



## Implementasi Algoritma Shortest Path untuk Optimasi Rute pada Sistem Navigasi Lokasi

Sirlia Sahid<sup>1\*</sup>, Maissy Angelica Pakpahan<sup>2</sup>, Rifqi Putra Winanda<sup>3</sup>, Muhammad Raihansyah Lubis<sup>4</sup>, Adidtya Perdana<sup>5</sup>

<sup>1-5</sup> Universitas Negeri Medan, Indonesia

\*Penulis korespondensi: [sirliasahid@gmail.com](mailto:sirliasahid@gmail.com)<sup>1</sup>

**Abstract.** *The increasing complexity of urban road networks demands intelligent navigation systems capable of determining optimal routes efficiently. This research implements the Dijkstra Shortest Path algorithm to optimize route search on a location navigation system in Medan City. The system models a road network as a weighted graph comprising 57 strategic locations and over 90 road connections, represented using adjacency list data structures. The Dijkstra algorithm, implemented in Python using the heapq module for priority queue management, achieves an optimal time complexity of  $O((V+E) \log V)$ . The system features five main functions: shortest route search, popular routes, location listing, dynamic location addition, and dynamic road connection addition. System testing using a case study from Kualanamu Airport to the University of North Sumatra (USU) yielded an optimal route of 16.5 km through 4 road segments. Results demonstrate that the system successfully determines the most efficient route, provides accurate distance and travel time information for multiple transport modes (motorcycle, car, walking), and presents step-by-step journey guidance. This research contributes as a practical reference for applying shortest path algorithms in urban areas and serves as a foundation for developing more complex navigation applications in the future.*

**Keywords:** Algorithm; Dijkstra; Graph; Navigation; Shortest Path.

**Abstrak.** Meningkatnya kompleksitas jaringan jalan perkotaan menuntut sistem navigasi cerdas yang mampu menentukan rute optimal secara efisien. Penelitian ini mengimplementasikan algoritma Dijkstra Shortest Path untuk mengoptimasi pencarian rute pada sistem navigasi lokasi di Kota Medan. Sistem memodelkan jaringan jalan sebagai graf berbobot yang terdiri dari 57 lokasi strategis dan lebih dari 90 koneksi jalan, direpresentasikan menggunakan struktur data adjacency list. Algoritma Dijkstra diimplementasikan dalam Python menggunakan modul heapq untuk manajemen priority queue, mencapai kompleksitas waktu optimal  $O((V+E) \log V)$ . Sistem ini memiliki lima fungsi utama: pencarian rute terpendek, rute populer, tampilan lokasi, penambahan lokasi dinamis, dan penambahan koneksi jalan dinamis. Pengujian sistem menggunakan studi kasus dari Bandara Kualanamu ke Universitas Sumatera Utara (USU) menghasilkan rute optimal sepanjang 16,5 km melalui 4 ruas jalan. Hasil menunjukkan bahwa sistem berhasil menentukan rute paling efisien, memberikan informasi jarak dan estimasi waktu tempuh yang akurat untuk berbagai moda transportasi (motor, mobil, jalan kaki), serta menyajikan panduan perjalanan langkah demi langkah. Penelitian ini berkontribusi sebagai referensi nyata penerapan algoritma pencarian jalur terpendek di kawasan perkotaan Indonesia dan menjadi fondasi pengembangan aplikasi navigasi yang lebih kompleks di masa mendatang.

**Kata kunci:** Algoritma; Dijkstra; Graf; Jalur Terpendek; Navigasi.

### 1. LATAR BELAKANG

Di era digital saat ini, mobilitas masyarakat modern sangat bergantung pada kecepatan dan efisiensi. Kebutuhan untuk berpindah dari satu lokasi ke lokasi lain dalam waktu sesingkat mungkin telah mendorong perkembangan teknologi sistem navigasi yang masif. Namun, permasalahan utama yang sering muncul adalah bagaimana menentukan rute terbaik di tengah jaringan jalan yang semakin kompleks dan padat.

Pemilihan rute secara manual sering kali tidak akurat karena keterbatasan manusia dalam memproses variabel jarak dan waktu secara simultan. Di sinilah peran penting algoritma

pencarian rute terpendek (Shortest Path). Dalam ilmu komputer, algoritma seperti Dijkstra atau A\* (A-Star) sering digunakan untuk menyelesaikan masalah graf yang merepresentasikan peta dunia nyata.

