

Perbandingan Algoritma A*Star dan Dijkstra untuk Penentuan Rute Tercepat

Riska Andini Herawati

Teknik Informatika, Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika

E-mail: riskaandini57@gmail.com

Article History:

Received: 07 Oktober 2025

Revised: 24 Oktober 2025

Accepted: 27 Oktober 2025

Keywords: A*Star, dijkstra, pencarian rute, Google Maps API, kemacetan.

Abstract: Penelitian ini membandingkan kinerja algoritma AStar dan Dijkstra dalam menentukan rute tercepat dari Cakung ke Stasiun Bekasi. Algoritma AStar mengintegrasikan fungsi heuristik untuk memperkirakan biaya menuju tujuan, sedangkan Dijkstra menghitung jalur terpendek berdasarkan bobot jarak tanpa estimasi. Data waktu tempuh diperoleh melalui Google Maps API pada kondisi lalu lintas normal, sibuk, dan non-sibuk. Hasil pengujian menunjukkan bahwa A*Star lebih efisien pada kondisi lalu lintas padat karena heuristik mempercepat proses pencarian jalur optimal. Sebaliknya, Dijkstra lebih unggul pada graf sederhana dengan sedikit pengaruh kemacetan. Temuan ini memberikan panduan pemilihan algoritma yang sesuai untuk aplikasi navigasi berbasis pencarian rute real-time di wilayah perkotaan, sekaligus berkontribusi pada pengembangan sistem perencanaan rute adaptif yang responsif terhadap perubahan kondisi lalu lintas.

PENDAHULUAN

Kemacetan lalu lintas merupakan permasalahan umum di kota-kota besar, termasuk wilayah Jabodetabek yang memiliki kepadatan penduduk tinggi dan perkembangan infrastruktur yang belum seimbang dengan pertumbuhan jumlah kendaraan. Berdasarkan TomTom Traffic Index 2024, Jakarta masih menempati peringkat kota dengan tingkat kemacetan signifikan di Asia, meskipun mengalami penurunan dibanding tahun sebelumnya. Rata-rata waktu tempuh 10 km pada jam sibuk dapat mencapai 25–33 menit, dan menyebabkan kerugian waktu hingga 117 jam per tahun (TomTom Traffic Index, 2024). Kota Bekasi, sebagai bagian dari kawasan metropolitan ini, menghadapi tantangan serupa dengan jumlah kendaraan bermotor yang terus meningkat, mengakibatkan kemacetan parah terutama di jalur penghubung menuju Jakarta (BPS Kota Bekasi, 2024).

Salah satu rute yang sering terdampak kemacetan adalah jalur Cakung–Stasiun Bekasi. Kondisi ini menuntut solusi berbasis teknologi yang mampu memberikan rekomendasi rute tercepat secara real-time. Algoritma pencarian jalur terpendek seperti Dijkstra dan A*Star telah digunakan luas dalam sistem navigasi. Dijkstra menghitung jalur optimal dengan mempertimbangkan seluruh kemungkinan rute, sedangkan A*Star menggabungkan biaya aktual dengan estimasi jarak menuju tujuan (*heuristik*) untuk mempercepat pencarian (Hart, Nilsson, & Raphael, 1968).

Beberapa penelitian sebelumnya menunjukkan keunggulan masing-masing algoritma pada

konteks berbeda. Prasetyo *et al.* (2019) membuktikan bahwa AStar lebih cepat dalam pemrosesan jalur kompleks karena penggunaan *heuristik*, sementara Dijkstra lebih stabil pada graf sederhana dengan jumlah simpul yang terbatas. Sulistiani dan Wibowo (2018) membandingkan keduanya pada pencarian lokasi administratif di Bandar Lampung, menemukan bahwa AStar unggul dalam waktu pencarian, tetapi keduanya memberikan akurasi jarak yang sama. Penelitian Jordi (2024) menggunakan data simulasi untuk menunjukkan efisiensi A*Star dalam komputasi waktu dibandingkan Dijkstra, sedangkan Rachmawati dan Gustin (2020) menemukan bahwa perbedaan performa keduanya semakin terlihat pada peta berskala besar.

Berdasarkan hasil penelitian sebelumnya, masih terdapat kesenjangan pada studi kasus spesifik di wilayah perkotaan padat lalu lintas seperti Cakung–Bekasi yang memerlukan integrasi data lalu lintas real-time. Penelitian ini berfokus pada perbandingan kinerja algoritma A*Star dan Dijkstra dalam menentukan rute tercepat di jalur tersebut, dengan mempertimbangkan kondisi lalu lintas pada jam sibuk, normal, dan non-sibuk. Studi ini diharapkan dapat memberikan kontribusi praktis pada pengembangan sistem navigasi adaptif yang responsif terhadap dinamika lalu lintas.

