based on the paired t-test statistical analysis, this performance difference is declared significant. The results of this analysis are expected to serve as a strategic reference for the development of digital transportation systems, specifically for routing algorithm adjustments in real traffic topology conditions in Indonesia.*

Keywords*: A-Star Algorithm; Dijkstra Algorithm; Vehicle Navigation; OpenStreetMap; Shortest Path.*

1. PENDAHULUAN

Layanan berbasis lokasi seperti ojek daring, pengiriman paket, aplikasi wisata semuanya bertumpu pada satu persoalan mendasar yang sama: bagaimana mencari jalur berbiaya minimum antara dua titik dalam suatu jaringan. Persoalan ini dikenal sebagai *shortest path problem*, dan dimodelkan menggunakan graf berbobot di mana node merepresentasikan persimpangan jalan, sedangkan edge merepresentasikan ruas jalan beserta bobot jarak atau waktu tempuhnya. Kebutuhan akan solusi yang cepat dan tepat telah mendorong penerapan nyata di berbagai bidang: sistem routing armada bus perkotaan (Chen, 2022), perencanaan jalur kendaraan cerdas berbasis sensor (Chu et al., 2024), navigasi kendaraan khusus pada jaringan jalan berskala besar (Mienye & Jere, 2025), hingga optimasi distribusi pada persoalan transportasi klasik (Vijayalaxmi, 2024).

Dari sekian banyak algoritma yang pernah dikembangkan, Dijkstra adalah salah satu yang paling mapan. Cara kerjanya sederhana: secara berulang dipilih simpul dengan jarak kumulatif terkecil dari titik asal, mengikuti prinsip *greedy*. Selama semua bobot bernilai non-negatif, algoritma

JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 3, No. 12 Tahun 2026
ISSN 3025-0919 (media online)
Hal 3075-3084

ini selalu menghasilkan jalur yang optimal. Namun ada kelemahan yang tidak bisa diabaikan. Dijkstra tidak memiliki arah eksplorasi, ia menjelajahi simpul ke segala penjuru tanpa memperhitungkan posisi simpul terhadap titik tujuan. Pada jaringan jalan yang padat, hal ini berujung pada lonjakan jumlah simpul yang dievaluasi dan membengkaknya waktu komputasi (Grujic & Grujic, 2025; Verma & Kumar, 2024). Persoalan ini diperumit oleh fakta bahwa performa Dijkstra sangat sensitif terhadap karakteristik topologi graf berdasarkan kerapatan simpul, arah sisi sehingga hasil pengujian di satu jaringan tidak serta-merta berlaku untuk jaringan lain.

Algoritma A* (*A-Star*) dirancang sebagai pengembangan langsung dari Dijkstra. Perbedaan utamanya terletak pada fungsi heuristik yang ditambahkan: fungsi ini mengestimasi biaya sisa dari simpul aktif menuju tujuan, sehingga eksplorasi dapat diarahkan ke simpul-simpul yang secara geometris lebih dekat ke tujuan. Hasilnya, jumlah simpul yang perlu dievaluasi jauh berkurang (Wang et al., 2022). Ketika keduanya dibandingkan secara langsung, panjang jalur yang dihasilkan relatif setara, tetapi A* konsisten mencapai solusi lebih cepat (Elshaer et al., 2025). Keunggulan ini pun tidak terbatas pada navigasi darat, dalam optimasi rute pelayaran dan logistik ladang angin lepas pantai, penghematan waktu komputasi yang dihasilkan A* terbukti signifikan secara operasional (Milin et al., 2025).

Studi perbandingan antara Dijkstra dan A* sebenarnya sudah cukup banyak. Tetapi sebagian besar masih mengandalkan data simulasi atau jaringan jalan di luar Indonesia. (Al Bager & Ahmed, 2021) membandingkan Dijkstra, A*, dan Bellman Ford secara langsung dan menemukan bahwa keunggulan tiap algoritma sangat bergantung pada skala dan struktur jaringan yang diuji. Temuan sejumlah kajian yang lebih baru memperkuat kesimpulan tersebut, tidak ada algoritma yang unggul secara mutlak di semua kondisi (Sapundzhi et al., 2025; Singh et al., 2025). (Ugwoke et al., 2025) bahkan memperluas tinjauan hingga algoritma metaheuristik dan menyimpulkan bahwa Dijkstra maupun A* tetap relevan untuk banyak skenario praktis, asalkan diterapkan pada jaringan dengan karakteristik yang sesuai. Jaringan jalan di kota-kota Indonesia, termasuk Kota Medan, memiliki kekhasan tersendiri: kepadatan simpul yang tinggi, dominasi jalan satu arah, dan hambatan geografis seperti sungai serta rel kereta. Karakteristik ini belum banyak dikaji dalam literatur yang ada.