Implementasi algoritma ini pada sistem navigasi bukan sekadar menemukan jalan, tetapi bagaimana melakukan optimasi agar konsumsi bahan bakar lebih efisien dan waktu tempuh menjadi minimal. Berdasarkan urgensi tersebut, penelitian ini mengkaji lebih dalam mengenai cara kerja dan implementasi algoritma shortest path dalam mengoptimalkan sistem navigasi lokasi di Kota Medan. Tujuan utama penelitian adalah mengimplementasikan algoritma Dijkstra untuk menemukan rute terpendek antara dua titik lokasi, menampilkan jalur optimal beserta estimasi jarak dan waktu, serta menganalisis efisiensi kinerja algoritma pada jaringan transportasi skala kota.

## **2. KAJIAN TEORITIS**

### **Dasar Optimasi Rute dan Pemodelan Graf**

Efisiensi dalam menentukan jalur transportasi menjadi faktor penentu utama untuk meminimalkan pengeluaran operasional serta mempercepat durasi pengiriman. Implementasi model pemrograman linear terbukti mampu memberikan solusi yang presisi dalam memetakan rute distribusi yang paling efektif (Hevlie et al., 2025). Secara teknis, jaringan jalan dipresentasikan ke dalam bentuk graf berbobot, di mana setiap lokasi atau kota diposisikan sebagai simpul (node) dan jalan penghubung sebagai sisi (edge). Penentuan rute terbaik dalam graf ini didasarkan pada perhitungan total bobot terkecil, baik itu berdasarkan jarak tempuh maupun estimasi waktu perjalanan (Rahmadi, Putri, & Ilmi, 2025). Pencarian rute paling efisien saat ini menjadi persoalan fundamental dalam pengembangan sistem informasi geografis, manajemen logistik, hingga perencanaan wilayah (Taneo, Ndun, Fallo, & Do'o, 2025). Permasalahan jalur terpendek memiliki urgensi tinggi karena berdampak langsung pada efisiensi penggunaan sumber daya, penghematan durasi perjalanan, serta pengurangan biaya operasional (Jelita, Fallo, & Miru, 2025). Di lingkungan perkotaan yang padat, tantangan ini semakin kompleks seiring bertambahnya jumlah titik lokasi dan hubungan antar jaringan jalan yang harus dikelola secara komputasional (Probrini, 2025). Oleh karena itu, integrasi antara pemodelan matematis dan data geospasial real-time menjadi pijakan utama dalam membangun aplikasi navigasi yang tangguh (Jumali, Rizki, Ulum, & Prianto, 2026).

### **Mekanisme dan Karakteristik Algoritma Dijkstra**

Algoritma Dijkstra merupakan salah satu teknik komputasi yang paling populer untuk memecahkan masalah lintasan terpendek pada graf dengan bobot positif. Cara kerjanya

mengandalkan prinsip *greedy*, yaitu sistem akan memilih jalur dengan beban paling rendah di setiap tahapannya hingga mencapai target (Amin & Hendrik, 2025). Namun, algoritma ini sering kali menghadapi kendala saat harus merespons perubahan kondisi jalan yang sangat dinamis atau data yang bersifat *real-time*. Kompleksitas jaringan jalan di wilayah perkotaan yang memiliki banyak persimpangan menjadi tantangan tersendiri bagi akurasi algoritma navigasi konvensional (Andini, Kultsum, Raihan, & Lestari, 2025).

Algoritma Dijkstra merupakan salah satu metode klasik yang paling banyak digunakan untuk menentukan lintasan minimum dari satu titik asal ke berbagai tujuan dalam graf berbobot positif (Probrini, 2025). Prinsip kerjanya didasarkan pada proses relaksasi nilai jarak secara iteratif, di mana simpul dengan bobot terkecil akan selalu diprioritaskan untuk diproses terlebih dahulu. Keunggulan utama dari pendekatan ini adalah kemampuannya dalam menjamin hasil yang optimal secara matematis tanpa bergantung pada estimasi atau perkiraan. Meskipun memberikan akurasi yang mutlak, Dijkstra cenderung memiliki kompleksitas komputasi yang tinggi dan memerlukan waktu eksekusi lebih lama saat diterapkan pada graf berskala besar (Angul, Fallo, Tanggo, Belo, & Hoar, 2025).

