

# Pemodelan Graf Berarah Berbobot untuk Optimasi Penentuan Rute Terpendek Antar Kampus Polimarin Berbasis Algoritma *Dijkstra*

Helena Devi Ariyani dkk.



Volume 14, Issue 1, Pages 151–162, April 2026

Diterima 24 Januari 2026, Direvisi 13 April 2026, Disetujui 18 April 2026, Diterbitkan 23 April 2026

To Cite this Article : H. D. Ariyani dkk., "Pemodelan Graf Berarah Berbobot untuk Optimasi Penentuan Rute Terpendek Antar Kampus Polimarin Berbasis Algoritma *Dijkstra*", *Euler J. Ilm. Mat. Sains dan Teknol.*, vol. 14, no. 1, pp. 151–162, 2026, <https://doi.org/10.37905/euler.v14i1.37524>

© 2026 by author(s)

## JOURNAL INFO • EULER : JURNAL ILMIAH MATEMATIKA, SAINS DAN TEKNOLOGI



	Homepage	:	<a href="http://ejurnal.ung.ac.id/index.php/euler/index">http://ejurnal.ung.ac.id/index.php/euler/index</a>
	Journal Abbreviation	:	Euler J. Ilm. Mat. Sains dan Teknol.
	Frequency	:	Three times a year
	Publication Language	:	English (preferable), Indonesia
	DOI	:	<a href="https://doi.org/10.37905/euler">https://doi.org/10.37905/euler</a>
	Online ISSN	:	2776-3706
	License	:	Creative Commons Attribution-NonCommercial 4.0 International License
	Publisher	:	Department of Mathematics, Universitas Negeri Gorontalo
	Country	:	Indonesia
	OAI Address	:	<a href="http://ejurnal.ung.ac.id/index.php/euler/oai">http://ejurnal.ung.ac.id/index.php/euler/oai</a>
	Google Scholar ID	:	QF_r-gAAAAJ
	Email	:	<a href="mailto:euler@ung.ac.id">euler@ung.ac.id</a>

## JAMBURA JOURNAL • FIND OUR OTHER JOURNALS



Jambura Journal of Biomathematics



Jambura Journal of Mathematics



Jambura Journal of Mathematics Education



Jambura Journal of Probability and Statistics

# Pemodelan Graf Berarah Berbobot untuk Optimasi Penentuan Rute Terpendek Antar Kampus Polimarin Berbasis Algoritma Dijkstra

Helena Devi Ariyani<sup>1,\*</sup>, Khoirotun Nafillah<sup>1</sup>, Kirtyana Nindita<sup>2</sup>, Ngatmin<sup>1</sup>, Sri Tutie Rahayu<sup>2</sup>

<sup>1</sup>Jurusan Teknik, Politeknik Maritim Negeri Indonesia, Semarang 50233, Indonesia

<sup>2</sup>Jurusan Nautika, Politeknik Maritim Negeri Indonesia, Semarang 50233, Indonesia

## ARTICLE HISTORY

Diterima 24 Januari 2026

Direvisi 13 April 2026

Disetujui 18 April 2026

Diterbitkan 23 April 2026

## KATA KUNCI

Algoritma Dijkstra  
Graf Berbobot  
Optimasi Rute  
Rute Terpendek

## KEYWORDS

Dijkstra's Algorithm  
Route Optimization  
Shortest Path  
Weighted Directed Graph

**ABSTRAK.** Mobilitas sivitas akademika antara dua kampus Politeknik Negara Maritim Indonesia (Polimarin), yaitu Kampus 1 Ungaran dan Kampus 2 Bendan Duwur Semarang, menimbulkan permasalahan efisiensi perjalanan akibat banyaknya alternatif rute yang belum teridentifikasi secara terukur. Penelitian ini bertujuan memodelkan jaringan jalan antar kampus dalam bentuk graf berarah berbobot serta menerapkan algoritma Dijkstra untuk menentukan rute terpendek secara akurat dan terverifikasi. Metode yang digunakan adalah penelitian terapan dengan pendekatan komputasional berbasis algoritma Dijkstra, menggunakan data jarak aktual dari Google Maps melalui mode berkendara (driving mode), yang diambil pada 22 Agustus 2024 pukul 09.00 WIB untuk merepresentasikan kondisi lalu lintas saat itu. Tahapan penelitian meliputi identifikasi simpul, pembentukan sisi, dan pemberian bobot berdasarkan jarak aktual, sehingga diperoleh graf dengan 9 simpul dan 10 sisi. Hasil penelitian menunjukkan bahwa rute terpendek adalah  $A \rightarrow B \rightarrow E \rightarrow G \rightarrow H \rightarrow I$  dengan total jarak 25,80 km dari 11 alternatif jalur, melalui proses iterasi sebanyak 8 langkah dengan evaluasi akumulasi bobot minimum pada setiap tahap. Hasil ini menunjukkan bahwa algoritma Dijkstra efektif dalam menentukan rute optimal melalui eliminasi jalur yang tidak efisien. Validasi dilakukan dengan membandingkan hasil algoritma dengan rekomendasi Google Maps berdasarkan parameter jarak tempuh, meskipun penelitian ini masih terbatas pada data statis sehingga pengembangan lebih lanjut diperlukan melalui integrasi data lalu lintas dinamis.

