



Department of Digital Business

Journal of Artificial Intelligence and Digital Business (RIGGS)

Homepage: <https://journal.ilmudata.co.id/index.php/RIGGS>

Vol. 4 No. 3 (2025) pp: 5017-5022

P-ISSN: 2963-9298, e-ISSN: 2963-914X

---

## Analisis Komparatif Algoritma Dijkstra dan Google Maps API dalam Penentuan Rute Tercepat

Raja Joko Musridho<sup>1\*</sup>, Hidayati Rusnedy<sup>2</sup>, Wahyu Febri Ramadhan Sudirman<sup>3</sup>

<sup>1,2</sup> Teknik Informatika, Fakultas Teknik, Universitas Pahlawan Tuanku Tambusai

<sup>3</sup> Perbankan Syari'ah, Fakultas Agama Islam, Universitas Pahlawan Tuanku Tambusai

rajajoko@gmail.com

### Abstrak

Penelitian ini membandingkan algoritma *shortest path* klasik, khususnya Dijkstra, dengan sistem penentuan rute berbasis waktu nyata seperti Google Maps dalam konteks efisiensi perjalanan. Simulasi manual dilakukan menggunakan graf berbobot statis, sementara studi kasus di Kota Pekanbaru digunakan untuk mengamati rekomendasi rute Google Maps pada dua waktu berbeda. Hasil simulasi menunjukkan bahwa algoritma Dijkstra menghasilkan rute optimal tetap (contoh:  $A \rightarrow C \rightarrow D \rightarrow E$  dengan total waktu 5 menit pada kondisi normal), namun tidak dapat beradaptasi ketika terjadi perubahan kondisi lalu lintas. Sebaliknya, hasil observasi Google Maps menunjukkan adanya variasi rute: pada malam hari rute tercepat ditempuh dalam  $\pm 19$  menit dengan jarak  $\pm 11,2$  km, sementara pada siang hari rute berbeda dipilih dengan waktu tempuh lebih lama  $\pm 26$  menit meskipun jaraknya hampir sama ( $\pm 10,9$  km). Perbedaan ini menegaskan bahwa Google Maps memperbarui bobot graf internalnya secara dinamis untuk menyesuaikan dengan kondisi lalu lintas. Temuan ini menyoroti keterbatasan algoritma konvensional di lingkungan perkotaan dinamis, sekaligus memperkuat keunggulan sistem navigasi berbasis data. Dengan demikian, penelitian ini menyimpulkan bahwa algoritma *shortest path* klasik relevan untuk tujuan edukatif dan simulatif, tetapi sistem navigasi berbasis data waktu nyata lebih efektif diterapkan dalam transportasi perkotaan yang kompleks.

Kata kunci: Algoritma Dijkstra, Rute Tercepat, Google Maps API, Navigasi Waktu Nyata, Lalu Lintas Dinamis

### 1. Latar Belakang

Kemajuan teknologi informasi telah mendorong perubahan besar dalam cara manusia berinteraksi dengan ruang geografis, terutama melalui sistem navigasi digital. Sistem navigasi berbasis peta digital telah menjadi elemen krusial dalam infrastruktur kota cerdas, mendukung efisiensi mobilitas harian, perencanaan transportasi, dan logistik urban (Stepniak et al., 2021). Salah satu tantangan utama dalam sistem ini adalah bagaimana menentukan rute tercepat antara dua titik, mengingat kondisi lalu lintas yang bersifat dinamis dan tidak dapat diprediksi secara statis.

Permasalahan penentuan rute optimal dikenal sebagai *shortest path problem*, yakni proses pencarian lintasan dengan bobot minimum dalam suatu graf berbobot. Dalam konteks sistem navigasi, simpul pada graf merepresentasikan lokasi, dan sisi mewakili jalur dengan bobot berupa jarak, waktu, atau biaya (Cormen et al., 2009). Algoritma klasik seperti Dijkstra banyak digunakan dalam sistem navigasi berbasis graf statis yang tidak mempertimbangkan dinamika waktu nyata seperti kemacetan lalu lintas (Jelita et al., 2025; Saputra & Nuryana, 2022).