Dari kesenjangan itulah penelitian ini berangkat. Dilakukan analisis komparatif antara algoritma Dijkstra dan A* menggunakan data jaringan jalan Kota Medan yang diperoleh dari OpenStreetMap melalui library OSMnx. Graf yang digunakan mencakup 42.692 simpul dan 99.736 segmen jalan. Sebanyak 100 pasang titik asal-tujuan diuji, dikelompokkan ke dalam tiga kategori jarak, dengan kinerja diukur berdasarkan waktu eksekusi, panjang jalur, jumlah node yang dikunjungi, dan speedup. Signifikansi perbedaan performa divalidasi menggunakan uji *paired t-test*. Hasil penelitian ini diharapkan menjadi landasan empiris yang lebih kontekstual bagi pengembangan sistem navigasi berbasis data geospasial di Indonesia.

2. METODE

2.1 Dataset dan Sumber Data

Data jaringan jalan yang digunakan dalam penelitian ini bersumber dari OpenStreetMap (OSM), sebuah platform peta digital berbasis kontribusi komunitas yang menyediakan data geospasial secara terbuka. Pengunduhan data dilakukan menggunakan library OSMnx dengan query lokasi "Medan, North Sumatra, Indonesia" dan tipe jaringan drive, yaitu jaringan yang hanya mencakup jalan yang dapat dilalui kendaraan bermotor. Setelah diunduh, data disimpan dalam format GraphML agar dapat dimuat kembali pada setiap sesi eksperimen tanpa perlu mengakses server OSM berulang kali.

Dataset yang dihasilkan memiliki 42.692 simpul yang merepresentasikan persimpangan jalan dan 99.736 sisi yang merepresentasikan segmen jalan. Setiap sisi dilengkapi atribut panjang dalam satuan meter yang digunakan sebagai bobot dalam proses pencarian jalur. Sebelum digunakan, graf melewati tahap validasi yang mencakup tiga pemeriksaan, yaitu konektivitas graf untuk memastikan semua simpul tergabung dalam satu komponen, kelengkapan atribut bobot pada setiap sisi, dan kevalidan koordinat geografis pada setiap simpul. Jika ditemukan komponen yang tidak terhubung, hanya komponen terbesar yang dipertahankan.

Seluruh eksperimen dijalankan pada satu perangkat komputer dengan prosesor Intel Core i5 atau AMD Ryzen 5, RAM minimal 4 GB, dan sistem operasi Windows 10 64-bit. Implementasi

JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 3, No. 12 Tahun 2026
ISSN 3025-0919 (media online)
Hal 3075-3084

dikerjakan menggunakan Python 3.13 dengan dukungan library utama berupa OSMnx 2.1.0 untuk pemrosesan data geospasial, NetworkX 3.6.1 untuk representasi graf dan eksekusi algoritma, Pandas 3.0.0 untuk penyimpanan hasil eksperimen, serta SciPy untuk keperluan analisis statistik.

2.2 Implementasi Algoritma

Kedua algoritma yang dibandingkan dalam penelitian ini telah tersedia pada library NetworkX, sehingga implementasi berfokus pada pengukuran performa dan pencatatan hasil. Algoritma Dijkstra dipanggil menggunakan fungsi *dijkstra_path()* dan *dijkstra_path_length()* dengan parameter bobot *length*. Algoritma bekerja secara *greedy* dengan menginisialisasi jarak simpul asal bernilai nol dan seluruh simpul lainnya bernilai tak hingga, lalu secara iteratif memperbarui jarak minimum ke setiap simpul tetangga menggunakan struktur *priority queue*.

Algoritma A* diimplementasikan melalui fungsi *astar_path()* dan *astar_path_length()* dengan tambahan parameter fungsi heuristik. Fungsi evaluasi A* menggunakan rumus:

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

di mana g(n) merupakan biaya aktual dari titik asal ke simpul n, dan h(n) merupakan estimasi biaya dari simpul n ke titik tujuan. Fungsi heuristik yang digunakan adalah jarak Haversine, yakni jarak garis lurus antara dua titik berdasarkan koordinat geografis lintang dan bujur. Fungsi ini bersifat *admissible* karena estimasi yang dihasilkan tidak pernah melebihi biaya sebenarnya, sehingga A* tetap menjamin solusi yang optimal.