**ABSTRACT.** The mobility of academicians between the two geographically separated campuses of the Indonesian Maritime State Polytechnic (Polimarin), namely Campus 1 Ungaran and Campus 2 Bendan Duwur Semarang, creates travel efficiency issues due to numerous unmeasured route alternatives. This study aims to model the inter-campus road network as a weighted directed graph and apply the Dijkstra algorithm to determine the shortest route accurately and verifiably. The research employs an applied computational approach using actual distance data obtained from Google Maps (driving mode), collected on August 22, 2024, at 09:00 WIB to represent traffic conditions at that time. The process includes node identification (strategic locations), edge formation (connecting road segments), and weight assignment based on actual distances, resulting in a graph with 9 nodes and 10 edges. The iteration process is conducted in 8 steps by evaluating the minimum accumulated weight at each stage. The results indicate that the shortest path is  $A \rightarrow B \rightarrow E \rightarrow G \rightarrow H \rightarrow I$ , passing through Jl. PTP Ngobo, Jl. Diponegoro–Jl. Slamet Riyadi, Jl. Moh. Yamin–Jl. Ahmad Yani, and Jl. Gatot Subroto, with a total distance of 25.80 km out of 11 possible routes. These findings demonstrate that the Dijkstra algorithm is effective for route optimization by eliminating inefficient paths through cumulative weight evaluation. Validation is performed by comparing the algorithm's results with Google Maps recommendations based on travel distance. However, this study is limited to static data, and further development is required through the integration of dynamic traffic data.



This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial 4.0 International License. **Editorial of EULER:** Department of Mathematics, Universitas Negeri Gorontalo, Jln. Prof. Dr. Ing. B. J. Habibie, Bone Bolango 96554, Indonesia.

\*Penulis Korespondensi.

## 1. Pendahuluan

Mobilitas antar lokasi merupakan bagian penting dalam aktivitas sehari-hari, terutama pada lingkungan dengan lebih dari satu pusat kegiatan seperti institusi pendidikan yang memiliki kampus terpisah. Kemampuan menentukan jalur perjalanan yang paling optimal karenanya menjadi aspek krusial yang telah menarik perhatian para peneliti sejak dekade 1950-an [1]. Seiring dengan pesatnya perkembangan ilmu pengetahuan dan teknologi, persoalan penentuan rute terpendek atau yang dikenal sebagai *shortest path problem* berkembang menjadi kajian multidisiplin yang sangat relevan dalam berbagai bidang, mulai dari navigasi, transportasi, logistik, hingga pengembangan sistem informasi berbasis digital [2].

Permasalahan pencarian rute terpendek muncul karena antara dua titik lokasi tidak hanya terdapat satu jalur tunggal, melainkan terbentuk suatu jaringan yang terdiri dari beragam alternatif rute dengan karakteristik yang berbeda-beda. Kondisi ini menjadikan pendekatan konvensional menggunakan peta fisik sebagai metode yang kurang efisien, sebab pemilihan rute dilakukan secara manual, memerlukan waktu yang panjang, serta memiliki potensi kesalahan estimasi yang cukup besar [3]. Oleh karena itu, diperlukan suatu pendekatan yang lebih sistematis dan terkomputasi untuk menghasilkan solusi pencarian rute yang akurat, cepat, dan dapat diandalkan. Dalam era perkembangan teknologi berbasis web, kebutuhan akan sistem pencarian rute yang terintegrasikan dalam platform digital semakin meningkat, di mana kemampuan pengolahan data secara dinamis menjadi landasan penting dalam pengembangan aplikasi semacam ini [4].

Salah satu pendekatan yang terbukti efektif dalam menyelesaikan permasalahan tersebut adalah dengan merepresentasikan jaringan jalan ke dalam struktur matematika berupa *graph*. Dalam representasi *graph*, setiap persimpangan atau titik lokasi dimodelkan sebagai simpul, sedangkan ruas jalan yang menghubungkan antartitik dimodelkan sebagai *edge* (sisi) yang memiliki bobot (*weight*) berupa jarak, waktu, atau biaya yang diperhitungkan [5]. Pendekatan berbasis *graph* ini memungkinkan penerapan berbagai algoritma komputasional untuk mencari lintasan terpendek secara sistematis, terstruktur, dan terukur sehingga menghasilkan keputusan yang lebih objektif dibandingkan metode manual [6].

Di antara berbagai algoritma yang telah dikembangkan, *Dijkstra's Algorithm* dikenal sebagai salah satu metode paling andal dan paling luas diimplementasikan dalam penentuan jalur terpendek pada *directed weighted graph* [7]. Algoritma ini bekerja secara iteratif dengan mengevaluasi bobot kumulatif dari setiap lintasan yang mungkin dilalui, kemudian secara bertahap memilih simpul dengan total bobot terkecil hingga ditemukan jalur paling optimal menuju titik tujuan [8]. Keunggulan utama algoritma *Dijkstra* terletak pada kemampuannya meminimalkan total biaya perjalanan dari titik asal menuju titik akhir, sehingga algoritma ini banyak dimanfaatkan dalam sistem navigasi *real-time*, *routing* jaringan internet, hingga berbagai aplikasi transportasi berbasis teknologi digital [9].