Kelemahan utama dari algoritma *shortest path* konvensional adalah ketidakmampuannya mempertimbangkan variabel dinamis seperti kemacetan, penutupan jalan, atau cuaca. Sebaliknya, sistem navigasi modern mengintegrasikan data *crowdsourced* dan model prediktif yang memanfaatkan informasi waktu nyata maupun historis dari pengguna, kendaraan, dan sensor lingkungan, sehingga rute dapat diperbarui secara adaptif. Temuan terbaru menegaskan tren ini melalui pemodelan Dijkstra berbobot dinamis berbasis prediksi lalu lintas (Udhan et al., 2022), kerangka navigasi real-time berbasis *Large Language Models* (Zhou et al., 2024), implementasi praktis bersama Google Maps API pada studi rute perkotaan (Faridawaty et al., 2024), pengembangan algoritma optimisasi berbasis sensor dan *IoT* untuk kualitas rute yang lebih holistik (Rosca et al., 2025), serta solusi rute dinamis terintegrasi di ekosistem komputasi modern (Wang et al., 2024).

Berbagai penelitian mutakhir juga menunjukkan perkembangan signifikan dalam optimasi rute dinamis. Udhan et al. (2022) mengusulkan Algoritma Dijkstra berbobot dinamis yang memasukkan prediksi lalu lintas. Zhou et al. (2024) memperkenalkan DynamicRouteGPT, yaitu kerangka kerja navigasi multi-kendaraan waktu nyata berbasis *Large Language Models* (LLM). Pham et al. (2025) membandingkan algoritma klasik dengan pendekatan *Reinforcement Learning* (RL) dalam jaringan transportasi yang berubah-ubah. Selain itu, Liu & Meidani (2025) memanfaatkan *Graph Neural Networks* (GNN) untuk estimasi jarak dan rekomendasi rute pada kondisi probabilistik, sementara Wei & Ouyang (2025) mengembangkan *Decision Transformer* (DT) berbasis *Generalized Policy Gradient* (GPG) untuk perencanaan rute dalam jaringan transportasi tidak pasti. Pada sisi aplikasi praktis, Faridawaty et al. (2024) mengombinasikan algoritma *A\** (*A-Star*) dengan Google Maps *Application Programming Interface* (API) untuk optimasi rute transportasi umum, dan Google Developers (2024) merilis *Navigation Software Development Kit* (SDK) yang memungkinkan integrasi *Estimated Time of Arrival* (ETA) berbasis lalu lintas nyata ke dalam aplikasi *mobile*.

Meskipun terdapat banyak inovasi, masih sedikit studi yang secara eksplisit membandingkan algoritma *shortest path* klasik dengan sistem navigasi digital berbasis data lalu lintas waktu nyata, khususnya Google Maps (Alyrchikov et al., 2024; Syaifullah, 2025). Gap ini menunjukkan urgensi dilakukannya studi komparatif antara pendekatan algoritmik tradisional dan sistem navigasi modern berbasis API dinamis. Oleh karena itu, penelitian ini bertujuan untuk menjelaskan prinsip kerja algoritma *shortest path* klasik, khususnya Dijkstra, sekaligus menganalisis mekanisme Google Maps API dalam menentukan rute tercepat. Selain itu, penelitian ini juga berupaya membandingkan hasil simulasi algoritma Dijkstra dengan rekomendasi Google Maps pada kondisi nyata, sehingga dapat diketahui perbedaan efektivitas dan kemampuan adaptasi kedua pendekatan tersebut dalam konteks transportasi perkotaan.

## 2. Metode Penelitian

Penelitian ini menggunakan pendekatan deskriptif-komparatif, yang bertujuan untuk membandingkan efektivitas penentuan rute tercepat antara algoritma *shortest path* klasik (Dijkstra) dan sistem navigasi digital berbasis Google Maps API. Pendekatan ini dilakukan melalui dua metode utama, yaitu simulasi graf berbobot statis dan observasi manual terhadap rekomendasi rute pada Google Maps.

