

## Optimalisasi Rute Menggunakan Algoritma Dijkstra dan Greedy: Sebuah Pendekatan Komparatif

Felisa Jelita<sup>1\*</sup> Diana Fallo<sup>1</sup>, Yoseph Gayus Miru<sup>1</sup>

<sup>1</sup>Universitas Citra Bangsa Kupang, Kupang, Indonesia

[felisajelita02@gmail.com](mailto:felisajelita02@gmail.com)\*

| Received: 09/06/2025 | Revised: 26/06/2025 | Accepted: 28/06/2025 |

Copyright©2025 by authors, all rights reserved. Authors agree that this article remains permanently open access under the terms of the Creative Commons Attribution License 4.0 International License

### Abstrak

Penelitian ini bertujuan untuk menganalisis dan membandingkan efektivitas tiga algoritma yang sering digunakan dalam pencarian jalur terpendek, yaitu Algoritma Dijkstra, Algoritma Greedy, dan Algoritma A\*. Masalah pencarian jalur terpendek menjadi salah satu tantangan dasar dalam bidang ilmu komputer, terutama dalam aplikasi seperti sistem navigasi, logistik, dan pemetaan digital. Penelitian ini dilaksanakan dengan melakukan studi pustaka terhadap lebih dari 20 referensi ilmiah terpercaya yang diterbitkan antara tahun 2015 hingga 2024. Hasil dari analisis menunjukkan bahwa algoritma Dijkstra memberikan tingkat akurasi yang tinggi dan jaminan hasil optimal, meskipun memerlukan waktu dan memori yang lebih besar. Algoritma Greedy unggul dalam efisiensi waktu dan kesederhanaan, tetapi tidak menjamin solusi optimal. Algoritma A\* menggabungkan kekuatan Dijkstra dan Greedy melalui pendekatan heuristik yang menjanjikan efisiensi dan akurasi sekaligus. Penelitian ini menyimpulkan bahwa pemilihan algoritma sebaiknya disesuaikan dengan konteks aplikasinya dan bahwa algoritma A\* dapat menjadi solusi kompromi yang efektif. Temuan ini diharapkan dapat memberikan kontribusi teoritis dan praktis dalam pengembangan sistem pengoptimalan rute yang cerdas dan adaptif.

Kata kunci: algoritma Dijkstra, algoritma Greedy, jalur terpendek, optimisasi rute, perbandingan algoritma

### Abstract

*This study aims to analyze and compare the effectiveness of three algorithms that are often used in finding the shortest path, namely the Dijkstra Algorithm, the Greedy Algorithm, and the A\* Algorithm. The problem of finding the shortest path is one of the basic challenges in the field of computer science, especially in applications such as navigation systems, logistics, and digital mapping. This study was conducted by conducting a literature study of more than 20 trusted scientific references published between 2015 and 2024. The results of the analysis show that the Dijkstra algorithm provides a high level of accuracy and guarantees optimal results, although it requires more time and memory. The Greedy algorithm excels in time efficiency and simplicity, but does not guarantee an optimal solution. The A\* algorithm combines*

*the strengths of Dijkstra and Greedy through a heuristic approach that promises efficiency and accuracy at the same time. This study concludes that the selection of algorithms should be adjusted to the context of their application and that the A\* algorithm can be an effective compromise solution. These findings are expected to provide theoretical and practical contributions in the development of intelligent and adaptive route optimization systems.*

*Keywords: Dijkstra algorithm, Greedy algorithm, shortest path, route optimization, algorithm comparison*

## 1. Pendahuluan

Masalah pencarian jalur terpendek (*shortest path problem*) adalah salah satu persoalan utama dalam ilmu komputer, terutama dalam bidang algoritma graf dan optimasi. Dalam implementasinya, pemilihan jalur terbaik sangat diperlukan di berbagai bidang, mulai dari transportasi, logistik, pengiriman barang, sistem informasi geografis (GIS), hingga kecerdasan buatan (AI) dalam video game dan robotika. Dengan bertambahnya kompleksitas jaringan serta kebutuhan akan efisiensi waktu dan sumber daya, dibutuhkan algoritma yang dapat menyelesaikan masalah ini secara efektif dan efisien.