Implementasi algoritma *Dijkstra* dalam konteks nyata telah banyak dilaporkan dan dikaji dalam berbagai penelitian terdahulu. Gusmao et al. [10] menggunakannya dalam pengembangan Sistem Informasi Geografis (SIG) berbasis web untuk mendukung pencarian jalur wisata terpendek secara interaktif. Ichsan [11] menerapkan algoritma serupa pada pemetaan dan rekomendasi objek pariwisata di Pulau Bali berdasarkan kedekatan jarak antarlokasi. Sementara itu, Sebayang dan Rosyida [12] mengintegrasikan algoritma *Dijkstra* dengan *Fuzzy Inference System* untuk menentukan rute ojek online sekaligus menetapkan tarif secara dinamis berdasarkan jarak tempuh dan kondisi medan yang dilalui. Penerapan algoritma untuk menentukan rute menuju lokasi wisata [14, 15] menunjukkan efektivitasnya dalam menghasilkan lintasan optimal berbasis jarak. Penerapan pada bidang distribusi logistik juga dilakukan oleh

Lusiani et al. [16], yang menegaskan bahwa algoritma ini dapat mendukung efisiensi distribusi barang. Selain itu, pengembangan lebih lanjut terhadap algoritma juga telah dilakukan, seperti modifikasi bobot yang disesuaikan secara dinamis oleh Jabbar et al. [17], serta penerapan pada jaringan yang lebih luas seperti rute wisata perkotaan oleh Sipayung et al. [18]. Penelitian oleh Suardinata et al. [19] mengintegrasikan algoritma *Dijkstra* dengan *Google Maps* untuk menentukan waktu tempuh dan rute tercepat. Di sisi lain, studi komparatif oleh Ardiansyah et al. [20] membandingkan performa algoritma *Dijkstra* dengan  $A^*$ , sementara He [21] menegaskan secara konseptual bahwa algoritma *Dijkstra* merupakan salah satu algoritma fundamental dalam permasalahan pencarian jalur terpendek pada graf berbobot. Berbagai penelitian tersebut secara konsisten menunjukkan bahwa algoritma *Dijkstra* mampu menghasilkan solusi yang optimal, efisien, dan dapat diimplementasikan dalam berbagai kondisi jaringan jalan yang kompleks.

Sejumlah penelitian sebelumnya menunjukkan bahwa algoritma ini dapat digunakan untuk menentukan lintasan terpendek dalam berbagai konteks, seperti sistem informasi geografis, pariwisata, dan transportasi. Namun, sebagian besar penelitian tersebut berfokus pada implementasi algoritma tanpa memberikan evaluasi yang jelas terhadap kesesuaian hasil dengan kondisi rute aktual di lapangan atau perbandingan dengan sistem navigasi yang umum digunakan. Berdasarkan hal tersebut, terdapat keterbatasan kajian yang menghubungkan hasil perhitungan algoritma dengan data rute nyata sebagai acuan evaluasi. Selain itu, implementasi pada skala jaringan sederhana dengan data aktual yang spesifik masih jarang dibahas sebagai dasar analisis awal.

Oleh karena itu, penelitian ini bertujuan untuk menerapkan algoritma *Dijkstra* dalam menentukan rute terpendek antara Kampus 1 dan Kampus 2 Polimarin dengan menggunakan representasi graf berbasis data jarak aktual. Hasil perhitungan algoritma kemudian dibandingkan dengan rute yang direkomendasikan oleh *Google Maps* sebagai bentuk evaluasi untuk melihat tingkat kesesuaian hasil. Berdasarkan permasalahan yang telah diuraikan, penelitian ini difokuskan untuk membangun representasi jaringan jalan antara Kampus 1 dan Kampus 2 Politeknik Maritim Negeri Indonesia (Polimarin) dalam bentuk graf berarah berbobot yang sesuai kondisi aktual. Selain itu, penelitian ini juga mengkaji penerapan algoritma *Dijkstra* dalam menentukan rute terpendek berdasarkan data jarak yang telah dimodelkan, serta mengevaluasi tingkat kesesuaian hasil perhitungan dengan rute referensi dari *Google Maps*. Penelitian ini diharapkan dapat memberikan gambaran mengenai penerapan algoritma pencarian rute terpendek pada kasus nyata skala terbatas serta menjadi dasar pengembangan sistem navigasi yang lebih lanjut.

## 2. Metode

Penelitian ini merupakan penelitian terapan yang berfokus pada implementasi dan evaluasi algoritma *Dijkstra* dalam penentuan rute terpendek. Data jarak diperoleh dari *Google Maps* dan dimodelkan dalam bentuk graf berbobot. Hasil perhitungan algoritma kemudian dibandingkan dengan rute referensi untuk menilai tingkat kesesuaian secara empiris.

Metode yang diterapkan dalam penelitian ini adalah *Dijkstra's Algorithm*, yaitu suatu prosedur komputasional yang digunakan untuk menentukan lintasan terpendek dari satu titik awal menuju titik-titik lainnya dalam suatu jaringan graf berbobot (*weighted graph*). Algoritma ini bekerja dengan cara mengidentifikasi jalur paling optimal secara bertahap, di mana pada setiap iterasi ke- $n$ , terdapat minimal  $n$  simpul yang jalur terpendeknya telah berhasil ditentukan [13]. Jaringan jalan dimodelkan ke dalam graf yang terdiri dari 9 simpul dan 10 sisi. Pemilihan jumlah simpul dan sisi tersebut didasarkan pada penyederhanaan jaringan jal-

an utama yang merepresentasikan titik–titik persimpangan penting dan jalur alternatif yang relevan antara Kampus 1 dan Kampus 2, sehingga model tetap representatif namun terkontrol untuk keperluan analisis. Data bobot berupa jarak antar simpul diperoleh dari *Google Maps* menggunakan rute berkendara (*driving mode*). Pengambilan data dilakukan pada tanggal 22 Agustus 2024 pukul 09.00 WIB dengan asumsi kondisi lalu lintas relatif normal. Data tersebut kemudian digunakan sebagai bobot pada setiap sisi dalam graf.