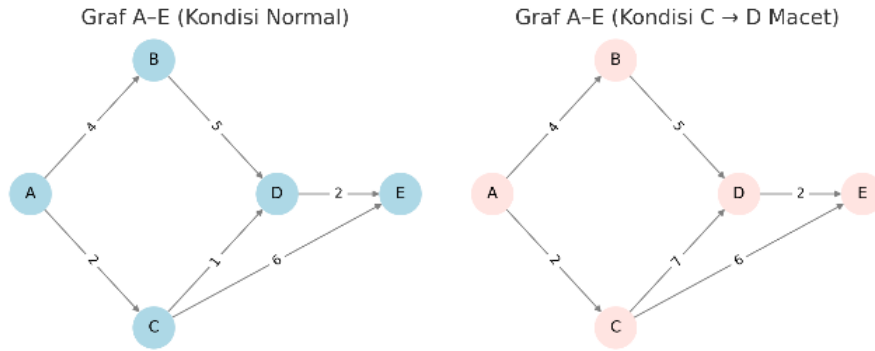
Simulasi dilakukan pada graf sederhana yang terdiri dari lima simpul (A–E) dan enam sisi berbobot. Setiap sisi memiliki bobot tetap yang merepresentasikan estimasi waktu tempuh antar titik. Penentuan rute tercepat dilakukan secara manual menggunakan algoritma Dijkstra. Simulasi ini digunakan untuk mewakili pendekatan statis yang tidak berubah terhadap kondisi eksternal seperti lalu lintas.

Observasi dilakukan secara langsung menggunakan aplikasi Google Maps pada perangkat mobile. Studi kasus dilakukan pada rute di dalam Kota Pekanbaru, yaitu dari Kampus II Universitas Pahlawan Tuanku Tambusai menuju SKA Mall. Waktu observasi dipilih pada dua titik berbeda, malam hari (23:48 WIB) dan siang hari (13:59 WIB) untuk menangkap dampak kondisi lalu lintas terhadap rekomendasi rute. Estimasi waktu tempuh dan rute yang disarankan dicatat dari tangkapan layar (screenshot) aplikasi.

Data hasil simulasi dan observasi dibandingkan berdasarkan tiga aspek utama, yaitu struktur rute yang dihasilkan, estimasi waktu tempuh, dan sensitivitas terhadap kondisi lalu lintas. Analisis ini digunakan untuk mengevaluasi perbedaan karakteristik pendekatan algoritma statis dan sistem navigasi dinamis berbasis data waktu nyata.

## 3. Hasil dan Diskusi

Simulasi dilakukan pada graf sederhana yang terdiri dari lima simpul (A–E) dan enam sisi berbobot, mewakili estimasi waktu tempuh antar titik dalam satuan menit. Tujuannya adalah mengilustrasikan cara kerja algoritma Dijkstra dalam kondisi statis. Simulasi dapat dilihat pada Gambar 1 berikut.



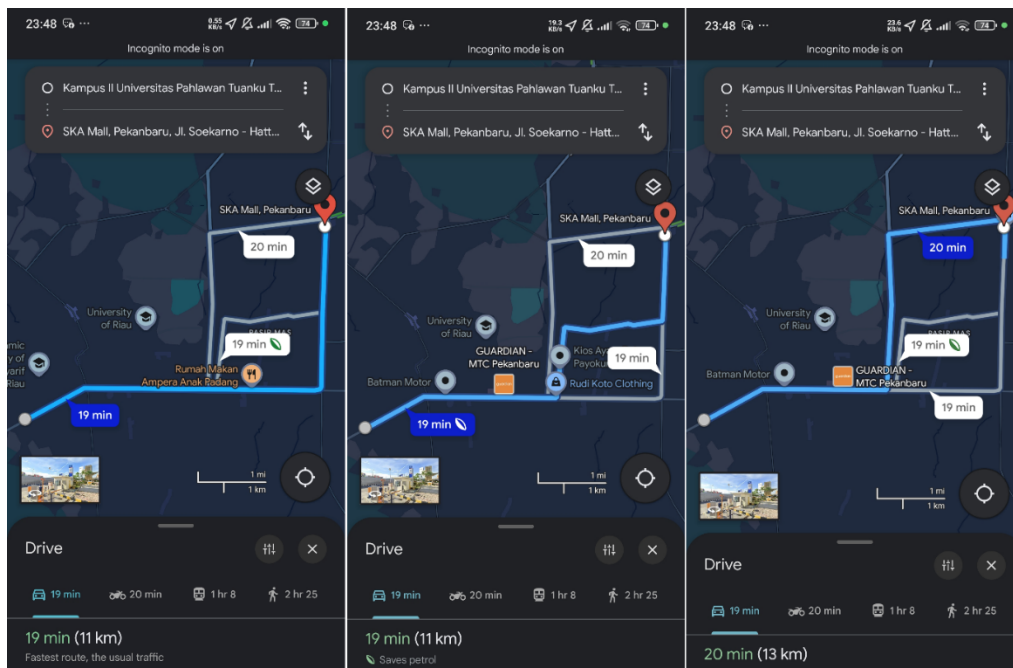
Gambar 1. Simulasi Graf A-E dengan Kondisi Normal dan Macet pada Titik C-D