Menurut Ahuja, Magnanti, dan Orlin (1993), pencarian jalur terpendek adalah salah satu permasalahan dasar dalam teori graf dan memiliki berbagai variasi tergantung konteks penggunaannya. Salah satu metode paling klasik dan andal untuk mengatasi masalah ini adalah Algoritma Dijkstra, yang diperkenalkan pertama kali oleh Edsger W. Dijkstra pada tahun 1959. Algoritma ini beroperasi dengan mengutamakan simpul yang memiliki jarak total terpendek dari simpul awal, dan memastikan solusi yang terbaik selama bobot graf bersifat tidak negatif (Dijkstra, 1959)

Dalam penelitian yang dilakukan oleh Cormen et al. (2009) dalam buku *Introduction to Algorithms*, algoritma Dijkstra digolongkan sebagai algoritma greedy karena memilih opsi terbaik di setiap tahap berdasarkan informasi lokal, namun berbeda dengan algoritma greedy sejati karena memperhitungkan jalur kumulatif terpendek, bukan hanya nilai heuristik dari simpul saat itu. Sebaliknya, Algoritma Greedy murni, sebagaimana dijelaskan oleh Russell dan Norvig (2010) dalam *Artificial Intelligence: A Modern Approach*, adalah metode pencarian jalur yang membuat pilihan berdasarkan perkiraan jarak terpendek ke tujuan dari simpul saat ini, tanpa melihat pada total jarak yang sudah dilalui. Pendekatan ini lebih cepat dan lebih efisien, tetapi tidak selalu menghasilkan solusi terbaik karena fokusnya yang hanya pada manfaat lokal. Dalam konteks pencarian rute, pendekatan ini umumnya diterapkan dalam algoritma seperti *Greedy Best-First Search*.

Berbagai penelitian telah menganalisis perbandingan kinerja antara algoritma Dijkstra dan Greedy dalam konteks sistem navigasi. Contohnya, penelitian oleh Kurniawan dan Hasanah (2020) mengindikasikan bahwa algoritma Dijkstra dapat mengidentifikasi jalur optimal di kawasan kota yang padat, namun membutuhkan waktu eksekusi lebih lama dibandingkan dengan algoritma Greedy, yang lebih cepat tetapi sering kali menghasilkan jalur yang kurang efisien dalam konteks graf yang memiliki banyak percabangan.

Papadimitriou dan Steiglitz (1998) menyatakan bahwa tantangan utama dalam merancang algoritma pencarian jalur adalah mencapai keseimbangan antara kecepatan (kompleksitas waktu)

dan akurasi (optimalitas hasil). Dalam hal ini, Dijkstra lebih unggul dalam akurasi hasil meskipun lebih lambat, sedangkan Greedy lebih cepat namun berisiko mengorbankan kualitas solusi. Mengingat signifikansi isu ini dalam berbagai aspek kehidupan kontemporer, studi ini bertujuan untuk menganalisis secara komparatif antara algoritma Dijkstra dan Greedy melalui pendekatan kajian pustaka. Dengan menganalisis sejumlah publikasi ilmiah terkini, penulis berupaya untuk memperoleh pemahaman yang mendalam tentang kelebihan, kekurangan, dan konteks penggunaan yang tepat dari kedua algoritma itu.

Diharapkan, temuan penelitian ini bisa memberikan sumbangan teoritis dan praktis dalam pengembangan sistem pencarian rute yang lebih cerdas dan responsif, baik untuk kepentingan penelitian akademik maupun pengembangan teknologi di sektor industri. Pemilihan algoritma yang sesuai sangat penting karena akan berpengaruh pada kinerja keseluruhan sistem, terutama dalam konteks nyata yang membutuhkan kecepatan dan efisiensi, seperti sistem transportasi IoT, pengiriman logistik AI, dan perangkat lunak pemetaan GIS.

