

Penerapan Algoritma Dijkstra untuk Menentukan Rute Terpendek pada Jaringan Jalan Raya di Kota Jakarta

Teti Desyani S.kom., M.kom.¹ Riyan Setiawan²

e-mail : [1dosen00839@unpam.ac.id](mailto:¹dosen00839@unpam.ac.id), [2riyanstwn1@gmail.com](mailto:²riyanstwn1@gmail.com)

Teknik Informatika, Universitas Pamulang, Tangerang selatan, Indonesia

Submitted Date : June 29,2024

Reviewed Date : -

Revised Date : July 2,2024

Accepted Date :-

Abstract

Due to increasing mobility in the city of Jakarta, traffic congestion is increasing. To reduce the impact of traffic congestion, this research implements the Dijkstra algorithm to find the shortest route on the Jakarta highway network. This algorithm was chosen because of its high efficiency in solving shortest path problems. The data used in this research includes Jakarta's main road network and average travel time. The research results show that the application of the Dijkstra algorithm can reduce travel time significantly compared to a randomly selected route.

Keywords: Dijkstra algorithm, shortest route, road network, Jakarta, route optimization.

Abstrak

Karena meningkatnya mobilitas di kota Jakarta, kemacetan lalu lintas semakin meningkat. Untuk mengurangi dampak kemacetan lalu lintas, penelitian ini mengimplementasikan algoritma Dijkstra untuk mencari rute terpendek pada jaringan jalan raya Jakarta. Algoritma ini dipilih karena efisiensinya yang tinggi dalam menyelesaikan permasalahan jalur terpendek. Data yang digunakan dalam penelitian ini meliputi jaringan jalan utama Jakarta dan waktu tempuh rata-rata. Hasil penelitian menunjukkan bahwa penerapan algoritma Dijkstra dapat mengurangi waktu perjalanan secara signifikan dibandingkan dengan rute yang dipilih secara acak.

Kata Kunci : Algoritma Dijkstra, rute terpendek, jaringan jalan, Jakarta, optimalisasi rute.

Pendahuluan

Perjalanan di kota besar seperti Jakarta sangat dipengaruhi oleh kemacetan lalu lintas. Kemacetan lalu lintas ini tidak hanya membuang waktu dan bahan bakar, namun juga meningkatkan polusi udara dan stres pengemudi. Oleh karena itu, sangat penting mencari solusi untuk mengurangi kemacetan. Solusi yang tepat adalah menentukan rute perjalanan yang paling efisien.

Algoritma Dijkstra adalah salah satu algoritma yang paling terkenal dan efektif untuk mencari rute terpendek dalam graf berbobot berarah. Algoritma ini ditemukan oleh Edsger W. Dijkstra pada tahun 1956 dan sejak itu telah digunakan dalam berbagai aplikasi seperti jaringan komputer, navigasi GPS, dan optimasi rute. Tujuan dari penelitian ini adalah untuk menerapkan algoritma Dijkstra pada jaringan jalan raya Jakarta dan mengevaluasi efisiensinya dalam mengurangi waktu perjalanan.

Metodologi

Penelitian ini melibatkan beberapa tahapan utama: pengumpulan data, pembuatan grafik, implementasi algoritma Dijkstra, dan analisis hasil. Masing-masing fase tersebut dijelaskan secara rinci di bawah ini.

Pengumpulan Data

Data yang dikumpulkan dalam studi ini mencakup peta jaringan jalan utama Jakarta dan perkiraan waktu tempuh rata-rata antar titik tertentu. Data tersebut diperoleh dari jasa transportasi, aplikasi pemetaan digital, dan observasi langsung di lapangan. Peta jaringan jalan berisi informasi tentang node (persimpangan jalan) dan edge (jalan yang menghubungkan persimpangan).

Pembentukan Graf

Jaringan jalan diubah menjadi diagram dengan simpul mewakili persimpangan jalan dan tepinya mewakili jalan yang menghubungkan persimpangan tersebut. Bobot tepi mewakili waktu tempuh rata-rata berdasarkan data yang dikumpulkan, dan grafik yang dihasilkan digunakan sebagai masukan pada algoritma Dijkstra.