Selanjutnya, titik awal ditetapkan sebagai simpul keberangkatan, kemudian dilakukan pengukuran jarak dari titik tersebut menuju seluruh simpul yang ada dalam jaringan. Pada tahap berikutnya, dilakukan pertimbangan terhadap simpul-simpul lain yang jarak kumulatifnya dari titik awal belum dihitung. Sebagai ilustrasi, apabila jarak dari simpul  $O$  menuju simpul  $A$  adalah 5 satuan, dan dari simpul  $A$  menuju simpul  $E$  adalah 3 satuan, maka total jarak kumulatif dari titik asal menuju simpul  $E$  melalui simpul  $A$  adalah  $5 + 3 = 8$  satuan. Nilai bobot jarak antar simpul dikalkulasikan sesuai dengan bobotnya, dan nilai jarak yang tersimpan selalu merepresentasikan jarak terpendek terkini, sehingga tidak diperlukan perhitungan ulang terhadap informasi sebelumnya. Prinsip ini menjamin efisiensi komputasi algoritma secara keseluruhan.

Sebagai langkah penutup setiap iterasi, simpul yang belum dikunjungi (*unvisited*) dengan nilai jarak terkecil dari titik asal ditetapkan sebagai simpul keberangkatan berikutnya, kemudian seluruh proses diulang kembali dari tahap identifikasi simpul yang belum terhitung. Pendekatan ini terbukti efektif dalam mengoptimalkan penentuan rute pada lingkungan berbasis graf, termasuk dalam konteks navigasi antar lokasi kampus [14].

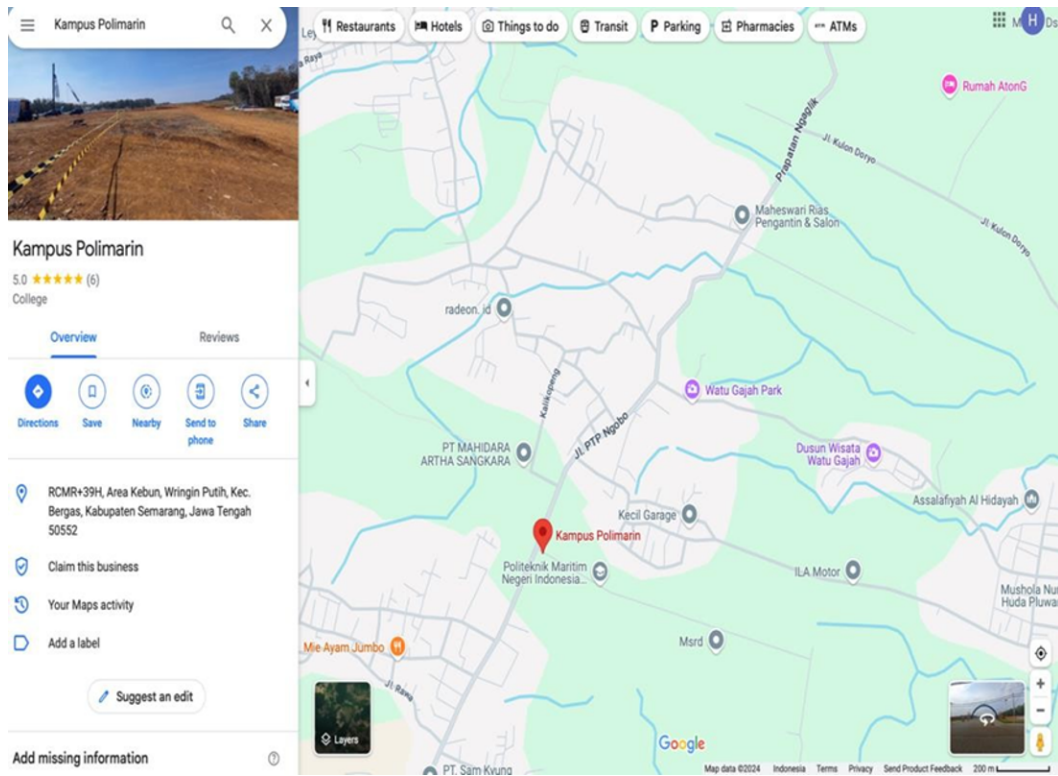
### 3. Hasil dan Pembahasan

Pada bab ini disajikan hasil pemodelan jaringan jalan antar Kampus 1 dan Kampus 2 Polimarin ke dalam bentuk graf berarah berbobot, serta hasil penerapan Algoritma *Dijkstra* untuk menentukan rute terpendek berdasarkan data jarak aktual. Penyajian hasil tidak hanya difokuskan pada perolehan lintasan optimum, tetapi juga pada proses pembentukan model, penentuan bobot antar simpul, dan tahapan iterasi algoritma hingga diperoleh simpulan rute yang paling efisien. Dengan demikian, pembahasan pada bagian ini bertujuan menunjukkan bahwa representasi graf yang dibangun mampu menggambarkan kondisi jaringan jalan secara sistematis dan mendukung proses penentuan rute secara terukur. Selanjutnya, hasil komputasi yang diperoleh dianalisis untuk melihat kesesuaian model dengan kondisi rute nyata, sehingga dapat memberikan gambaran mengenai efektivitas penggunaan Algoritma *Dijkstra* pada kasus penentuan rute antar kampus Polimarin.