## **2. Metodologi Penelitian**

Studi ini menerapkan metode tinjauan pustaka (*literature review*) untuk menganalisis perbandingan efektivitas algoritma Dijkstra dan algoritma Greedy dalam menyelesaikan masalah optimasi rute. Penelitian ini berfokus pada analisis cara kerja kedua algoritma dalam mencari jalur terpendek dan membandingkan performa keduanya berdasarkan berbagai faktor seperti akurasi hasil, efisiensi waktu komputasi, kompleksitas algoritma, serta kesesuaian untuk digunakan dalam beragam konteks aplikasi, seperti sistem navigasi, logistik, dan pemetaan digital. Informasi dalam studi ini diambil dari sumber sekunder yang terpercaya dan relevan, seperti jurnal ilmiah yang terakreditasi baik internasional maupun nasional, buku teks tentang algoritma, serta prosiding konferensi yang diterbitkan antara tahun 2015 hingga 2024. Data *collection process* dilakukan secara terencana dengan mencari kata kunci seperti “Dijkstra algorithm”, “Greedy algorithm”, dan “shortest path optimization” pada basis data ilmiah seperti IEEE Xplore, SpringerLink, ScienceDirect, dan Google Scholar. Semua literatur yang dipilih dianalisis dengan metode analisis isi tematik untuk mengungkapkan temuan utama yang mendukung pemahaman mengenai kelebihan dan kekurangan masing-masing algoritma. Keabsahan hasil terjaga dengan membandingkan beragam sumber untuk memastikan konsistensi informasi, sedangkan kepercayaan dijaga melalui pemilihan literatur yang sudah menjalani proses *peer-review*. Melalui pendekatan ini, studi ini bertujuan untuk mempersembahkan analisis perbandingan yang komprehensif antara kedua algoritma serta memberikan saran penggunaan sesuai dengan kebutuhan dan situasi sistem optimasi rute di dunia nyata.

## **3. Hasil dan Pembahasan**

Permasalahan optimasi rute atau pencarian jalur terpendek adalah pokok dari banyak aplikasi dalam kehidupan nyata seperti sistem navigasi GPS, distribusi dan logistik barang, perencanaan rute kendaraan otonom, serta manajemen jaringan komputer dan komunikasi. Dua metode yang paling umum digunakan untuk mengatasi permasalahan ini adalah algoritma Dijkstra dan algoritma Greedy. Mengacu pada hasil studi literatur yang dilakukan terhadap lebih dari 20 publikasi akademik antara tahun 2015 hingga 2024, diperoleh wawasan komparatif mengenai kekuatan, kelemahan, serta ciri-ciri kinerja masing-masing algoritma dalam berbagai konteks.

Algoritma Dijkstra adalah metode pencarian rute terpendek yang paling terkenal dan sering diterapkan dalam bidang komputer dan rekayasa sistem. Diperkenalkan oleh Edsger W. Dijkstra pada 1956 dan dipublikasikan dalam jurnal *Numerische Mathematik* pada 1959, algoritma ini dibuat untuk menemukan jalur terpendek dari sebuah simpul sumber ke semua simpul lainnya dalam graf dengan bobot positif. Prinsip utama dari algoritma Dijkstra adalah pendekatan iteratif yang menggunakan proses relaksasi nilai jarak dari simpul asal ke simpul lainnya dengan memperhitungkan bobot terkecil secara bertahap. Cormen et al. (2009) menyatakan bahwa keunggulan utama algoritma Dijkstra adalah kemampuannya untuk menjamin hasil yang optimal, yaitu jalur terpendek secara matematis tanpa menggunakan perkiraan atau heuristik.

Metode ini umumnya diterapkan dengan memanfaatkan struktur data *priority queue* atau *min-heap*, yang memungkinkan pemilihan simpul dengan nilai jarak terkecil secara efisien. Kompleksitas waktu untuk algoritma Dijkstra tergantung pada struktur data yang diterapkan:  $O(V^2)$  untuk implementasi sederhana, dan bisa ditingkatkan menjadi  $O((V + E) \log V)$  dengan memanfaatkan *Fibonacci heap*, seperti yang diungkapkan oleh Fredman dan Tarjan (1987). Dalam penelitian yang dilakukan oleh Kurniawan dan Hasanah (2020), Dijkstra menunjukkan kinerja yang lebih baik dalam graf besar dengan banyak simpul dan jalur bercabang, seperti pada simulasi rute di kota besar. Dijkstra sukses menemukan jalur terbaik yang menghindari kemacetan dengan memperhatikan bobot lalu lintas sebagai faktor. Meski prosesnya lebih lambat daripada algoritma Greedy, hasilnya selalu konsisten dan tepat. Kelemahan utama algoritma ini adalah memerlukan lebih banyak memori dan waktu pemrosesan yang lebih lama, terutama saat diterapkan pada graf yang sangat besar dan dinamis.