Graf kiri menggambarkan rute tercepat dari titik A ke E dalam kondisi normal, di mana jalur C → D memiliki bobot 1. Dalam skenario ini, algoritma Dijkstra menghasilkan rute optimal A → C → D → E dengan total waktu tempuh 5 menit. Sebaliknya, graf kanan menunjukkan skenario simulasi kemacetan, di mana bobot pada ruas C → D ditingkatkan menjadi 7. Perubahan bobot ini menyebabkan algoritma Dijkstra memilih rute alternatif A → B → D → E dengan total waktu tempuh 11 menit.

Simulasi dilakukan pada dua graf statis yang terpisah, graf kiri mewakili kondisi normal dan graf kanan merepresentasikan kondisi lalu lintas yang mengalami kemacetan. Kedua graf dianalisis secara independen menggunakan algoritma Dijkstra, tanpa adanya pembaruan bobot selama proses pencarian lintasan berlangsung. Dengan demikian, pendekatan ini tetap berada dalam ruang lingkup graf statis. Perubahan rute terjadi bukan karena dinamika sistem atau pembaruan waktu nyata, melainkan semata-mata akibat perbedaan kondisi awal dari struktur graf. Hal ini menegaskan bahwa algoritma Dijkstra selalu mengikuti bobot yang telah ditetapkan sejak awal dan tidak mampu beradaptasi terhadap perubahan eksternal seperti kemacetan lalu lintas.

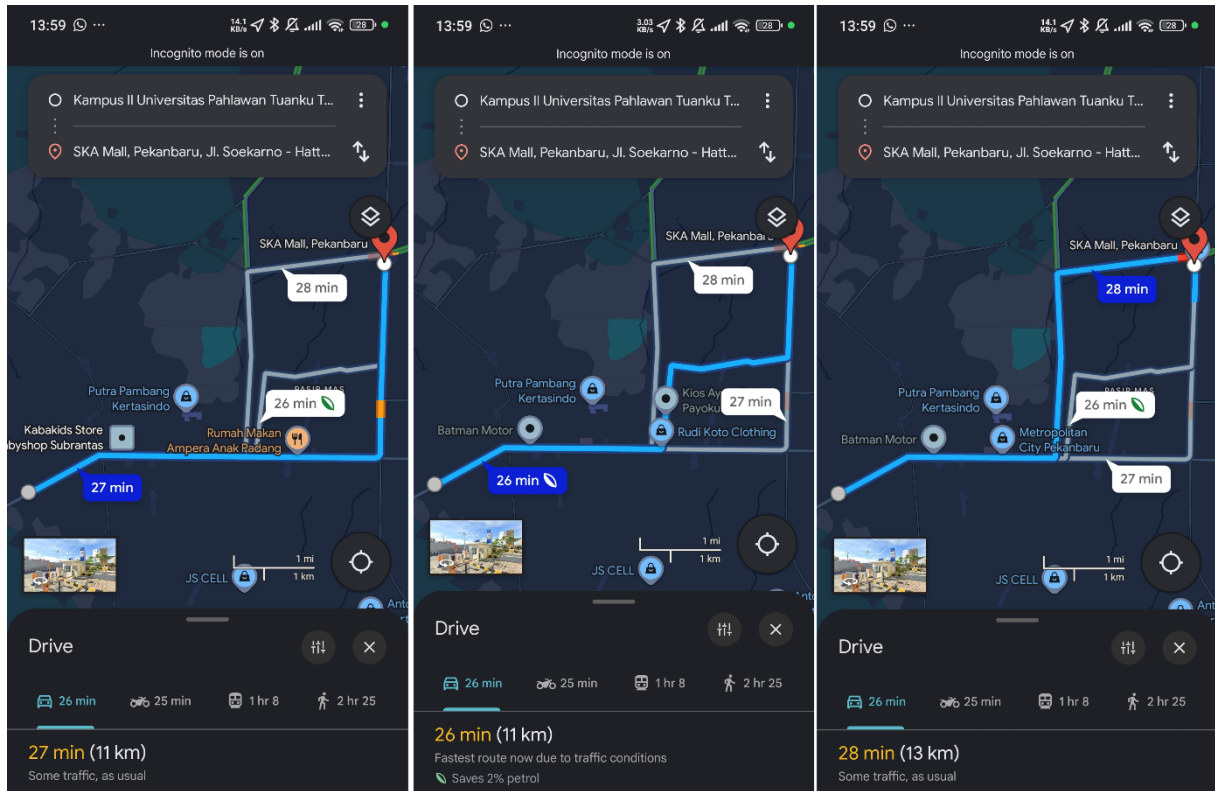
Observasi dilakukan menggunakan layanan Google Maps pada dua waktu berbeda dengan titik awal dan tujuan yang sama, yakni dari Kampus II Universitas Pahlawan Tuanku Tambusai (F953+3WR) menuju SKA Mall Pekanbaru (FCX9+HJ). Waktu observasi dilakukan pada pukul 23:48 WIB (Waktu Indonesia Barat) dan 13:59 WIB, untuk menganalisis dinamika rekomendasi jalur tercepat berdasarkan kondisi lalu lintas.