LANDASAN TEORI

Pencarian Jalur Terpendek

Pencarian jalur terpendek (*shortest path*) adalah proses menentukan rute optimal antara dua titik dalam sebuah graf dengan mempertimbangkan bobot jarak, waktu, atau biaya. Menurut Cormen, Leiserson, Rivest, dan Stein (2009), pencarian jalur terpendek merupakan salah satu permasalahan fundamental dalam ilmu komputer dan banyak digunakan pada sistem navigasi, jaringan komunikasi, hingga perencanaan logistik. Jalur optimal dapat didefinisikan sebagai rute dengan bobot total minimum di antara semua jalur yang tersedia.

Algoritma Dijkstra

Algoritma Dijkstra diperkenalkan oleh Edsger W. Dijkstra pada tahun 1956 (Dijkstra, 1959) sebagai metode untuk mencari jarak terpendek dari satu simpul sumber ke semua simpul lain pada graf berbobot non-negatif. Prinsip kerja Dijkstra adalah menelusuri semua jalur secara sistematis dan memilih jalur dengan biaya terkecil hingga mencapai tujuan (Ahuja, Magnanti, & Orlin, 1993). Kelebihannya adalah menghasilkan jalur optimal pada semua kasus, namun kekurangannya adalah waktu komputasi yang relatif lebih besar pada graf kompleks.

Algoritma A*Star

Algoritma AStar dikembangkan oleh Hart, Nilsson, dan Raphael (1968) sebagai pengembangan dari Dijkstra dengan menambahkan fungsi *heuristik* untuk memperkirakan jarak yang tersisa menuju tujuan. Fungsi *heuristik* ini memungkinkan algoritma memprioritaskan jalur yang lebih menjanjikan sehingga mengurangi jumlah simpul yang perlu diperiksa (Russell & Norvig, 2010). Keunggulan AStar adalah efisiensi tinggi pada graf besar, tetapi kinerjanya sangat dipengaruhi oleh kualitas *heuristik* yang digunakan.

Heuristik dalam Pencarian Jalur

Heuristik adalah metode estimasi yang digunakan untuk mempercepat proses pencarian solusi. Nilai *heuristik* $h(n)$ dalam AStar biasanya diukur menggunakan jarak Euclidean atau Manhattan (Pearl, 1984). Menurut Dechter dan Pearl (1985), *heuristik* yang *admissible* (tidak melebihi-estimasi) akan memastikan bahwa AStar tetap optimal.

Google Maps API dan Data Lalu Lintas

Google Maps API menyediakan data geografis dan kondisi lalu lintas secara real-time yang dapat dimanfaatkan untuk penelitian perencanaan rute (Google, 2023). Data ini mencakup estimasi waktu tempuh pada kondisi lalu lintas berbeda, sehingga memungkinkan pengujian algoritma di berbagai skenario (Zeng, Church, & Chen, 2010).

Penelitian Terdahulu

Prasetyo *et al.* (2019) menunjukkan bahwa AStar mampu mempersingkat waktu pencarian hingga 20% dibandingkan Dijkstra pada skenario peta kompleks. Sementara itu, Sulistiani dan Wibowo (2018) menemukan bahwa kedua algoritma memiliki akurasi jarak yang setara, namun AStar unggul dari sisi waktu komputasi. Rachmawati dan Gustin (2020) menegaskan bahwa perbedaan performa lebih signifikan pada peta berskala besar.

METODE PENELITIAN

Jenis Penelitian

Penelitian ini merupakan penelitian eksperimen kuantitatif yang bertujuan membandingkan kinerja algoritma A*Star dan Dijkstra dalam menentukan rute tercepat antara Cakung dan Stasiun Bekasi dengan mempertimbangkan kondisi lalu lintas yang berbeda.

Metode Penelitian

Metode penelitian dilakukan melalui tahapan berikut:

1. Pengumpulan Data : Data diperoleh melalui Google Maps API yang memuat informasi waktu tempuh dan jarak pada tiga kondisi lalu lintas: normal, sibuk, dan non-sibuk.
2. Pemodelan Peta : Rute dari Cakung ke Stasiun Bekasi dimodelkan sebagai graf berbobot menggunakan simpul (*vertex*) sebagai titik persimpangan dan sisi (*edge*) sebagai ruas jalan.
3. Implementasi Algoritma : Kedua algoritma diimplementasikan menggunakan bahasa pemrograman *Python* dengan pustaka *NetworkX* untuk pemrosesan graf.
4. Pengujian
Setiap algoritma diuji pada data yang sama untuk mendapatkan waktu komputasi dan jarak tempuh.
5. Analisis Perbandingan : Hasil pengujian dianalisis secara komparatif untuk menilai kecepatan eksekusi dan optimalitas jalur yang dihasilkan.