### **Analisis Algoritma Greedy Best-First Search**

Sebagai alternatif untuk situasi yang membutuhkan respons cepat, algoritma Greedy Best-First Search menawarkan strategi pengambilan keputusan berdasarkan solusi lokal terbaik di setiap langkahnya. Algoritma ini beroperasi dengan cara mengeksplorasi simpul yang secara heuristik dianggap paling dekat dengan target tujuan, seringkali menggunakan fungsi jarak tertentu. Pendekatan ini unggul dalam hal kecepatan eksekusi dan kesederhanaan proses karena tidak perlu menelusuri seluruh kemungkinan jalur (Dima, Hamzah, Tallo, & Fallo, 2025). Namun, sifatnya yang hanya fokus pada manfaat lokal berisiko menghasilkan rute yang sub-optimal dan tidak selalu menjamin tercapainya lintasan terpendek secara global (Angul et al., 2025).

### **Pengembangan Algoritma A-Star dan Teknologi Visual**

Sebagai bentuk optimasi dari metode sebelumnya, algoritma A-Star (A\*) menawarkan proses pencarian rute yang jauh lebih cepat dibandingkan Dijkstra. Keunggulan ini didapat karena penggunaan fungsi heuristik yang mampu memandu sistem untuk fokus pada jalur yang mengarah ke tujuan saja (Wijaya, Athallah, Noor'afina, Telnoni, & Budiwati, 2023). Saat ini, integrasi algoritma A\* dengan teknologi *Augmented Reality* (AR) berbasis GPS telah memungkinkan sistem navigasi memberikan petunjuk arah yang lebih interaktif dan intuitif. Pemanfaatan teknologi ini sangat membantu pengguna dalam memahami navigasi lokasi

secara visual melalui perangkat seluler mereka (Adam Puspabhuana, Andhika, Yudi Triyana, Muhammad Rifky Adhani, 2025).

### **Integrasi Algoritma A dan Teknik Metaheuristik**

Untuk mengatasi keterbatasan metode konvensional, algoritma A\* hadir sebagai solusi kompromi yang menggabungkan kekuatan sistematis Dijkstra dengan efisiensi heuristik (Alvito & Ikhsan, 2025). Selain algoritma pencarian jalur, penggunaan teknik metaheuristik seperti Particle Swarm Optimization (PSO) juga sering diterapkan untuk menyelesaikan permasalahan rute yang kompleks seperti Traveling Salesman Problem (Kurniawan, Farida, & Agustini, 2021). Penggunaan Algoritma Genetika yang diintegrasikan dengan Google Maps API terbukti mampu memberikan data jarak dan waktu tempuh yang lebih akurat secara real-time (Technology et al., 2025). Model optimasi yang menggabungkan keputusan alokasi distribusi dan penentuan rute minimum mampu memangkas jarak tempuh serta menurunkan biaya operasional transportasi secara signifikan (Jumali et al., 2026).

### **Kompleksitas Rute pada Permasalahan TSP dan Navigasi Kota**

Optimasi rute juga sering kali dihadapkan pada skenario di mana seorang pengantar harus mengunjungi beberapa titik sekaligus sebelum kembali ke lokasi awal, yang dikenal sebagai *Traveling Salesman Problem*. Permasalahan ini diselesaikan dengan memastikan setiap simpul dalam graf dikunjungi tepat satu kali melalui jalur yang paling singkat (M, 2023). Pendekatan lain seperti metode *Branch and Bound* sangat efektif digunakan untuk memangkas jalur-jalur yang tidak efisien dalam sistem pencarian navigasi wisata atau distribusi (Chandra et al., 2024). Di era digital, algoritma rute tidak hanya diterapkan pada jalan fisik, tetapi juga pada pengaturan lalu lintas data dalam jaringan komputer dan infrastruktur kota pintar (Hadi & Ibrahim, 2025). Terakhir, sistem navigasi yang tangguh harus mampu mengolah variabel jalan seperti sistem satu arah serta hambatan yang terjadi di persimpangan lampu lalu lintas (Grujic & Grujic, 2025).