Implementasi Algoritma Dijkstra

Algoritma Dijkstra diterapkan untuk menentukan rute terpendek dari suatu titik awal sampai suatu tujuan. Algoritma ini bekerja dengan memperbarui secara iteratif jarak terpendek yang diketahui dari titik awal ke semua titik lainnya. Berikut langkah-langkah utama algoritma Dijkstra:

1. Menginisialisasi jarak ke semua node lainnya hingga tak terhingga, kecuali jarak ke node awal yang diinisialisasi ke nol.
2. Tambahkan semua node ke himpunan yang belum dikunjungi.
3. Pilih node dengan jarak terpendek dari himpunan yang belum dikunjungi.
4. Perbarui jarak node yang dipilih ke tetangganya jika jalur baru lebih pendek.
5. Tandai node yang dipilih sebagai telah dikunjungi dan hapus node tersebut dari grup yang belum dikunjungi.
6. Ulangi langkah 3 hingga 5 hingga semua node diakses.

Analisis Hasil

Hasil penerapan algoritma dianalisis untuk mengetahui seberapa efektif algoritma Dijkstra dalam mengurangi waktu tempuh dibandingkan metode pemilihan rute lainnya. Analisis dilakukan dengan membandingkan waktu tempuh yang ditentukan oleh algoritma Dijkstra dengan waktu tempuh suatu rute yang dipilih secara acak.

Hasil dan Pembahasan

Hasil penerapan algoritma Dijkstra pada jaringan jalan Jakarta menunjukkan bahwa algoritma tersebut dapat menentukan rute terpendek dengan waktu tempuh yang lebih efisien. Beberapa skenario pengujian dilakukan untuk menguji keakuratan dan efisiensi algoritma Dijkstra. Berikut hasil pengujian dan pembahasannya.

Skenario Tes 1: Asal dan tujuan dekat.

Dalam skenario pertama, titik awal dan tujuan dipilih berdekatan. Misalnya jarak node A ke node B di pusat kota relatif pendek. Hasil pengujian menunjukkan algoritma Dijkstra mampu menemukan rute terpendek dengan waktu tempuh 15 menit dibandingkan rute acak yang membutuhkan waktu 25 menit. Perbedaan waktu tempuh ini disebabkan oleh kemampuan algoritma Dijkstra dalam memilih jalan alternatif yang lebih sedikit kemacetannya.

Titik Awal	Titik Tujuan	Waktu Tempuh (Algoritma Dijkstra)	Waktu Tempuh (Rute Acak)
A	B	15 menit	25 menit

Skenario pengujian 2: Asal dan tujuan Panjang

Pada skenario kedua, asal dan tujuan yang berjauhan akan dipilih. Misalnya dari node A ke node C. Algoritma Dijkstra mencari rute terpendek dengan waktu tempuh 20 menit, sedangkan rute acak membutuhkan waktu 35 menit. Penghematan waktu ini disebabkan algoritma Dijkstra mampu menghindari titik kemacetan dan memilih jalan dengan waktu tempuh lebih stabil.

Titik Awal	Titik Tujuan	Waktu Tempuh (Algoritma Dijkstra)	Waktu Tempuh (Rute Acak)
A	C	20 menit	35 menit

Skenario pengujian 3: Rute dengan banyak persimpangan

Pada skenario ketiga, rute yang dipilih mencakup misalnya dari rute B ke simpul D. Algoritma Dijkstra dapat mencari rute terpendek dengan waktu tempuh 10 menit, sedangkan rute acak membutuhkan waktu 18 menit. Hasil tersebut menunjukkan bahwa algoritma Dijkstra sangat efektif dalam menangani jaringan jalan yang kompleks dengan banyak persimpangan.

Titik Awal	Titik Tujuan	Waktu Tempuh (Algoritma Dijkstra)	Waktu Tempuh (Rute Acak)
B	D	10 menit	18 menit

Pembahasan lebih lanjut

Keuntungan utama algoritma Dijkstra adalah ia secara efisien mengambil rute terpendek dengan mempertimbangkan semua kemungkinan jalur menemukan. Hal ini sangat

penting terutama di kota-kota besar seperti Jakarta, dimana kemacetan lalu lintas dan perubahan kondisi lalu lintas merupakan hal yang biasa.

1. Efisiensi algoritma: Algoritma Dijkstra menunjukkan efisiensi yang tinggi dalam menentukan rute terpendek. Dalam setiap skenario pengujian, algoritme secara konsisten menghasilkan rute dengan waktu tempuh lebih pendek dibandingkan rute acak. Efisiensi ini sangat bermanfaat bagi pengemudi yang ingin menghindari kemacetan dan menghemat waktu perjalanan.
2. Implementasi dalam Sistem Navigasi: Hasil penelitian ini menunjukkan potensi besar untuk mengintegrasikan algoritma Dijkstra ke dalam sistem navigasi berbasis GPS. Data lalu lintas waktu nyata memungkinkan sistem navigasi menyediakan rute optimal secara lebih akurat kepada pengguna dan merespons dengan cepat terhadap perubahan kondisi lalu lintas.
3. Kemungkinan pengembangan lebih lanjut: Mengintegrasikan data lalu lintas waktu nyata dapat meningkatkan keakuratan prediksi waktu perjalanan. Selain itu, penggunaan algoritme pengoptimalan lainnya, seperti algoritme A*, yang memperhitungkan heuristik tertentu, dapat memberikan hasil yang lebih baik dalam beberapa kasus.
4. Keterbatasan dan Tantangan: Algoritma Dijkstra efektif, namun implementasinya memerlukan data yang akurat dan lengkap tentang jaringan jalan dan waktu tempuh. Tantangan lainnya adalah menghadapi data lalu lintas yang dinamis dan perubahan kondisi jalan yang tidak terduga.