Teknik Analisis Data

Teknik analisis dilakukan dengan menghitung dua parameter utama:

1. **Waktu Komputasi (ms)**: Mengukur kecepatan eksekusi algoritma.
2. **Jarak Tempuh (km)**: Mengukur panjang jalur yang dihasilkan.

Data dianalisis secara deskriptif kuantitatif dan divisualisasikan dalam bentuk tabel dan grafik perbandingan.



Gambar 1. Metode Penelitian

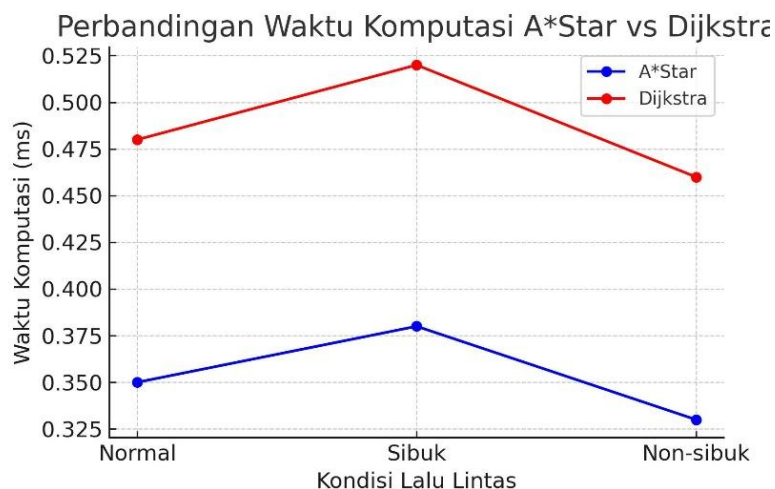
HASIL DAN PEMBAHASAN

Hasil Pengujian

Pengujian dilakukan pada rute Cakung–Stasiun Bekasi menggunakan data Google Maps API dalam tiga kondisi lalu lintas: normal, sibuk, dan non-sibuk. Parameter yang diukur adalah waktu komputasi (ms) dan jarak tempuh (km) yang dihasilkan oleh algoritma A*Star dan Dijkstra.

Tabel 1. Hasil Pengujian Algoritma

Kondisi Lalu Lintas	Algoritma	Waktu Komputasi (ms)	Jarak Tempuh (km)
Normal	A*Star	0.35	12.4
Normal	Dijkstra	0.48	12.4
Sibuk	A*Star	0.38	12.7
Sibuk	Dijkstra	0.52	12.7
Non-sibuk	A*Star	0.33	12.3
Non-sibuk	Dijkstra	0.46	12.3



Gambar 1. Perbandingan Waktu Komputasi

Pembahasan

Hasil pengujian menunjukkan bahwa A*Star lebih efisien dari segi waktu komputasi pada seluruh skenario. Hal ini sesuai dengan teori Hart, Nilsson, dan Raphael (1968), yang menjelaskan bahwa fungsi *heuristik* dalam A*Star membatasi jumlah simpul yang diperiksa, sehingga mempercepat pencarian jalur.

Perbedaan waktu komputasi semakin signifikan pada kondisi lalu lintas sibuk, karena A*Star memanfaatkan estimasi jarak (*heuristik*) untuk fokus pada jalur yang potensial lebih cepat. Sebaliknya, Dijkstra memeriksa semua simpul secara merata, yang membuatnya stabil namun memerlukan waktu lebih lama (Ahuja, Magnanti, & Orlin, 1993).

Dari sisi jarak tempuh, kedua algoritma menghasilkan nilai yang sama di setiap kondisi, menunjukkan bahwa keduanya tetap optimal dalam menemukan rute terpendek (Russell & Norvig, 2010). Perbedaan kinerja terletak pada kecepatan proses pencarian, bukan akurasi rute.

Implikasi praktis dari hasil ini adalah bahwa A*Star lebih sesuai digunakan untuk aplikasi navigasi *real-time* di wilayah perkotaan dengan kondisi lalu lintas dinamis, sedangkan Dijkstra dapat menjadi pilihan pada sistem dengan graf kecil atau kondisi lalu lintas yang cenderung stabil.