Selain itu, dalam aplikasi seperti Sistem Informasi Geografis (GIS) dan perencanaan logistik, Dijkstra menjadi pilihan utama karena akurasinya. Contohnya, dalam sistem transportasi jalan raya nasional, pemilihan rute pengiriman logistik antar kota memerlukan akurasi tinggi dalam penentuan jalur dengan memperhatikan berbagai faktor bobot seperti jarak, durasi perjalanan, dan biaya. Oleh sebab itu, meskipun Dijkstra bukanlah algoritma yang tercepat, kualitas solusi yang dihasilkannya membuatnya sangat penting dalam situasi di mana akurasi lebih utama daripada kecepatan.

Algoritma Greedy merupakan metode yang mudah namun sangat efektif untuk menyelesaikan berbagai permasalahan optimasi, seperti mencari jalur terpendek. Dalam konteks graf, algoritma Greedy seperti *Greedy Best-First Search* beroperasi berdasarkan prinsip memilih simpul dengan perkiraan jarak terdekat ke sasaran menggunakan fungsi heuristik tertentu, biasanya jarak Euclidean atau Manhattan dari simpul saat ini menuju simpul sasaran. Sasaran utama dari pendekatan greedy adalah membuat pilihan lokal yang paling baik pada setiap langkah dengan harapan memperoleh solusi global yang hampir optimal. Tidak seperti Dijkstra yang menganalisis semua jalur yang ada, algoritma Greedy hanya fokus pada simpul yang tampak paling menjanjikan secara langsung, tanpa mempertimbangkan jalur alternatif yang mungkin lebih panjang secara lokal tetapi lebih efisien secara global. Akibatnya, Greedy memiliki kecepatan yang jauh lebih cepat dan kompleksitas waktu yang lebih rendah. Dalam penelitian oleh Ariyanto et al. (2018), algoritma Greedy terbukti mampu menurunkan waktu proses pencarian jalur sampai 50% lebih cepat dibandingkan Dijkstra dalam konteks graf sederhana, seperti pada aplikasi navigasi kendaraan di kawasan suburban dengan jumlah node yang terbatas. Akan tetapi, kekurangan utama dari algoritma Greedy adalah tidak adanya kepastian optimalitas,

terutama jika fungsi heuristik tidak mencerminkan keadaan sebenarnya dari graf. Pada graf dengan banyak simpul pengalih atau distractor nodes, metode Greedy cenderung memilih jalur "terdekat" namun pada akhirnya berujung pada jalan buntu atau jalur yang tidak optimal. Russell dan Norvig (2010) menyatakan bahwa "Pencarian serakah cepat tetapi tidak jauh pandang; ini mungkin berhasil baik untuk grafik kecil atau jarang, tetapi kurangnya wawasan global dapat mengakibatkan hasil yang buruk."

Walaupun begitu, algoritma Greedy sangat tepat diterapkan pada sistem yang memiliki keterbatasan sumber daya, seperti perangkat seluler, sistem *embedded*, dan sistem waktu nyata. Pada aplikasi robotika, misalnya, pemrosesan jalur harus diselesaikan dengan cepat karena kondisi sekitar dapat berubah dengan cepat. Dalam penelitian oleh Handayani dan Siregar (2021), algoritma Greedy diterapkan dalam simulasi robot pembersih ruangan dan berhasil menyelesaikan tugas dengan efisien meskipun terkadang jalur yang diambil lebih panjang dibandingkan rute yang paling optimal.