Pada pukul 23:48 WIB, Google Maps menyarankan jalur tercepat melalui Jalan H.R. Soebrantas, dilanjutkan ke arah Simpang Pasar Pagi Arengka, kemudian belok kiri ke Jalan Soekarno-Hatta menuju SKA Mall. Jalur ini ditampilkan dengan estimasi waktu tempuh tercepat yakni 19 menit, sebagaimana ditunjukkan pada Gambar 2.



Gambar 2. Screenshot Google Maps Rute Kampus II UP → SKA Mall – 23:48 WIB

Sementara itu, dapat dilihat pada Gambar 3 pada pukul 13:59 WIB, saat volume lalu lintas meningkat, Google Maps menyarankan rute berbeda untuk menghindari kepadatan lalu lintas. Jalur tercepat saat itu melalui Jalan H.R. Soebrantas, berbelok ke Jalan Rajawali Sakti, kemudian melalui Jalan Melati, Jalan Lobak, dan akhirnya masuk ke Jalan Soekarno-Hatta untuk mencapai SKA Mall. Rute alternatif yang ditampilkan namun tidak pernah dipilih sebagai rute tercepat adalah rute terpanjang, yaitu sejauh 13 km melalui Jalan H.R. Soebrantas kemudian berbelok ke Jalan S.M. Amin, dilanjutkan ke Jalan Tuanku Tambusai, lalu Jalan Soekarno-Hatta dan memutar balik untuk menuju mall tersebut.



Gambar 3. Screenshot Google Maps Rute Kampus II UP → SKA Mall – 13:59 WIB Rangkuman dari hasil observasi dapat dilihat pada tabel 1 berikut.

Ringkasan hasil observasi ditampilkan pada Tabel 1. Dari tabel tersebut terlihat bahwa pada malam hari (23:48 WIB) Google Maps menyarankan rute langsung melalui Jalan H.R. Soebrantas dengan estimasi waktu 19 menit karena lalu lintas relatif lancar. Sebaliknya, pada siang hari (13:59 WIB), rute yang disarankan lebih memutar melalui Jalan Rajawali Sakti, Jalan Melati, dan Jalan Lobak dengan waktu tempuh 26 menit. Faktor utama perbedaan ini adalah kondisi lalu lintas yang padat pada rute utama di siang hari, sehingga sistem secara dinamis memilih rute alternatif meskipun jaraknya hampir sama.

Tabel 1. Hasil Observasi Rute Google Maps Pekanbaru (Kampus II UP → SKA Mall)

Waktu Observasi	Rute Tercepat yang Disarankan	Estimasi Waktu	Estimasi Jarak	Catatan
23:48 WIB	H.R. Soebrantas → Simpang Pasar Pagi Arengka → Soekarno-Hatta → SKA	19 menit	± 11,2 km	Lalu lintas lancar, rute langsung
13:59 WIB	H.R. Soebrantas → Rajawali Sakti → Melati → Lobak → Soekarno-Hatta → SKA	26 menit	± 10,9 km	Rute memutar, dipilih karena kondisi padat di rute langsung
23:48 WIB dan 13:59 WIB	H.R. Soebrantas → S.M. Amin → Tuanku Tambusai → Soekarno-Hatta (putar balik) → SKA	Tidak dipilih	± 13,0 km	Rute terpanjang, tidak pernah disarankan

Hasil perbandingan antara simulasi algoritma Dijkstra dan observasi nyata melalui Google Maps menunjukkan perbedaan pendekatan yang mendasar. Algoritma Dijkstra bekerja pada graf berbobot statis, di mana setiap nilai bobot ditentukan sejak awal dan tidak berubah selama proses pencarian lintasan. Rute yang dihasilkan bersifat

tetap dan hanya bergantung pada struktur graf awal, tanpa mempertimbangkan faktor eksternal seperti kondisi lalu lintas atau waktu aktual.