### **Implementasi Teknologi Geospasial Modern**

Dalam implementasinya pada sistem navigasi, penghitungan jarak antar koordinat seringkali memanfaatkan rumus Haversine untuk memperoleh hasil yang akurat pada permukaan bumi yang melengkung (Asis, Mude, Astiani, & Kurnia Prihandani, 2024). Visualisasi proses pencarian rute pada data geospasial seperti OpenStreetMap membantu dalam menganalisis efisiensi algoritma secara lebih interaktif (Alvito & Ikhsan, 2025). Kesuksesan sistem navigasi lokasi pada akhirnya sangat bergantung pada ketepatan pemilihan algoritma yang disesuaikan dengan karakteristik data serta batasan sistem yang ada (Probrini, 2025).

### 3. METODE PENELITIAN

Berikut penjelasan metode penelitian dari desain sistem:

#### Desain Sistem

Sistem dirancang dengan arsitektur 3 lapisan: (1) Data Layer yang mencakup kelas Graf, 57 node, dan lebih dari 90 edge; (2) Algorithm Layer berisi fungsi Dijkstra, min-heap priority queue, dan rekonstruksi jalur; serta (3) Presentation Layer berupa tampilan konsol dengan 5 menu utama, yaitu Cari Rute, Rute Populer, Lihat Lokasi, Tambah Lokasi, dan Tambah Koneksi Jalan.

#### Struktur Data Graf

Graf direpresentasikan sebagai adjacency list berbasis dictionary Python. Setiap key adalah ID node (2 digit), sedangkan value-nya adalah list of tuple yang berisi (node\_tujuan, jarak\_km, nama\_jalan). Kelas Graph menyediakan metode tambah\_lokasi(), tambah\_jalan(), dan accessor seperti nama(), kategori(), dan alamat(). Representasi ini memungkinkan operasi penambahan node dan edge secara dinamis tanpa harus merekonstruksi ulang seluruh struktur data.

#### Data Lokasi Kota Medan

Sistem mencakup 57 lokasi strategis dalam 13 kategori: Bandara (1), Stasiun (1), Terminal (2), Mall (9), Pasar (3), Kuliner (3), Universitas (14), Rumah Sakit (5), Wisata (9), Hotel (5), Kantor (3), Pelabuhan (1), dan Lainnya. Setiap lokasi memiliki ID unik, nama, kategori, dan alamat lengkap. Koneksi antar lokasi disimpan dengan bobot jarak dalam satuan kilometer beserta nama ruas jalan yang menghubungkan kedua lokasi tersebut.

#### Implementasi Algoritma Dijkstra

Fungsi `dijkstra(graph, src, dst)` menggunakan modul `heapq` Python. Proses inisialisasi memberikan jarak `math.inf` kepada semua node, sedangkan node sumber diinisialisasi dengan jarak 0. Setiap iterasi mengambil node dengan jarak terkecil dari priority queue, memeriksa tetangga-tetangganya, dan memperbarui jarak apabila ditemukan jalur yang lebih pendek. Sistem menyimpan array `prev[]` sebagai predecessor dan `jalan[]` sebagai nama jalan untuk keperluan rekonstruksi jalur. Kompleksitas waktu algoritma ini adalah  $O((V+E) \log V)$ .

#### Rekonstruksi Jalur

Setelah proses Dijkstra selesai, jalur direkonstruksi dari array `prev[]` mulai dari node tujuan mundur ke node sumber, kemudian dibalik urutannya. Setiap segmen menghasilkan struktur `steps` yang berisi node tujuan segmen, jarak segmen, nama jalan, dan akumulasi jarak perjalanan. Struktur ini digunakan untuk menghasilkan panduan perjalanan langkah demi langkah yang mudah dipahami pengguna.

## **Fitur Sistem**

Sistem memiliki lima fitur utama: (1) Cari Rute Terpendek, yaitu input ID asal dan tujuan dengan output jalur optimal beserta estimasi waktu untuk 3 moda transportasi (motor 40 km/j, mobil 30 km/j, jalan kaki 5 km/j); (2) Rute Populer, berisi 12 rute preset yang sering digunakan; (3) Lihat Semua Lokasi dengan fitur filter berdasarkan kategori; (4) Tambah Lokasi Baru melalui input dinamis saat runtime; dan (5) Tambah Koneksi Jalan yang memungkinkan penambahan edge baru secara dinamis.