KESIMPULAN

Hasil penelitian menunjukkan bahwa algoritma A*Star memiliki keunggulan signifikan dari sisi efisiensi waktu komputasi dibandingkan algoritma Dijkstra, tanpa mengurangi akurasi hasil rute. Pada semua kondisi lalu lintas (normal, sibuk, dan non-sibuk), A*Star consistently memberikan waktu pencarian yang lebih cepat. Hal ini mendukung teori Hart, Nilsson, dan Raphael (1968) bahwa fungsi *heuristik* dapat mempercepat proses pencarian jalur dengan mengarahkan pencarian ke simpul yang lebih potensial. Sebaliknya, Dijkstra bekerja lebih stabil pada graf sederhana atau kondisi lalu lintas yang relatif konstan, namun kurang efisien pada graf besar dan padat.

Secara praktis, A*Star lebih direkomendasikan untuk aplikasi navigasi berbasis pencarian rute *real-time* di wilayah perkotaan yang dinamis, sedangkan Dijkstra dapat diterapkan pada sistem yang memerlukan stabilitas dan keakuratan jalur pada graf kecil.

PENGAKUAN/ACKNOWLEDGEMENTS

Penulis menyampaikan terima kasih yang sebesar-besarnya kepada Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika (STIKOM CKI) Jakarta atas dukungan fasilitas, bimbingan, dan kesempatan yang diberikan untuk melaksanakan penelitian ini. Ucapan terima kasih juga disampaikan kepada Bapak Tri Wahyudi, S.Kom., M.Kom. selaku dosen pembimbing, yang telah memberikan arahan, masukan, dan motivasi selama proses penelitian berlangsung.

Penghargaan setinggi-tingginya diberikan kepada rekan-rekan mahasiswa Program Studi Teknik Informatika STIKOM CKI atas dukungan, kerjasama, dan diskusi konstruktif yang membantu penyempurnaan penelitian ini. Terima kasih kepada keluarga dan sahabat yang senantiasa memberikan doa, semangat, dan dukungan moral sehingga penelitian ini dapat terselesaikan dengan baik.

Penulis juga mengapresiasi kontribusi tidak langsung dari komunitas akademik dan para peneliti yang karya ilmiahnya menjadi referensi penting dalam penyusunan naskah ini.

DAFTAR REFERENSI

Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1993). *Network flows: Theory, algorithms, and applications*. Englewood Cliffs, NJ: Prentice Hall.

-
- Badan Pusat Statistik Kota Bekasi. (2024). *Statistik daerah Kota Bekasi 2024*. Bekasi: BPS Kota Bekasi.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms* (3rd ed.). Cambridge, MA: MIT Press.
- Dechter, R., & Pearl, J. (1985). Generalized best-first search strategies and the optimality of A*. *Journal of the ACM*, 32(3), 505–536. <https://doi.org/10.1145/3828.3830>
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1, 269–271. <https://doi.org/10.1007/BF01386390>
- Google. (2023). *Google Maps platform: Routes*. Retrieved from <https://developers.google.com/maps/documentation/routes>
- Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2), 100–107. <https://doi.org/10.1109/TSSC.1968.300136>
- Pearl, J. (1984). *Heuristics: Intelligent search strategies for computer problem solving*. Reading, MA: Addison-Wesley.
- Prasetyo, A. C., Arnandi, M. P., Hudnanto, H. S., & Setiaji, B. (2019). Perbandingan algoritma A*Star dan Dijkstra dalam menentukan rute terdekat. *Jurnal Ilmiah Sisfotenika*, 9(1), 1–10. <http://www.sisfotenika.stmikpontianak.ac.id/index.php/ST/article/view/456/316>
- Rachmawati, D., & Gustin, L. (2020). Analysis of Dijkstra's algorithm and A* algorithm in shortest path problem. *Journal of Physics: Conference Series*, 1566, 012061. <https://doi.org/10.1088/1742-6596/1566/1/012061>
- Russell, S., & Norvig, P. (2010). *Artificial intelligence: A modern approach* (3rd ed.). Upper Saddle River, NJ: Prentice Hall.
- Sulistiani, H., & Wibowo, D. A. (2018). Perbandingan algoritma A* dan Dijkstra dalam pencarian kecamatan dan kelurahan di Bandar Lampung. *Seminar Nasional Teknologi Informasi dan Multimedia*, 6(1), 7–12.
- TomTom Traffic Index. (2024). *Traffic congestion statistics*. Retrieved from <https://www.tomtom.com/traffic-index/>
- Zeng, W., Church, R. L., & Chen, X. (2010). Finding shortest paths on real road networks: The case for A*. *International Journal of Geographical Information Science*, 24(3), 451–476. <https://doi.org/10.1080/13658810902904964>