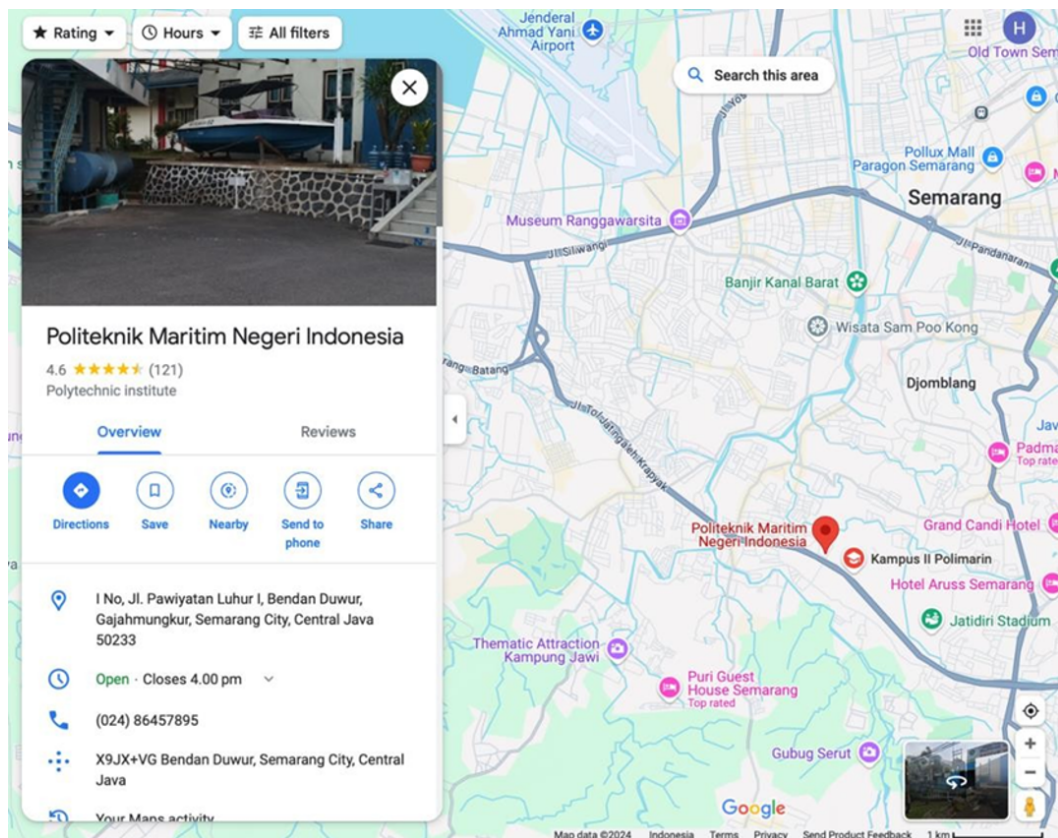
#### 3.1. Algoritma *Dijkstra*

Pada tahap awal, dilakukan penentuan titik asal dan titik tujuan sebagai dasar dalam proses analisis rute terpendek. Penetapan kedua titik ini penting karena menjadi acuan utama dalam pembentukan lintasan yang akan dianalisis menggunakan Algoritma *Dijkstra*. Titik asal merepresentasikan lokasi dimulainya perjalanan, sedangkan titik tujuan menunjukkan lokasi akhir yang hendak dicapai. Dalam penelitian ini, titik asal ditetapkan pada Kampus 1 Polimarin yang berlokasi di Jl. PTP Ngobo, Area Kebun, Wringin Putih, Kecamatan Bergas, Semarang, sedangkan titik tujuan adalah Kampus 2 Polimarin di Bendan Duwur, Semarang. Penentuan titik awal dan titik akhir tersebut dilakukan untuk membatasi ruang analisis pada jalur yang relevan sesuai dengan kebutuhan mobilitas antar kampus. **Gambar 1** dan **Gambar 2** menyajikan representasi visual lokasi yang digunakan sebagai titik keberangkatan dan titik tujuan dalam penelitian ini. Visualisasi tersebut membantu memperjelas posisi geografis kedua kampus

sekaligus menjadi dasar dalam identifikasi simpul-simpul yang akan digunakan pada tahap pemodelan graf.