Di bawah ini contoh visualisasi hasil tes pada peta Jakarta :

Dalam visualisasi ini, rute yang dihasilkan oleh algoritma Dijkstra diwakili oleh garis biru, dan rute acak diwakili oleh garis merah. Jelas bahwa rute yang dihasilkan oleh algoritma Dijkstra lebih pendek dan menghindari titik kemacetan besar.

Analisis Efektivitas dan Efisiensi

Algoritma Dijkstra terbukti efektif dan efisien dalam menciptakan rute terpendek pada jaringan jalan kota Jakarta. Hal ini dapat dilihat dari beberapa faktor berikut :

- Kecepatan: Algoritma Dijkstra dapat dengan cepat menemukan rute terpendek, bahkan pada jaringan jalan yang besar dan kompleks.
- Akurasi: Algoritma Dijkstra menghasilkan rute terpendek yang hanya berbeda sedikit dengan rute yang disarankan oleh aplikasi navigasi.

- Kesederhanaan: Algoritma Dijkstra mudah diimplementasikan dan dipahami.

Faktor-Faktor yang Mempengaruhi Hasil Faktor-faktor berikut dapat mempengaruhi hasil algoritma Dijkstra :

- Keakuratan data: Algoritma Dijkstra memerlukan data jaringan jalan yang akurat dan terkini.
- Kompleksitas jaringan: Algoritma Dijkstra membutuhkan waktu untuk menemukan rute terpendek dalam jaringan jalan yang kompleks.
- Kondisi lalu lintas: Algoritma Dijkstra tidak memperhitungkan kondisi lalu lintas saat ini seperti kemacetan atau kecelakaan.

Di bawah ini contoh visualisasi hasil tes pada peta Jakarta.

Dalam visualisasi ini, rute yang dihasilkan oleh algoritma Dijkstra diwakili oleh garis biru, dan rute acak diwakili oleh garis merah. Jelas bahwa rute yang dihasilkan oleh algoritma Dijkstra lebih pendek dan menghindari titik kemacetan besar.

Kesimpulan

Penelitian ini membuktikan bahwa algoritma Dijkstra efektif digunakan untuk menentukan rute terpendek pada jaringan jalan raya Jakarta. Dengan menerapkan algoritma ini, pengguna jalan dapat menghemat waktu perjalanan secara signifikan. Selain itu, implementasi ini juga dapat berkontribusi pada pengembangan sistem navigasi yang lebih cerdas dan efisien.

Saran

Untuk penelitian lebih lanjut, kami menyarankan untuk mengintegrasikan data situasi lalu lintas secara real-time untuk meningkatkan akurasi prediksi waktu perjalanan. Selain itu, pengujian pada jaringan jalan yang lebih luas dan kompleks diperlukan untuk memvalidasi temuan penelitian ini. Integrasi dengan teknologi pembelajaran mesin juga dapat menjadi pilihan untuk meningkatkan prediksi rute dengan mempertimbangkan pola lalu lintas yang dinamis.

Daftar Pustaka

1. Dijkstra, E. W. (1959). "A note on two problems in connexion with graphs". *Numerische Mathematik*, 1, 269–271.

2. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). "Introduction to Algorithms". MIT Press.
3. BPS DKI Jakarta. (2020). "Statistik Transportasi DKI Jakarta 2020". Badan Pusat Statistik DKI Jakarta.
4. Google Maps API. (2021). "Google Maps Traffic Data". Google Inc.
5. Newman, M. (2010). "Networks: An Introduction". Oxford University Press.
6. Taha, H. A. (2011). "Operations Research: An Introduction". Pearson.
7. Zhan, F. B., & Noon, C. E. (1998). "Shortest path algorithms: An evaluation using real road networks". *Transportation Science*, 32(1), 65-73.
8. Department of Transportation Jakarta. (2020). "Jakarta Traffic Data Report 2020". Department of Transportation Jakarta