2.3 Desain Eksperimen dan Metrik Evaluasi

Eksperimen dirancang menggunakan 100 pasang titik asal-tujuan yang dipilih secara acak dari graf melalui fungsi *random.choice()* pada Python. Setiap pasang titik harus memenuhi dua syarat, yaitu titik asal dan tujuan tidak boleh sama, dan harus terdapat jalur yang valid antara keduanya. Pasang titik yang tidak memenuhi syarat diganti hingga terkumpul 100 pasang yang valid. Seluruh pasang titik kemudian dikelompokkan ke dalam tiga kategori berdasarkan jarak jalur yang dihasilkan, yaitu kategori dekat untuk jarak di bawah 5 km sebanyak 34 pasang, kategori sedang untuk jarak 5 hingga 15 km sebanyak 33 pasang, dan kategori jauh untuk jarak di atas 15 km sebanyak 33 pasang.

Pada setiap pasang titik, kedua algoritma dijalankan secara bergantian dan hasilnya diukur berdasarkan empat metrik. Waktu eksekusi diukur menggunakan fungsi *time.perf_counter()* dengan mencatat selisih waktu sebelum dan sesudah algoritma berjalan, kemudian dikonversi ke satuan milidetik. Panjang jalur dihitung dengan menjumlahkan bobot seluruh sisi yang dilalui dalam satuan meter. Jumlah simpul yang dikunjungi dicatat sebagai representasi efisiensi proses eksplorasi algoritma. Rasio waktu eksekusi Dijkstra terhadap A* pada pasangan titik yang sama, dihitung dengan rumus Speedup. Seluruh hasil disimpan secara otomatis ke dalam struktur tabel menggunakan Pandas dan diekspor ke format CSV.

2.4 Metode Analisis Data

Analisis data dilakukan melalui dua pendekatan yang saling melengkapi. Pendekatan deskriptif digunakan untuk menghitung nilai rata-rata, median, standar deviasi, serta nilai minimum dan maksimum dari setiap metrik pada masing-masing algoritma. Selain itu, dihitung pula nilai *speedup* sebagai rasio waktu eksekusi Dijkstra terhadap A* menggunakan rumus:

$$\text{Speedup} = \frac{Waktu\ Dijkstra}{Waktu\ A*} \quad (2)$$

Nilai *speedup* lebih dari 1 menandakan A* bekerja lebih cepat dari Dijkstra pada pasang titik tersebut.

Pendekatan inferensial dilakukan menggunakan uji *paired t-test* dua sisi untuk menguji apakah perbedaan rata-rata waktu eksekusi antara kedua algoritma signifikan secara statistik. Hipotesis nol yang diuji menyatakan tidak terdapat perbedaan rata-rata waktu eksekusi antara Dijkstra dan A*, sementara hipotesis alternatif menyatakan sebaliknya. Batas signifikansi yang digunakan adalah $\alpha = 0{,}05$. Seluruh hasil analisis divisualisasikan dalam bentuk grafik batang, grafik garis, *box plot*, dan *scatter plot* menggunakan library Matplotlib.

JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 3, No. 12 Tahun 2026
ISSN 3025-0919 (media online)
Hal 3075-3084

3. ANALISA DAN PEMBAHASAN

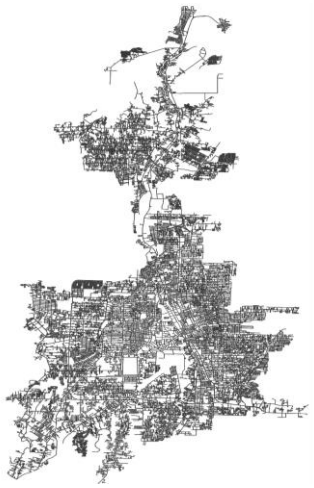
3.1 Deskripsi Dataset

Data jaringan jalan Kota Medan berhasil diunduh dan divalidasi sebelum digunakan dalam eksperimen. Karakteristik lengkap dataset disajikan pada Tabel 1.

Tabel 1. Statistik Dataset Jaringan Jalan Kota Medan

Atribut Dataset	Nilai
Wilayah	Kota Medan, Sumatera Utara, Indonesia
Jenis Jaringan	Drive network (kendaraan bermotor)
Jumlah Node	42.692 persimpangan jalan
Jumlah Edge	99.736 segmen jalan
Tipe Jalan Dominan	Residential: 57.337 edge (57,5%)
Tipe Jalan Lain	Living street: 26.298 Tertiary: 6.529 Primary: 2.899 Secondary: 1.867
Tipe Jalan Utama	Trunk: 1.142 Motorway: 3
Format File	GraphML (.graphml), ukuran ±40 MB
Hasil Test Jalur Awal	Jarak: 4,19 km Node dilalui: 48

Jaringan jalan Kota Medan memiliki pola distribusi yang tidak seragam. Wilayah pusat kota bagian selatan menunjukkan kepadatan simpul yang jauh lebih tinggi dibandingkan wilayah utara. Keberadaan Sungai Deli yang membelah kota serta banyaknya jalan satu arah di kawasan pusat menjadi dua faktor utama yang menambah kompleksitas topologi jaringan. Kondisi geografis ini berpengaruh langsung terhadap akurasi fungsi heuristik pada algoritma A*, terutama ketika rute yang ditempuh tidak linear terhadap arah garis lurus menuju titik tujuan. Visualisasi jaringan jalan Kota Medan ditunjukkan pada Gambar 1.