## **4. HASIL DAN PEMBAHASAN**

Berikut penjelasan hasil dan pembahasan dari hasil implementasi program yang telah dibuat:

### **Hasil Implementasi Program**

Pada tahap implementasi, sistem yang telah dikembangkan berhasil menampilkan hasil pencarian rute terpendek berdasarkan data yang diinputkan oleh pengguna. Program ini dirancang untuk menentukan jalur optimal dari titik awal ke tujuan dengan mempertimbangkan jarak minimum.

Berdasarkan hasil implementasi, sistem mampu menampilkan informasi secara lengkap, meliputi: (1) lokasi awal (source) dan lokasi tujuan (destination); (2) total jarak tempuh dalam satuan kilometer; (3) jumlah ruas jalan yang dilalui; (4) rincian jalur perjalanan secara berurutan; (5) panduan perjalanan langkah demi langkah; serta (6) estimasi waktu tempuh berdasarkan jenis kendaraan.

Sebagai contoh, sistem berhasil menemukan rute terpendek dari Bandara Kualanamu menuju Universitas Sumatera Utara (USU) dengan total jarak 16,5 km dan melalui 4 ruas jalan. Rute yang ditempuh adalah Bandara Kualanamu → Terminal Amplas → Medan Area (UMA) → Universitas Negeri Medan (UNIMED) → Universitas Sumatera Utara (USU). Fitur tambahan berupa panduan perjalanan membuat pengguna lebih mudah memahami alur perjalanan, termasuk jalan yang harus dilalui serta lokasi yang dilewati.

### **Contoh Kasus**

Sebagai bentuk pengujian sistem, digunakan sebuah studi kasus pencarian rute dari Bandara Kualanamu menuju Universitas Sumatera Utara (USU).

Input: Titik awal: Bandara Kualanamu; Titik tujuan: Universitas Sumatera Utara (USU).

**Tabel 1.** Rincian Rute Terpendek Bandara Kualanamu – USU.

No.	Segmen Rute	Jarak (km)	Akumulasi (km)
1	Bandara Kualanamu → Terminal Amplas	4,5	4,5
2	Terminal Amplas → Universitas Medan Area (UMA)	1,5	6,0
3	Universitas Medan Area → Universitas Negeri Medan	1,5	7,5
4	Universitas Negeri Medan → Universitas Sumatera Utara	4,0	11,5

*Sumber: Data penelitian (2026).*

Estimasi waktu tempuh yang dihasilkan sistem adalah: Motor (rata-rata 40 km/j)  $\pm 25$  menit; Mobil (rata-rata 30 km/j)  $\pm 33$  menit; Jalan kaki (5 km/j)  $\pm 198$  menit. Hasil tersebut menunjukkan bahwa sistem mampu: (1) menentukan rute paling efisien berdasarkan bobot jarak minimum; (2) memberikan informasi jarak dan waktu secara akurat untuk berbagai moda transportasi; dan (3) menyajikan panduan perjalanan yang membantu pengguna dalam menavigasi wilayah yang telah dipetakan.

Kompleksitas waktu  $O((V+E) \log V)$  yang dicapai sistem memastikan bahwa proses pencarian rute tetap cepat meskipun digunakan pada jaringan transportasi skala kota yang cukup padat dengan 57 node dan lebih dari 90 edge. Hasil ini konsisten dengan temuan Hadi & Ibrahim (2025) yang menyatakan bahwa Dijkstra merupakan pilihan optimal untuk graf berbobot positif dalam sistem navigasi.

Bagian ini memuat proses pengumpulan data, rentang waktu dan lokasi penelitian, dan hasil analisis data (yang dapat didukung dengan ilustrasi dalam bentuk tabel atau gambar, bukan data mentah, serta bukan dalam bentuk *printscreen* hasil analisis), ulasan tentang keterkaitan antara hasil dan konsep dasar, dan atau hasil pengujian hipotesis (jika ada), serta kesesuaian atau pertentangan dengan hasil penelitian sebelumnya, beserta interpretasinya masing-masing. Bagian ini juga dapat memuat implikasi hasil penelitian, baik secara teoritis maupun terapan. Setiap gambar dan tabel yang digunakan harus diacu dan diberikan penjelasan di dalam teks, serta diberikan penomoran dan sumber acuan. Berikut ini diberikan contoh tata cara penulisan subjudul, sub-subjudul, sub-sub-subjudul, dan seterusnya.