Gambar 1. Titik Asal (Titik Awal) Kampus Polimarin 1 Ungaran (diakses dari <https://maps.google.com>)



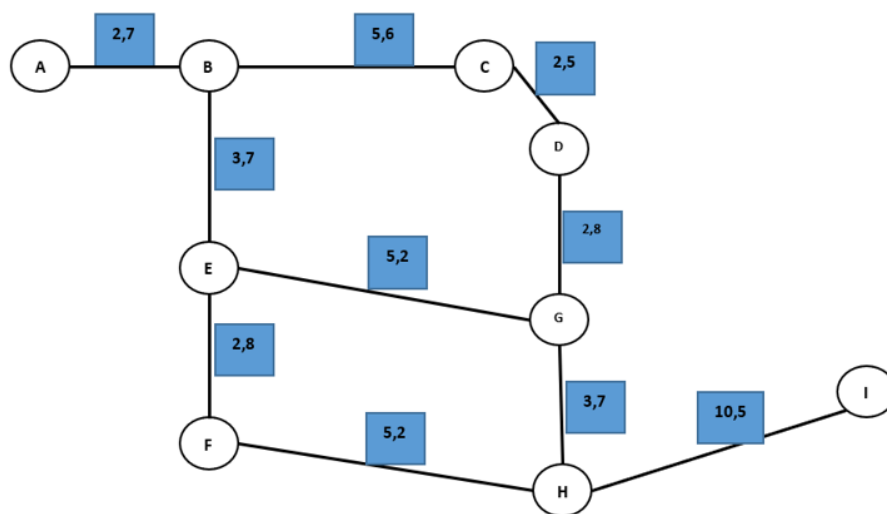
Gambar 2. Destination Point Kampus Polimarin 2 Semarang (diakses dari <https://maps.google.com>)

Pada tahap berikutnya, dilakukan penentuan dan identifikasi titik–titik rute yang akan dilalui sebagai bagian dari proses pemodelan jaringan jalan. Pada langkah ini, *Google Maps* digunakan untuk mengidentifikasi titik rute dari titik awal ke titik tujuan. Titik-titik rute direpresentasikan menggunakan pemetaan simpul untuk setiap lokasi, bersama dengan informasi deskriptif untuk memfasilitasi presentasinya dalam bentuk graf berarah. Hal ini dapat dilihat pada **Tabel 1**.

**Tabel 1.** Titik awal, tujuan, dan jalur

Simpul	Lokasi	Label
A	Kampus 1 Polimarin Ungaran	Titik Awal
B	Persimpangan Jalan PTP Ngobo menuju Leyangan – Jalan PTP Ngobo menuju Karangjati	Jalan
C	Jalan Tol Tembalang – Jalan Jatiraya	Jalan
D	Jalan Tol Ungaran – Jalan Letjen Suprpto	Jalan
E	Persimpangan Jalan Diponegoro dan Jalan Slamet Riyadi	Jalan
F	Persimpangan Jalan HOS Cokroaminoto dan Jalan Brigjen Suciarto	Jalan
G	Persimpangan Jalan Moh. Yamin dan Jalan Ahmad Yani	Jalan
H	Persimpangan Jalan Gatot Subroto dan Jalan Moh. Yamin	Jalan
I	Kampus Polimarin Bendan Duwur 2	Titik Tujuan

Tahap berikutnya adalah penentuan bobot pada setiap ruas jalan berdasarkan jarak tempuh antar titik. Bobot ini digunakan sebagai parameter dalam model graf berarah berbobot untuk merepresentasikan besaran jarak pada setiap sisi, sehingga mendukung proses perhitungan rute terpendek. Berdasarkan semua data yang diperoleh dari *Google Maps*, informasi tersebut kemudian direpresentasikan menggunakan graf berarah seperti pada **Gambar 3**.



**Gambar 3.** Graf dengan 9 simpul dan 10 tepi

Selanjutnya, dilakukan proses iterasi dalam perhitungan bobot untuk menentukan jalur terpendek ke titik tujuan. Pada langkah ini, peneliti melakukan perhitungan berulang menggunakan algoritma *Dijkstra* dalam proses *looping*, mulai dari titik keberangkatan awal hingga lokasi tujuan. Iterasi algoritma *Dijkstra* dilakukan dengan mempertimbangkan rute atau jalur perjalanan terpendek dari Kampus 1 Polimarin menuju Kampus 2 di Semarang. **Tabel 2** mewakili langkah perhitungan algoritma *Dijkstra* yang digunakan untuk menentukan jarak terpendek dari titik awal (A) ke titik tujuan (I). Untuk setiap langkah dari simpul yang dipilih, tambahkan nilai terkecil sebelumnya ke semua jarak dari simpul yang dipilih ke simpul yang terhubung, seperti yang dijelaskan di bawah ini.

**Tabel 2.** Jarak untuk setiap simpul dalam graf berarah

Dari	Ke	Jarak (km)
A	B	2,70
B	C	5,60
B	E	3,70
C	D	2,50
D	F	2,80
E	G	5,20
E	F	2,80
F	H	5,20
G	H	3,70
H	I	10,50