Gambar 1. Visualisasi Jaringan Jalan Kota Medan (42.692 Node, 99.736 Edge). Sumber: OpenStreetMap, diolah dengan OSMnx

3.2 Hasil Eksperimen Keseluruhan

Eksperimen berhasil dijalankan pada 100 pasang titik yang valid, terdiri dari 34 pasang kategori dekat, 33 pasang kategori sedang, dan 33 pasang kategori jauh. Pada seluruh 100 pasang

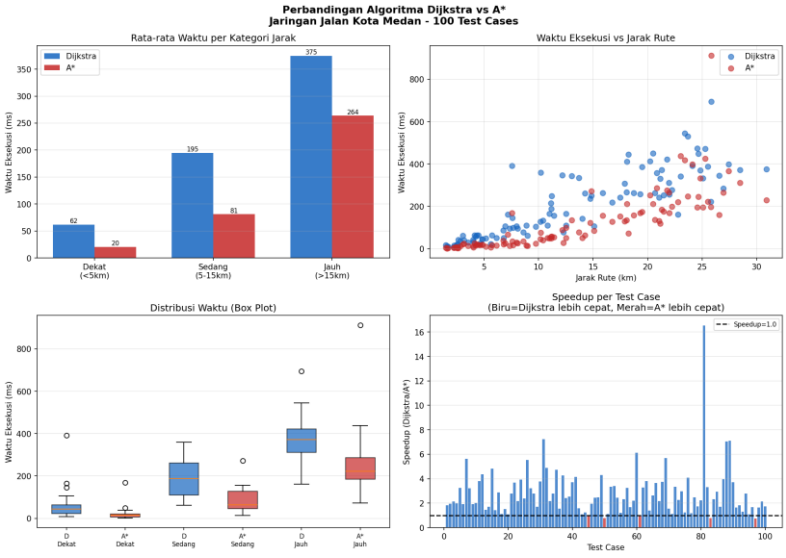
JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 3, No. 12 Tahun 2026
ISSN 3025-0919 (media online)
Hal 3075-3084

titik tersebut, kedua algoritma menghasilkan panjang jalur yang sepenuhnya identik. Temuan ini mengonfirmasi bahwa penggunaan heuristik haversine yang bersifat *admissible* pada A* tidak mengorbankan optimalitas solusi demi kecepatan komputasi. Statistik deskriptif lengkap dari hasil eksperimen disajikan pada Tabel 2.

Tabel 2. Statistik Keseluruhan Hasil Eksperimen (100 Test Cases)

Metrik	Dijkstra	A*
Rata-rata (ms)	208,98	120,74
Median (ms)	176,97	64,91
Std Deviasi (ms)	155,16	137,13
Minimum (ms)	7,73	2,00
Maksimum (ms)	695,14	912,65
Rata-rata Speedup	—	**2,83x**
Jumlah Kemenangan	5 / 100 (5%)	**95 / 100 (95%)**

Secara keseluruhan, A* menyelesaikan pencarian jalur dengan rata-rata waktu 120,74 ms, sementara Dijkstra membutuhkan rata-rata 208,98 ms. Selisih ini setara dengan rata-rata *speedup* 2,83x, artinya A* bekerja hampir tiga kali lebih cepat dari Dijkstra secara keseluruhan. A* berhasil unggul pada 95 dari 100 pasang titik yang diuji. Nilai median A* sebesar 64,91 ms yang jauh lebih rendah dari nilai rata-ratanya mengindikasikan distribusi data yang condong ke kanan (*right-skewed*). Kondisi ini disebabkan oleh beberapa pasang titik dengan waktu eksekusi A* yang sangat tinggi, yaitu lima kasus anomali yang akan dibahas pada subbagian 3.4. Standar deviasi Dijkstra yang lebih besar (155,16 ms) dibandingkan A* (137,13 ms) mengindikasikan waktu eksekusi Dijkstra lebih bervariasi dan kurang dapat diprediksi antar pasang titik. Perbandingan rata-rata waktu eksekusi per kategori jarak ditunjukkan pada Gambar 2.