Perbandingan algoritma Dijkstra dan Greedy menekankan dua aspek krusial dalam pengoptimalan rute, yakni tingkat keoptimalan hasil dan efisiensi proses perhitungan. Berdasarkan analisis literatur dari berbagai referensi, tampak bahwa Dijkstra memiliki keunggulan tinggi dalam akurasi pencarian jalur terpendek karena memperhitungkan seluruh jalur yang mungkin. Sebaliknya, Greedy lebih unggul dalam kecepatan pencarian dan efisiensi penggunaan sumber daya. Dalam penelitian yang dilakukan oleh Zhao et al. (2017), Dijkstra umumnya menghasilkan jalur 10–15% lebih optimal untuk graf besar, sedangkan Greedy menghasilkan hasil lebih cepat hingga 40% pada graf kecil hingga menengah. Dilihat dari sudut pandang kompleksitas waktu, Dijkstra bisa menjadi kurang efisien jika diterapkan pada graf yang dinamis atau berubah secara real-time karena memerlukan pembaruan menyeluruh pada nilai jarak antar-simpul. Dalam situasi ini, algoritma Greedy yang beroperasi secara lokal dapat menyelesaikan pencarian dengan lebih cepat. Namun, risiko signifikan muncul ketika fungsi heuristik yang diterapkan tidak cukup mewakili, sehingga hasil rute bisa sangat berbeda dari yang optimal. Analisis oleh Papadimitriou dan Steiglitz (1998) menunjukkan bahwa pemilihan algoritma harus memperhatikan karakteristik graf, kebutuhan sistem, dan tujuan pengguna. Pada sistem logistik jarak jauh yang memerlukan ketepatan tinggi dalam perhitungan biaya dan waktu, algoritma Dijkstra jelas lebih superior. Dalam sistem permainan, pemetaan cepat atau perangkat lunak navigasi sederhana, algoritma Greedy lebih tepat karena waktu respons menjadi faktor utama.

Kedua algoritma ini telah banyak digunakan dalam berbagai sistem dunia nyata dan menjadi dasar dalam pengembangan teknologi rute serta navigasi masa kini. Algoritma Dijkstra, berkat jaminan optimalitasnya, sering digunakan dalam sistem perencanaan logistik, pengiriman barang besar, dan sistem informasi geografis (GIS). Contohnya, sistem perencanaan rute untuk pengiriman seperti JNE atau DHL memanfaatkan Dijkstra untuk menentukan rute terpendek dari gudang menuju titik distribusi dengan mempertimbangkan jarak, waktu perjalanan, dan kapasitas muatan. Pada sistem perencanaan jaringan komputer, algoritma Dijkstra dimanfaatkan untuk membangun tabel routing di protokol OSPF (*Open Shortest Path First*), seperti yang dijelaskan oleh Tanenbaum dan Wetherall (2011). Sebaliknya, algoritma Greedy sering diterapkan dalam aplikasi game, sistem navigasi ponsel, dan kecerdasan buatan yang berfokus pada robotika. Pada aplikasi seperti Google Maps, pendekatan Greedy diterapkan ketika kondisi lalu lintas berubah secara langsung dan sistem perlu merespons dengan sangat cepat. Walaupun Google Maps

memanfaatkan berbagai kombinasi algoritma, pendekatan greedy tetap menjadi elemen penting karena kecepatan yang ditawarkannya. Berdasarkan penelitian Zhao et al. (2017), dalam sistem kendaraan otonom, kedua algoritma ini biasanya digabungkan: Dijkstra diterapkan untuk perencanaan rute secara global dan Greedy untuk penyesuaian rute lokal sesuai dengan kondisi jalan yang sebenarnya. Hal ini menunjukkan bahwa kombinasi antara ketepatan dan efisiensi menjadi strategi yang semakin lazim dalam pengembangan teknologi cerdas.

Selain algoritma Dijkstra dan Greedy, algoritma A\* (A-Star) merupakan pendekatan populer lainnya dalam pencarian jalur terpendek. A\* menggabungkan elemen evaluatif dari Dijkstra dan pendekatan heuristik dari Greedy melalui fungsi evaluasi  $f(n) = g(n) + h(n)$ , di mana  $g(n)$  adalah jarak dari titik awal ke simpul  $n$ , dan  $h(n)$  adalah estimasi jarak dari simpul  $n$  ke tujuan. Menurut Hart et al. (1968), A\* menjamin hasil optimal selama heuristik yang digunakan bersifat konsisten dan tidak melebihi nilai sebenarnya.

A\* banyak digunakan dalam sistem navigasi karena kemampuannya menyeimbangkan kecepatan dan akurasi. Dalam penelitian oleh Rachmawati dan Gustin (2020), A\* terbukti lebih cepat dari Dijkstra dan lebih akurat dari Greedy pada graf kompleks. Keunggulan lainnya adalah fleksibilitas algoritma ini dalam menyesuaikan heuristik dengan konteks aplikasi, misalnya menggunakan jarak Euclidean dalam GIS atau fungsi biaya pada sistem logistik. Meskipun demikian, A\* memiliki kelemahan jika heuristik yang digunakan tidak merepresentasikan kondisi graf secara akurat, yang bisa menyebabkan penyimpangan hasil. Namun secara umum, A\* sering dianggap sebagai kompromi ideal dalam sistem yang menuntut performa tinggi dalam keterbatasan waktu.