- Langkah 1. Perhitungan dimulai dari titik awal (simpul A). Simpul A hanya terhubung ke simpul B dengan jarak 2,70, sehingga jalur terpendek yang dipilih adalah dari simpul A ke simpul B.
- Langkah 2. Perhitungan berlanjut dari simpul dengan nilai terpendek pada langkah 1, yaitu simpul B. Simpul B terhubung ke dua simpul, yaitu simpul C dengan jarak total 8,30 dan simpul E dengan jarak 6,40. Jarak total diperoleh dengan menambahkan jarak dari simpul A ke simpul B dan jarak dari simpul B ke simpul yang terhubung (simpul C dan E). Jarak terpendek pada langkah 2 adalah dari simpul B ke simpul E, sehingga jalur yang dipilih adalah  $A \rightarrow B \rightarrow E$ .
- Langkah 3. Perhitungan berlanjut dari simpul dengan nilai terpendek pada langkah 2 yang belum dipilih, yaitu simpul E. Simpul E terhubung ke beberapa simpul, di antaranya simpul G dengan total jarak 11,60. Jarak total diperoleh dengan menambahkan jarak dari simpul A ke simpul B, simpul B ke simpul E, dan simpul E ke simpul yang terhubung. Jarak terpendek pada langkah 3 adalah dari simpul E ke simpul G dengan jarak 9,20, sehingga jalur yang dipilih adalah  $A \rightarrow B \rightarrow E \rightarrow G$ .
- Langkah 4. Perhitungan dilanjutkan dari simpul dengan nilai terpendek pada langkah 3 yang belum dipilih, yaitu simpul G. Simpul G terhubung ke beberapa simpul, termasuk simpul H dengan total jarak 14,40. Jarak total diperoleh dengan menambahkan jarak dari simpul A ke simpul B, simpul B ke simpul E, simpul E ke simpul G, dan simpul G ke simpul yang terhubung. Jarak terpendek pada langkah 4 adalah dari simpul G ke simpul H dengan jarak 3,70, sehingga jalur yang dipilih adalah  $A \rightarrow B \rightarrow E \rightarrow G \rightarrow H$ . Proses dilanjutkan hingga tidak ada lagi simpul yang dapat dipilih.

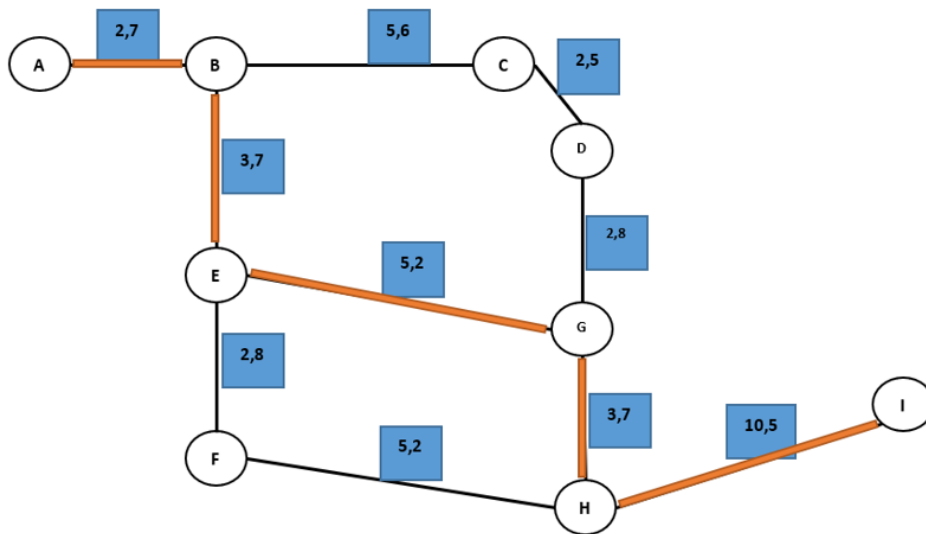
Iterasi yang dilakukan pada tahapan ini diperoleh melalui penerapan algoritma *Dijkstra* seperti yang terangkum pada **Tabel 3**. Dari hasil perhitungan Algoritma *Dijkstra*, ditemukan bahwa jalur terpendek adalah  $A \rightarrow B \rightarrow E \rightarrow G \rightarrow H \rightarrow I$  dengan total jarak 25,80 km, seperti yang ditunjukkan pada **Gambar 4**.

Hasil perhitungan menunjukkan bahwa rute tersebut merupakan jalur terpendek yang dapat ditempuh dari Kampus 1 Polimarin menuju Kampus 2 Polimarin. Uraian rute yang dilalui adalah sebagai berikut:

- Kampus Polimarin 1,
- Persimpangan Jl. PTP Ngobo menuju Leyangan – Jl. PTP Ngobo menuju Karangjati,
- Persimpangan Jl. Diponegoro – Jl. Slamet Riyadi,
- Persimpangan Jl. Moh. Yamin – Jl. Ahmad Yani,
- Persimpangan Jl. Gatot Subroto – Jl. Moh. Yamin,