Gambar 2. Rata-rata Waktu Eksekusi Dijkstra vs A per Kategori Jarak (100 Test Cases)

3.3 Analisis per Kategori Jarak

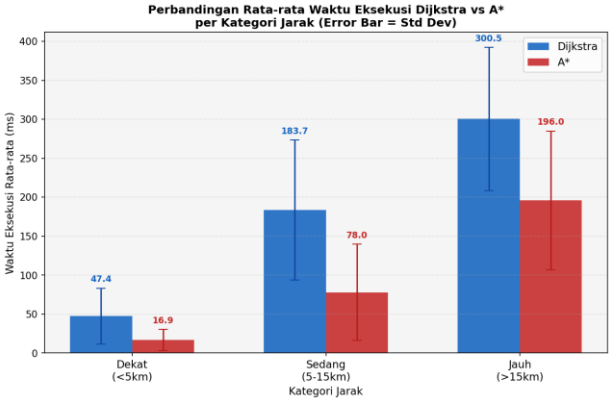
Salah satu kontribusi utama penelitian ini adalah pengamatan pola performa berdasarkan kategori jarak. Tabel 3 menyajikan perbandingan performa kedua algoritma pada ketiga kategori yang telah ditetapkan.

JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 3, No. 12 Tahun 2026
ISSN 3025-0919 (media online)
Hal 3075-3084

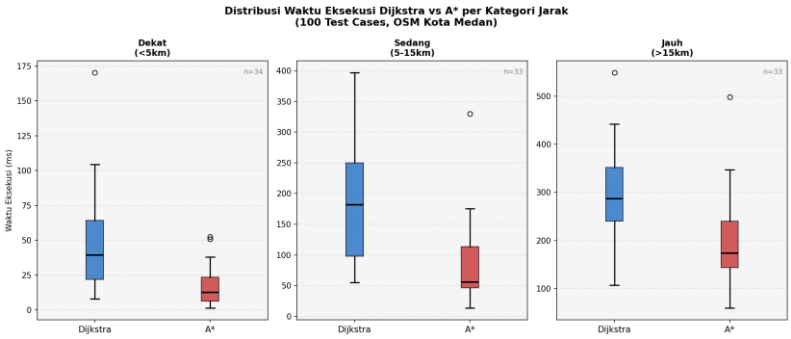
Tabel 3. Perbandingan Performa per Kategori Jarak

Kategori	N	Avg Dijkstra (ms)	Avg A* (ms)	Avg Speedup
Dekat (<5 km)	34	61,91	20,46	**3,77x**
Sedang (5–15 km)	33	194,89	81,04	**3,03x**
Jauh (>15 km)	33	374,60	263,75	**1,68x**
TOTAL / RATA-RATA	**100**	**208,98**	**120,74**	**2,83x**

Terlihat pola yang konsisten bahwa keunggulan A* paling besar pada jarak dekat dan secara bertahap berkurang seiring bertambahnya jarak. Pada kategori dekat, heuristik haversine bekerja sangat akurat karena arah garis lurus ke tujuan hampir selalu konsisten dengan jalur jalan yang tersedia, sehingga A* langsung mengarahkan eksplorasi ke simpul relevan tanpa penelusuran yang tidak produktif. *Speedup* tertinggi tercatat pada pasang titik ke-81 dengan jarak 2,63 km sebesar 16,54x. Pada kategori sedang, keunggulan sedikit menurun karena jaringan jalan mulai menunjukkan kompleksitasnya, namun heuristik masih cukup akurat. Adapun pada kategori jauh, keberadaan Sungai Deli, jalan layang, dan banyaknya jalan satu arah menyebabkan rute aktual sering menyimpang dari estimasi garis lurus, sehingga akurasi heuristik menurun dan A* lebih sering merevisi prioritas eksplorasinya. Temuan ini sejalan dengan (Aldhafferi, 2025) yang menyatakan efisiensi A* sangat bergantung pada representativitas fungsi heuristik terhadap kondisi jaringan nyata. Distribusi waktu eksekusi per kategori ditunjukkan pada Gambar 3 dan Gambar 4.



Gambar 3. Rata-rata Waktu per Kategori Jarak



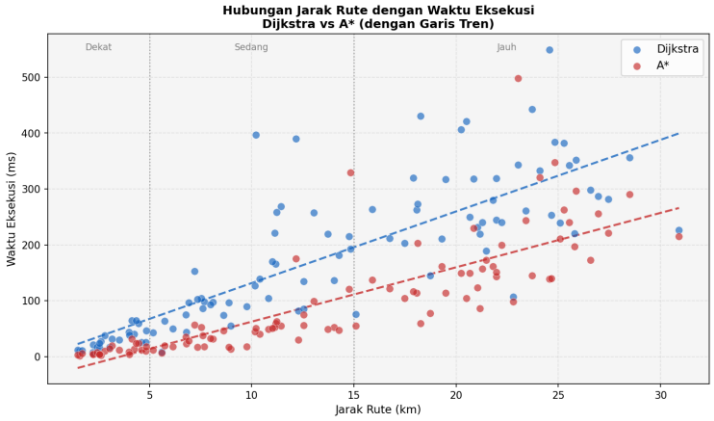
Gambar 4. Distribusi Waktu Eksekusi Dijkstra vs A per Kategori Jarak (Box Plot)