Sebaliknya, sistem navigasi Google Maps bersifat adaptif. Rekomendasi rute yang diberikan dapat berubah secara otomatis mengikuti dinamika lalu lintas waktu nyata, termasuk kemacetan, kepadatan kendaraan, serta estimasi durasi perjalanan. Dalam observasi yang dilakukan, perbedaan rute antara waktu malam dan siang menunjukkan bahwa sistem Google Maps memperbarui bobot sisi pada graf internalnya secara dinamis. Bahkan ketika jalur alternatif lebih jauh secara jarak, sistem tetap memilihnya jika estimasi waktunya lebih singkat.

Temuan ini menegaskan keunggulan pendekatan navigasi berbasis data waktu nyata dalam konteks dunia nyata yang dinamis. Meskipun prinsip dasarnya tetap bertumpu pada struktur graf dan pencarian lintasan terpendek, sistem Google Maps merepresentasikan evolusi konsep shortest path menuju aplikasi yang ditopang oleh big data dan pemrosesan waktu nyata. Sementara algoritma Dijkstra tetap relevan dalam skenario statis, edukatif, atau sistem tertutup, sistem seperti Google Maps lebih sesuai untuk kebutuhan navigasi kompleks dan lingkungan transportasi perkotaan yang berubah-ubah.

Ringkasan perbedaan antara kedua pendekatan tersebut dapat dilihat pada Tabel 2 berikut.

Tabel 2. Perbandingan Algoritma Dijkstra dan Google Maps

Aspek	Algoritma Dijkstra	Google Maps API
Tipe Sistem	Statis	Dinamis berbasis data waktu nyata
Sumber Data	Bobot tetap pada graf	Data waktu nyata dan prediktif
Respons terhadap Kemacetan	Tidak adaptif	Adaptif terhadap kondisi lalu lintas
Metode Penentuan Rute	Berdasarkan bobot terkecil	Berdasarkan estimasi waktu tercepat
Fleksibilitas Jalur	Jalur tetap	Jalur dapat berubah setiap saat
Konteks Penggunaan	Simulasi algoritmik, game, warehouse	Navigasi jalan raya dan transportasi umum
Kelebihan	Stabil, deterministik, sederhana	Adaptif, akurat di dunia nyata

#### 4. Kesimpulan

Penelitian ini bertujuan untuk menjelaskan prinsip kerja algoritma shortest path klasik, khususnya Dijkstra, sekaligus menganalisis mekanisme Google Maps API dalam menentukan rute tercepat, serta membandingkan hasil keduanya pada kondisi nyata. Hasil penelitian menunjukkan bahwa algoritma Dijkstra bekerja secara statis dengan menghasilkan rute tetap berdasarkan bobot awal, sedangkan Google Maps API bersifat adaptif dengan menyesuaikan rute sesuai kondisi lalu lintas waktu nyata. Dengan demikian, seluruh tujuan penelitian ini telah tercapai. Temuan penelitian menegaskan adanya perbedaan mendasar antara pendekatan algoritmik tradisional dan sistem navigasi modern. Algoritma Dijkstra hanya relevan pada lingkungan dengan bobot tetap atau skenario edukatif, karena tidak dapat beradaptasi terhadap perubahan eksternal. Sebaliknya, Google Maps mampu memperbarui bobot graf internalnya secara otomatis, sehingga dapat menyarankan jalur tercepat meskipun jalur tersebut lebih panjang secara geometris. Hal ini membuktikan keunggulan pendekatan berbasis data waktu nyata dalam konteks transportasi perkotaan yang dinamis. Implikasi dari penelitian ini adalah bahwa sistem navigasi cerdas yang berbasis big data dan pemrosesan waktu nyata lebih efektif dibandingkan algoritma shortest path klasik untuk aplikasi dunia nyata. Kontribusi penelitian ini terletak pada penguatan pemahaman mengenai keterbatasan algoritma konvensional sekaligus memperlihatkan bagaimana integrasi data crowdsourced, sensor, dan model prediktif dapat meningkatkan akurasi penentuan rute tercepat. Dengan demikian, penelitian ini relevan tidak hanya untuk bidang ilmu komputer, tetapi juga untuk pengembangan transportasi cerdas di perkotaan. Meski demikian, penelitian ini masih memiliki keterbatasan pada lingkup kasus yang relatif sempit, karena hanya menguji satu rute pada dua kondisi waktu di Kota Pekanbaru. Untuk penelitian selanjutnya, disarankan memperluas cakupan observasi pada berbagai lokasi, waktu, dan kondisi lalu lintas yang lebih beragam, serta melakukan perbandingan dengan platform navigasi lain. Hal ini diharapkan dapat memperkuat generalisasi temuan dan memberikan kontribusi yang lebih luas bagi pengembangan sistem navigasi berbasis big data.