## **5. KESIMPULAN DAN SARAN**

Berdasarkan hasil perancangan dan implementasi yang dilakukan, beberapa kesimpulan penting dapat diambil. Algoritma Dijkstra yang diimplementasikan menggunakan Python dengan modul `heapq` terbukti sangat efektif dalam pencarian rute terpendek, mencapai kompleksitas waktu optimal  $O((V+E) \log V)$  sehingga proses pencarian tetap cepat pada jaringan transportasi skala kota. Struktur data graf berbobot berbasis adjacency list berhasil merepresentasikan jaringan jalan Kota Medan secara akurat dengan mengintegrasikan lebih dari 90 koneksi jalan dan 57 titik lokasi strategis ke dalam satu kesatuan data yang terorganisir. Sistem yang dikembangkan menawarkan fleksibilitas lebih besar dibanding model statis karena mendukung pembaruan data dinamis, serta mampu menghasilkan output berupa jarak dan estimasi waktu tempuh untuk berbagai moda transportasi. Pengembangan sistem ini diharapkan dapat berfungsi sebagai referensi nyata penerapan algoritma pencarian jalur terpendek di kawasan perkotaan Indonesia.

Beberapa rekomendasi untuk penelitian lanjutan adalah: (1) integrasi data real-time dengan API peta digital eksternal seperti Google Maps atau OpenStreetMap untuk memungkinkan bobot jalur berubah otomatis sesuai kondisi lalu lintas; (2) pengembangan antarmuka grafis (GUI) berbasis web maupun Android/iOS yang lebih interaktif dan ramah pengguna; (3) perluasan jangkauan geografis mencakup seluruh wilayah Sumatera Utara; serta (4) penambahan fitur perbandingan performa antara algoritma Dijkstra dengan algoritma lain seperti A\* atau Bellman-Ford untuk menemukan metode paling efisien pada karakteristik jalan di Indonesia.

## **UCAPAN TERIMA KASIH**

Penulis mengucapkan terima kasih kepada Bapak Adidtya Perdana, S.T., M.Kom selaku dosen pengampu mata kuliah Desain dan Analisis Algoritma yang telah memberikan bimbingan dan arahan dalam penyelesaian penelitian ini. Penelitian ini merupakan bagian dari tugas mini riset pada Program Studi Ilmu Komputer, Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Medan Tahun 2026.