3.4 Analisis Hubungan Jarak dan Waktu Eksekusi

Gambar 5 menampilkan *scatter plot* hubungan antara jarak rute dengan waktu eksekusi kedua algoritma. Terlihat korelasi positif yang jelas pada keduanya, semakin jauh jarak, semakin besar waktu yang dibutuhkan. Titik-titik A* secara konsisten berada di bawah titik Dijkstra pada hampir

JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 3, No. 12 Tahun 2026
ISSN 3025-0919 (media online)
Hal 3075-3084

seluruh rentang jarak, dengan variansi Dijkstra yang lebih besar pada jarak menengah ke atas mengindikasikan performa yang kurang stabil.



Gambar 5. Hubungan Jarak Rute dengan Waktu Eksekusi Dijkstra dan A (dengan Garis Tren)

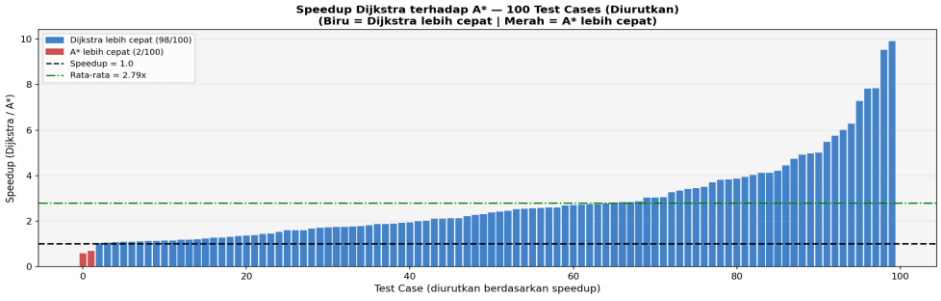
3.5 Analisis Kasus Anomali

Dari 100 pasang titik, terdapat lima kasus di mana Dijkstra lebih cepat dari A*, sebagaimana dirinci pada Tabel 4.

Tabel 4. Rincian 5 Kasus Anomali (Dijkstra Lebih Cepat dari A*)

TC	Kategori	Dijkstra (ms)	A* (ms)	Speedup	Jarak (km)
45	Sedang (5-15 km)	250,62	271,39	0,923x	14,84
50	Jauh (>15 km)	342,52	437,76	0,782x	23,02
61	Jauh (>15 km)	392,15	398,74	0,983x	24,09
83	Jauh (>15 km)	695,14	912,65	0,762x	25,86
97	Jauh (>15 km)	161,11	220,17	0,732x	22,80

Empat dari lima kasus berasal dari kategori jarak jauh. Kondisi ini terjadi akibat fenomena *dead-end heuristic*, di mana jalur sebenarnya menyimpang jauh dari estimasi garis lurus sehingga A* harus mengevaluasi ulang banyak simpul yang sebelumnya diprioritaskan. Meski demikian, selisih *speedup* pada kasus anomali hanya berkisar 0,73x hingga 0,98x, jauh lebih kecil dibandingkan keunggulan A* yang bisa mencapai 16,54x pada kasus terbaiknya. Distribusi *speedup* seluruh pasang titik ditunjukkan pada Gambar 6.



Gambar 6. Speedup per Test Case

3.6 Uji Statistik Paired T-Test

Uji *paired t-test* dilakukan untuk memverifikasi signifikansi statistik dari perbedaan performa yang diamati. Hasil uji disajikan pada Tabel 5.

Tabel 5. Hasil Uji Statistik Paired T-Test

JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 3, No. 12 Tahun 2026
ISSN 3025-0919 (media online)
Hal 3075-3084

Parameter Uji	Nilai
Uji yang digunakan	Paired t-test (dua sisi)
Jumlah sampel (n)	100 pasang test case
Rata-rata selisih (Dijkstra – A*)	88,24 ms
Tingkat signifikansi (α)	0,05
Hasil keputusan	**H_0 ditolak: perbedaan signifikan secara statistik (p-value << 0,05)**

Dengan p-value jauh di bawah 0,05, H_0 ditolak. Keunggulan A* yang teramati bukan produk variasi acak, melainkan perbedaan performa yang nyata dan dapat diandalkan pada jaringan jalan Kota Medan.