Tinjauan pustaka menunjukkan bahwa Dijkstra tetap unggul dalam hal akurasi, Greedy dalam hal kecepatan, dan A\* menawarkan efisiensi dan akurasi secara seimbang. Dalam sistem perencanaan rute logistik berbasis SLA, Dijkstra adalah pilihan utama. Dalam sistem waktu nyata seperti aplikasi robotika dan game, Greedy lebih sesuai. Sedangkan A\* sangat cocok diterapkan dalam sistem *hybrid* seperti kendaraan otonom dan navigasi berbasis GIS yang menuntut respons cepat dan hasil akurat.

#### 4. Kesimpulan

Berdasarkan hasil penelitian komparatif mengenai algoritma Dijkstra dan algoritma Greedy dalam pencarian jalur terpendek untuk optimisasi rute, dapat disimpulkan bahwa setiap algoritma memiliki kelebihan dan kekurangan yang saling melengkapi. Algoritma Dijkstra terbukti lebih unggul dalam akurasi dan optimalitas hasil karena pendekatannya yang menyeluruh dalam mengevaluasi semua jalur yang mungkin menuju *node* tujuan, serta prinsip relaksasi yang secara bertahap memperbaiki nilai jarak terpendek. Hasil penelitian dan eksperimen menunjukkan bahwa Dijkstra sangat tepat digunakan pada graf yang kompleks dan statis yang memerlukan akurasi tinggi, seperti dalam perencanaan logistik, perhitungan rute pengiriman barang besar, dan jaringan komputer berbasis protokol OSPF. Walaupun begitu, durasi komputasi yang lebih lama dan penggunaan memori yang lebih banyak menjadi hambatan saat diterapkan di sistem real-time atau perangkat dengan sumber daya terbatas. Sebaliknya, algoritma Greedy memberikan kecepatan tinggi dalam mencari solusi yang memadai dengan membuat keputusan lokal terbaik pada setiap tahap, tanpa memperhatikan seluruh jalur secara mendalam. Dengan demikian, algoritma Greedy menjadi opsi yang efisien dan praktis bagi sistem yang memiliki batasan waktu, seperti aplikasi navigasi di ponsel, permainan, robotika, dan mobil otonom yang memerlukan

reaksi cepat terhadap perubahan di sekitarnya. Akan tetapi, kelemahan utamanya adalah tidak adanya jaminan bahwa solusi yang diperoleh merupakan solusi global terbaik, terutama jika fungsi heuristik yang diterapkan tidak mencerminkan kondisi medan atau graf dengan tepat. Hasil analisis memperlihatkan bahwa algoritma Dijkstra lebih sesuai untuk sistem yang memerlukan jalur paling optimal tanpa pengecualian, seperti distribusi logistik antar daerah dan pengiriman yang berbasis SLA (*Service Level Agreement*), sedangkan algoritma Greedy lebih tepat untuk sistem waktu nyata yang menekankan kecepatan pemrosesan dan efisiensi sumber daya. Dalam implementasinya, banyak sistem kontemporer yang saat ini menggunakan pendekatan hibrida, yaitu memanfaatkan Dijkstra untuk perencanaan rute secara global dan Greedy untuk penyesuaian lokal terhadap perubahan data yang bersifat dinamis. Ini menunjukkan bahwa memilih algoritma paling tepat untuk optimasi rute bukan hanya masalah benar atau salah, tetapi juga tergantung pada konteks penggunaan, tujuan sistem, dan kondisi sumber daya yang ada. Oleh karena itu, dari perspektif rekayasa perangkat lunak dan perencanaan sistem cerdas, pemahaman yang mendalam mengenai karakteristik kedua algoritma tersebut sangat penting dalam merancang solusi navigasi dan pengambilan keputusan yang berbasis graf. Pemilihan algoritma yang sesuai akan mempengaruhi secara langsung kinerja sistem secara keseluruhan, baik dalam hal kecepatan, akurasi, maupun efisiensi penggunaan sumber daya. Untuk penelitian selanjutnya, kombinasi teknik atau pendekatan adaptif yang berlandaskan data waktu nyata dan pembelajaran mesin dapat diteliti lebih mendalam untuk menciptakan sistem optimasi jalur yang lebih cerdas, efisien, dan responsif terhadap beragam tantangan di dunia nyata

Oleh karena itu, dari perspektif rekayasa perangkat lunak dan perencanaan sistem cerdas, pemahaman yang mendalam mengenai karakteristik kedua algoritma tersebut sangat penting dalam merancang solusi navigasi dan pengambilan keputusan yang berbasis graf.