#### Referensi

Alyrchikov, I., Moiseev, N., & Knyaz, V. A. (2024). An algorithm for operational navigation in urban development using reinforcement learning. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 48, 1–7. <https://doi.org/https://doi.org/10.5194/isprs-archives-XLVIII-2-W5-2024-1-2024>

- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms (3-rd edition). In *MIT Press and McGraw-Hill*. The MIT Press. <https://mitpress.mit.edu/9780262033848/introduction-to-algorithms/>
- Faridawaty, Arnita, & Dewi, S. (2024). Using A\* Algorithm and Google Maps API for Web-Based Path Optimisation Public Vehicles Routes in Medan City. *Jurnal Penelitian Pendidikan IPA*, 10(SpecialIssue), 332–336.
- Google Developers. (2024). *Directions API (Legacy) overview*. <https://developers.google.com/maps/documentation/directions/overview>
- 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/https://doi.org/10.53863/kst.v7i01.1659>
- Liu, T., & Meidani, H. (2025). Graph neural networks for travel distance estimation and route recommendation under probabilistic hazards. *International Journal of Transportation Science and Technology*.
- Pham, H. D., Narasimhamurthy, S. M., Mehran, B., Manley, E., & Ashraf, A. (2025). Reinforcement learning based estimation of shortest paths in dynamically changing transportation networks. *Frontiers in Future Transportation*, 6, 1524232.
- Rosca, C.-M., Stancu, A., & Gortoescu, I.-A. (2025). Advanced Sensor Integration and AI Architectures for Next-Generation Traffic Navigation. *Applied Sciences (2076-3417)*, 15(8).
- Saputra, M. A., & Nuryana, I. K. D. (2022). SIG Penentuan Rute Terdekat Menuju Faskes di Sidoarjo Menggunakan Dynamic Dijkstra. *Journal of Informatics and Computer Science (JINACS)*, 4(01), 45–55. <https://doi.org/https://doi.org/10.26740/jinacs.v4n01.p45-55>
- Stepniak, C., Jelonek, D., Wyrwicka, M., & Chomiak-Orsa, I. (2021). Integration of the infrastructure of systems used in smart cities for the planning of transport and communication systems in cities. *Energies*, 14(11), 3069. <https://doi.org/https://doi.org/10.3390/en14113069>
- Syaifulallah, M. B. (2025). *Implementasi Algoritma Dijkstra Untuk Menentukan Rute Terpendek Dalam Pendistribusian Produk*. <https://repository.its.ac.id/118237/>
- Udhan, P., Ganeshkar, A., Murugesan, P., Permani, A. R., Sanjeeva, S., & Deshpande, P. (2022). Vehicle Route Planning using Dynamically Weighted Dijkstra's Algorithm with Traffic Prediction. *ArXiv Preprint ArXiv:2205.15190*.
- Wang, B., Tian, Y., Chen, F., Wang, J., & Zhang, Y. (2024). Dynamic Route Planning System Integrated with Traffic Flow Sensing. *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*, 383–387.
- Wei, X., & Ouyang, Y. (2025). GPG-HT: Generalized Policy Gradient with History-Aware Decision Transformer for Probabilistic Path Planning. *ArXiv Preprint ArXiv:2508.17218*.
- Zhou, Z., Zhou, B., & Liu, H. (2024). Dynamicrouteqpt: A real-time multi-vehicle dynamic navigation framework based on large language models. *ArXiv Preprint ArXiv:2408.14185*.