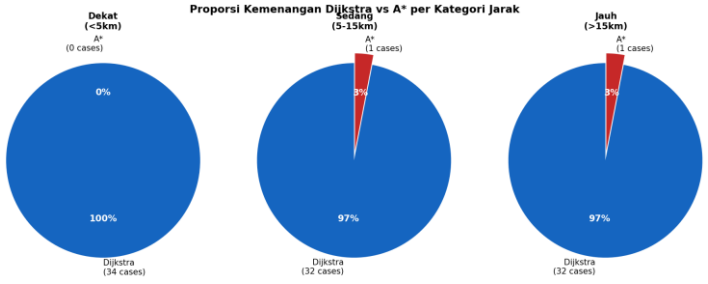
3.7 Perbandingan dengan Penelitian Terdahulu

Tabel 6 memposisikan penelitian ini terhadap studi-studi sebelumnya yang relevan.

Tabel 6. Perbandingan dengan Penelitian Terdahulu

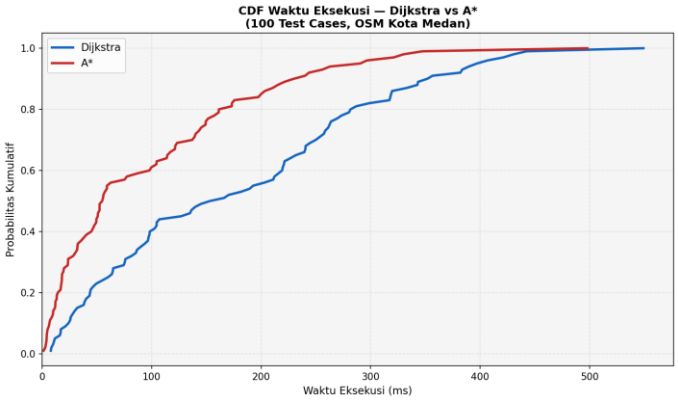
Penelitian	Dataset	Metode Ukur	Temuan Utama
Verma & Kumar (2024)	Grid simulasi 5×5	Nodes explored	A* eksplorasi lebih sedikit node dari Dijkstra
Elshaer et al. (2025)	Peta digital simulasi	Waktu eksekusi	A* lebih cepat, jalur identik panjangnya
Grujic & Grujic (2025)	OSM jaringan jalan perkotaan	Waktu eksekusi, optimasi rute	Dijkstra menghasilkan rute optimal pada jaringan jalan nyata
Penelitian ini (2026)	**OSM Medan, 42.692 node**	**Waktu, speedup, t-test**	**A* 2,83x lebih cepat, unggul 95/100 kasus, signifikan statistik**

Dibandingkan studi sebelumnya, penelitian ini menghadirkan tiga kebaruan utama: penggunaan data OSM nyata berskala besar dari Indonesia, stratifikasi kategori jarak yang mengungkap pola baru bahwa keunggulan A* justru terbesar pada jarak dekat, serta validasi inferensial melalui uji *paired t-test* yang memberikan jaminan statistik lebih kuat. Proporsi kemenangan per kategori dan distribusi kumulatif waktu eksekusi ditunjukkan pada Gambar 7 dan Gambar 8.



Gambar 7. Proporsi Kemenangan per Kategori

JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 3, No. 12 Tahun 2026
ISSN 3025-0919 (media online)
Hal 3075-3084



Gambar 8. Cumulative Distribution Function (CDF) Waktu Eksekusi Dijkstra vs A*

4. KESIMPULAN

Penelitian ini membandingkan kinerja algoritma Dijkstra dan A* pada jaringan jalan Kota Medan menggunakan data OpenStreetMap yang terdiri atas 42.692 simpul dan 99.736 segmen jalan. Pengujian dilakukan pada 100 pasang titik asal-tujuan yang dikelompokkan ke dalam tiga kategori jarak, dengan evaluasi mencakup waktu eksekusi, panjang jalur, jumlah node yang dikunjungi, dan speedup.

Kedua algoritma berhasil diimplementasikan dan terbukti menghasilkan panjang jalur yang identik pada seluruh 100 pasang titik yang diuji. Hal ini membuktikan bahwa penggunaan heuristik haversine yang bersifat *admissible* pada A* tidak mengurangi kualitas jalur yang dihasilkan. Dari sisi efisiensi waktu eksekusi, A* terbukti lebih unggul secara signifikan dengan rata-rata *speedup* 2,83x terhadap Dijkstra dan berhasil menang pada 95 dari 100 kasus pengujian. Keunggulan ini dikonfirmasi melalui uji *paired t-test* dengan p-value jauh di bawah 0,05, memastikan bahwa perbedaan yang teramati bukan merupakan kebetulan statistik.