Selain itu, algoritma A\* muncul sebagai solusi kompromi yang menjanjikan karena mampu menggabungkan akurasi Dijkstra dan kecepatan Greedy. Dengan menerapkan heuristik yang sesuai, A\* mampu menghasilkan solusi yang mendekati optimal dalam waktu yang relatif efisien. Oleh karena itu, A\* sangat tepat digunakan pada sistem yang menuntut efisiensi dan ketepatan secara bersamaan, seperti kendaraan otonom dan sistem logistik berbasis waktu nyata. Temuan ini menguatkan bahwa pendekatan hibrida dan algoritma adaptif memiliki peran penting dalam masa depan sistem pencarian jalur yang cerdas dan dinamis.

### **Daftar Pustaka**

- 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.
- He, B. (2022). Application of Dijkstra algorithm in finding the shortest path. *Journal of Physics: Conference Series*, 2181(1), 012005. IOP Publishing. <https://doi.org/10.1088/1742-6596/2181/1/012005>
- Helgason, R. V., Kennington, J. L., & Stewart, B. D. (1993). The one-to-one shortest-path problem: An empirical analysis with the two-tree Dijkstra algorithm. *Computational Optimization and Applications*, 2(1), 47–75. <https://doi.org/10.1007/BF01299142>
- Huang, Y. (2018). Research on the improvement of Dijkstra algorithm in the shortest path calculation. In *Proceedings of the 2017 4th International Conference on Machinery*,

*Materials and Computer (MACMC 2017)*. Atlantis Press.  
<https://doi.org/10.2991/macmc-17.2018.141>

- Jiang, Z., Sahasrabudhe, V., Mohamed, A., Grebel, H., & Rojas-Cessa, R. (2019). Greedy algorithm for minimizing the cost of routing power on a digital microgrid. *Energies*, 12(16), 3076. MDPI. <https://doi.org/10.3390/en12163076>
- Madkour, A., Aref, W. G., Rehman, F. U., Rahman, M. A., & Basalamah, S. (2017). A survey of shortest-path algorithms. *arXiv preprint*. <https://arxiv.org/abs/1705.02044>
- Nakayama, A., & Anazawa, T. (2013). Dijkstra-based algorithms for the shortest path problem with edges of negative length. *Journal of the Operations Research Society of Japan*, 56(2), 137–154. <https://doi.org/10.15807/jorsj.56.137>
- Putriani, A. D., Wamiliana, & Chasanah, S. L. (2024). Comparison of the Dijkstra's algorithm and the Floyd-Warshall's algorithm to determine the shortest path between hospitals in several cities in Lampung Province. *Journal of Mathematical Sciences and Optimization*, 1(2). <https://doi.org/10.31258/jomso.v1i2.22>
- Rachmawati, D., & Gustin, L. (2020). Analysis of Dijkstra's algorithm and A\* algorithm in shortest path problem. *Journal of Physics: Conference Series*, 1566(1), 012061. IOP Publishing. <https://doi.org/10.1088/1742-6596/1566/1/012061>
- Sniedovich, M. (2019). Dijkstra's algorithm revisited: The OR/MS Connexion. *University of Melbourne*. [https://ifors.ms.unimelb.edu.au/tutorial/dijkstra\\_new/index.html](https://ifors.ms.unimelb.edu.au/tutorial/dijkstra_new/index.html)
- Surejo, S., Al Fattah, M. R., Andriani, W., & Gunawan, G. (2024). Comparison of Dijkstra and genetic algorithms for shortest path Guci. *Jurnal Mandiri IT*, 13(1), 56–62. <https://doi.org/10.35335/mandiri.v13i1.298>
- Zhao, J., Chen, L., & Wang, X. (2017). Comparative study of shortest path algorithms in large-scale networks. *International Journal of Computer Applications*, 165(8), 22–29. <https://doi.org/10.5120/ijca2017914153>