**DAFTAR REFERENSI**

- Adam Puspabhuana, Andhika, Yudi Triyana, Muhammad Rifky Adhani, A. S. (2025). Komtekinfo journal, 102-114. <https://doi.org/10.35134/komtekinfo.v12i2.645>
- Alvito, P., & Ikhsan, M. (2025). Visualisasi dan perbandingan efisiensi algoritma A\*, Greedy, dan Dijkstra pada data OpenStreetMap Kota Medan. *Jurnal Riset Komputer*, 12(4), 2407-389. <https://doi.org/10.30865/jurikom.v12i4.8869>
- Amin, A., & Hendrik, B. (2025). Analisis penerapan algoritma Dijkstra dalam optimasi penentuan rute: Sebuah kajian literatur sistematis. *Journal of Education Research*, 6(1), 100-106. <https://doi.org/10.37985/jer.v6i1.2155>
- Andini, M., Kultsum, R. U., Raihan, M. H. R., & Lestari, S. (2025). 1746-16-4798-1-10-20250208 (1). 9(1), 290-302. <https://doi.org/10.52362/jisamar.v9i1.1746>
- Angul, A., Fallo, D., Tanggo, K. V., Belo, I. N. A., & Hoar, F. (2025). Implementasi algoritma Dijkstra dan Greedy dalam penyelesaian masalah rute terpendek. *Jurnal Kridatama Sains Dan Teknologi*, 7(01), 489-496. <https://doi.org/10.53863/kst.v7i01.1654>
- Asis, M. A., Mude, M. A., Astiani, R., & Kurnia Prihandani, S. N. (2024). Shortest route navigation indoors using digital maps. *Indonesian Journal of Data and Science*, 4(3), 169-175. <https://doi.org/10.56705/ijodas.v4i3.110>
- Chandra, R., Arifin Prasetyo, T., Sitorus, R. A. P., Situmorang, P. H., Hutasoit, Y. S. R., & Manik, T. E. (2024). Design and implementation of the shortest path navigation in Samosir District using branch and bound algorithm. *Jurnal RESTI*, 8(2), 242-249. <https://doi.org/10.29207/resti.v8i2.5585>
- Dima, J., Hamzah, M. S., Tallo, C. G., & Fallo, D. Y. A. (2025). Tinjauan literatur tentang pemanfaatan algoritma Greedy untuk pencarian jalur terpendek. *Jurnal Kridatama Sains Dan Teknologi*, 7(01), 519-528. <https://doi.org/10.53863/kst.v7i01.1683>
- Grujic, Z., & Grujic, B. (2025). Optimal routing in urban road networks: A graph-based approach using Dijkstra's algorithm. *Applied Sciences (Switzerland)*, 15(8). <https://doi.org/10.3390/app15084162>
- Hadi, H. M., & Ibrahim, I. M. (2025). A comprehensive review of shortest path algorithms for network routing. *Asian Journal of Research in Computer Science*, 18(3), 152-175. <https://doi.org/10.9734/ajrcos/2025/v18i3584>
- Hevlie, W., Nazry, S., Riza, F., Rizky, F., Gultom, Z. A., Haris, M., ... Barus, B. (2025). Model optimasi rute transportasi berbasis pemrograman linear. *Jurnal Sistem Informasi Triguna Dharma (JURSI TGD)*, 4(1), 75-81. <https://doi.org/10.53513/jursi.v4i1.10586>
- Jelita, F., Fallo, D., & Miru, Y. G. (2025). Optimalisasi rute menggunakan algoritma Dijkstra dan Greedy: Sebuah pendekatan komparatif. *Jurnal Kridatama Sains Dan Teknologi*, 7(01), 555-562. <https://doi.org/10.53863/kst.v7i01.1659>
- Jumali, M. A., Rizki, G., Ulum, M. B., & Prianto, Y. C. (2026). Model optimasi distribusi minimum biaya berbasis transportation problem dan shortest path. *Jurnal Teknik Industri Terintegrasi (JUTIN)*, 9(1 SE-Articles of Research), 328-333. Retrieved from <https://journal.universitaspahlawan.ac.id/index.php/jutin/article/view/52517>
- Kurniawan, M., Farida, & Agustini, S. (2021). Rute terpendek algoritma particle swarm optimization dan brute force untuk optimasi travelling salesman problem. *Jurnal Teknik Informatika*, 14(2), 191-200. Retrieved from <http://journal.uinjkt.ac.id/index.php/ti>

- M, P. M. P. (2023). Optimasi metode travelling salesman problem (TSP) dalam menyelesaikan kasus rute terpendek pengiriman barang. *MEANS (Media Informasi Analisa Dan Sistem)*, 8(1), 34-39. <https://doi.org/10.54367/means.v8i1.2637>
- Probrini, E. (2025). ... with Dijkstra's algorithm shortest route search with Dijkstra's algorithm: Pencarian rute terpendek dengan algoritma Dijkstra shortest route search with Dijkstra's ... *Jurnal/ Edu elektromatika*, 6(1), 50-57. Retrieved from <https://www.jurnal.sttp.ac.id/index.php/eduelektromatika/article/view/81>
- Rahmadi, D., Putri, T. N., & Ilmi, D. A. (2025). Optimizing travel routes between Jakarta and Surabaya using the Dijkstra algorithm. *Al-Aqlu: Jurnal Matematika, Teknik Dan Sains*, 3(2985-4369), 74-81. <https://doi.org/10.59896/aqlu.v3i1.140>
- Taneo, R. E., Ndun, R., Fallo, D. Y. A., & Do'o, F. (2025). Optimasi jalur terpendek menggunakan algoritma Dijkstra dan Greedy pada sistem informasi geografis. *Jurnal Kridatama Sains Dan Teknologi*, 7(01), 572-852. <https://doi.org/10.53863/kst.v7i01.1664>
- Technology, N., Baidawi, T., Kuswara, H., Wahyudi, E., Bina, U., Informatika, S., ... Routing, V. (2025). ALGORITMA GENETIKA BERBASIS GOOGLE MAPS API. 5(2), 57-65.
- Wijaya, D. R., Athallah, A., Noor'afina, T. N., Telnoni, P. A., & Budiwati, S. D. (2023). Cargo route optimization using shortest path algorithms: Runtime and validity comparison. *Journal of Computer Science*, 19(11), 1369-1379. <https://doi.org/10.3844/jcssp.2023.1369.1379>