Analisis per kategori jarak mengungkap pola yang konsisten, yaitu keunggulan A* paling besar pada jarak dekat dengan *speedup* rata-rata 3,77x, diikuti jarak sedang sebesar 3,03x, dan jarak jauh sebesar 1,68x. Penurunan keunggulan pada jarak jauh disebabkan oleh kompleksitas topologi jaringan jalan Kota Medan yang membuat rute aktual kerap menyimpang dari estimasi garis lurus fungsi heuristik. Ditemukan pula lima kasus anomali di mana Dijkstra lebih cepat dari A*, seluruhnya terjadi pada rute jarak jauh atau batas atas kategori sedang akibat fenomena *dead-end heuristic*, meskipun selisihnya tergolong kecil.

Berdasarkan temuan tersebut, A* direkomendasikan sebagai pilihan utama untuk sistem navigasi kendaraan pada jaringan jalan perkotaan, khususnya untuk rute jarak dekat hingga sedang. Untuk pengembangan selanjutnya, perlu dikaji penerapan fungsi heuristik yang lebih adaptif terhadap kondisi topologi lokal guna meningkatkan performa A* pada rute jarak jauh. Penelitian lanjutan juga dapat mempertimbangkan variabel dinamis seperti kepadatan lalu lintas dan kondisi jalan aktual sebagai parameter bobot tambahan dalam proses pencarian jalur.

REFERENCES

Al Bager, A. A., & Ahmed, A. S. (2021). Designing and implementing shortest and fastest paths A comparison of Bellman-Ford algorithm, A and Dijkstra's algorithms. *International Journal of Computer Trends and Technology*, *69*(5), 6–12. httpsdoi.org10.1444522312803IJCTT-V69I5P102

Aldhafferi, N. (2025). Time and Memory Trade-Offs in Shortest-Path Algorithms Across Graph Topologies: A*, Bellman–Ford, Dijkstra, AI-Augmented A* and a Neural Baseline. *Computers*, *14*(12), 545.

Chen, R. (2022). Bus routing system based on Dijkstra algorithm. *Advances in Economics, Business and Management Research*, *217*, 321–325.

Chu, L., Wang, Y., Li, S., Guo, Z., Du, W., Li, J., & Jiang, Z. (2024). Intelligent vehicle path planning based on optimized A-Star algorithm. *Sensors*, *24*(10), 3149.

JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 3, No. 12 Tahun 2026
ISSN 3025-0919 (media online)
Hal 3075-3084

Elshaer, A. M., Elmanfaloty, R. A., Abou-Bakr, E., Elrakaiby, M., & Saada, K. (2025). Exploring algorithmic efficiency of A-Star and Dijkstra for optimal route planning in green transportation. *International Journal of Intelligent Transportation Systems Research*, *23*(2), 1097–1107.

Grujic, Z., & Grujic, B. (2025). Optimal routing in urban road networks A graph-based approach using Dijkstra's algorithm. *Applied Sciences*, *15*(8), 4162.

Mienye, I. D., & Jere, N. (2025). Efficient route planning algorithm for special vehicles using large-scale road network data. *ISPRS International Journal of Geo-Information*, *14*(2), 71.

Milin, V., Stanivuk, T., Skoko, I., & Bulic, T. (2025). Dijkstra and A-Star algorithms for algorithmic optimization of maritime routes and offshore wind farm logistics. *Journal of Marine Science and Engineering*, *13*(10), 1863.

Sapundzhi, F., Danev, K., Ivanova, A., Popstoilov, M., & Georgiev, S. (2025). A performance comparison of shortest path algorithms in directed graphs. *Engineering Proceedings*, *100*(31), 1–12.

Singh, H., Gumber, S., & Rishu. (2025). A review and comparative analysis of fundamental shortest path algorithms. *International Journal for Research in Applied Science and Engineering Technology*, *13*(12).

Ugwoke, K. C., Nnanna, N. A., & Abdullahi, S. E. Y. (2025). A simulation-based review of classical, heuristic, and metaheuristic path planning algorithms. *Scientific Reports*, *15*, 12643.

Verma, G., & Kumar, A. (2024). Algorithmic pathfinding Comparing Dijkstra's and A algorithms in complex grid environment. *International Journal of Engineering Research and Development*, *20*(8), 70–79.

Vijayalaxmi, M. K. (2024). Finding the shortest path in the transport problem using Dijkstra algorithm. *International Journal of Engineering Research and Technology*, *13*(7).

Wang, R., Lu, Z., Jin, Y., & Liang, C. (2022). Application of A algorithm in intelligent vehicle path planning. *Mathematical Models in Engineering*, *8*(3), 82–90. httpsdoi.org10.21595mme.2022.22828