**Tabel 3.** Perhitungan Algoritma Dijkstra

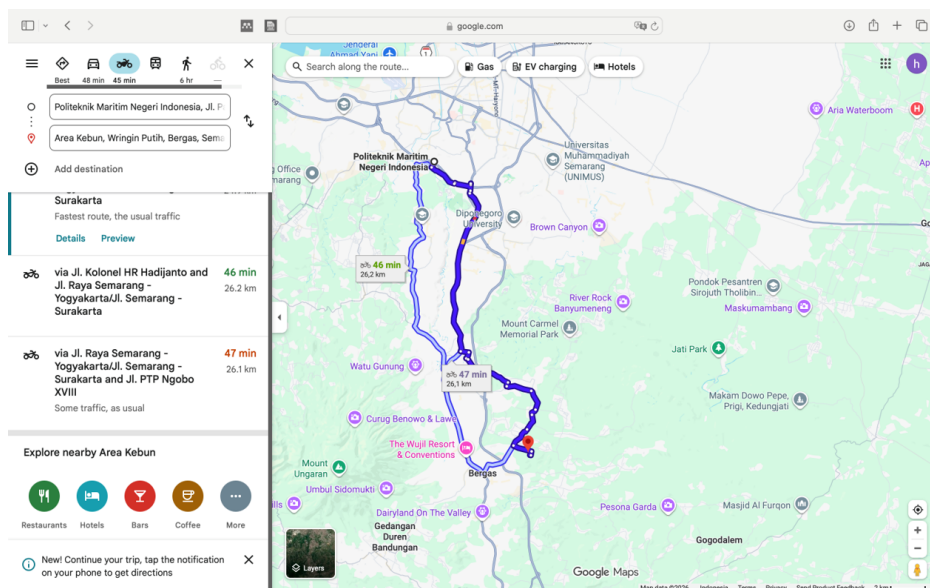
Langkah	Simpul	A	B	C	D	E	F	G	H	I
1	A	0	2,70	~	~	~	~	~	~	~
2	B	0	2,70	8,30	~	6,40	~	~	~	~
3	C	0	2,70	8,30	~	6,40	11,40	9,20	~	~
4	D	0	2,70	8,30	10,80	6,40	11,40	9,20	~	~
5	E	0	2,70	8,30	10,80	6,40	11,40	9,20	14,40	~
6	F	0	2,70	8,30	10,80	6,40	11,40	9,20	14,40	~
7	G	0	2,70	8,30	10,80	6,40	11,40	9,20	14,40	~
8	H	0	2,70	8,30	10,80	6,40	11,40	9,20	14,40	25,80
9	I	0	2,70	8,30	10,80	6,40	11,40	9,20	14,40	25,80



**Gambar 4.** Graf dengan jalur terpendek

6. Kampus 2 Polimarin.

Selanjutnya dilakukan perbandingan dengan rute *Google Maps*. Tangkapan layar jarak antar kampus ditunjukkan pada Gambar 5.



**Gambar 5.** Jarak Kampus 1 ke Kampus 2 Polimarin dari *Google Maps* (diakses 22 Agustus 2024 pukul 09.00 WIB)

Berdasarkan hasil evaluasi, jarak rute yang diperoleh menggunakan Algoritma *Dijkstra* adalah 25,80 km, sedangkan jarak berdasarkan *Google Maps* adalah 26,20 km. Dengan demikian, Algoritma *Dijkstra* menghasilkan rute yang lebih pendek dibandingkan rute referensi *Google Maps*, sehingga lebih optimal dalam hal jarak tempuh.

### 3.2. Pembahasan

#### 3.2.1. Representasi Geografis Rute Polimarin dalam Bentuk Graf

Pemodelan representasi geografis antara Kampus 1 dan Kampus 2 Polimarin dilakukan dengan mengonversi kondisi nyata jaringan jalan ke dalam struktur graf berarah berbobot (*directed weighted graph*). Pendekatan ini sejalan dengan pernyataan Sari bahwa algoritma *Dijkstra* bekerja secara optimal pada graf yang setiap simpulnya merepresentasikan lokasi nyata dengan bobot yang mencerminkan jarak tempuh aktual [8]. Dalam penelitian ini, sembilan titik lokasi diidentifikasi menggunakan *Google Maps* dan dipetakan sebagai simpul, mulai dari Simpul A (Kampus Polimarin 1 Ungaran) hingga Simpul I (Kampus Polimarin 2 Bendan Duwur), dengan sepuluh sisi (*edge*) berbobot yang merepresentasikan ruas jalan penghubung antar simpul. Proses identifikasi titik rute ini selaras dengan metode yang diterapkan oleh Yuliani et al., yang memodelkan jaringan destinasi wisata Bandung ke dalam graf berbobot sebelum menerapkan algoritma *Dijkstra* [15]. Hasil pemodelan menghasilkan graf dengan total sebelas kemungkinan jalur, di mana setiap sisi memiliki nilai bobot berbeda yang mencerminkan jarak dalam satuan kilometer. Pemodelan ini memastikan bahwa seluruh kondisi geografis jalan utama dari Ungaran menuju Semarang dapat direpresentasikan secara sistematis dan kompatibel dengan kebutuhan komputasi algoritma *Dijkstra*.

#### 3.2.2. Penerapan Algoritma *Dijkstra* dalam Penentuan Rute Terpendek

Penerapan algoritma *Dijkstra* dilakukan secara iteratif, dimulai dari Simpul A sebagai titik keberangkatan dengan nilai jarak awal nol, kemudian menelusuri seluruh simpul yang terhubung secara bertahap. Lusiani et al. menjelaskan bahwa algoritma *Dijkstra* bekerja dengan memilih simpul berbobot terkecil pada setiap iterasi hingga simpul tujuan tercapai, sehingga menjamin diperolehnya jalur dengan total bobot minimum [16]. Proses komputasi dalam penelitian ini menghasilkan delapan langkah iterasi, di mana pada setiap langkah dipilih simpul dengan akumulasi jarak terpendek yang belum dieksplorasi. Jabbar et al. (2022) menegaskan bahwa pembaruan nilai bobot secara dinamis pada setiap iterasi menjadikan algoritma *Dijkstra* mampu mengakomodasi perubahan pada topologi jaringan jalan [17]. Berdasarkan hasil perhitungan, jalur terpendek yang diperoleh adalah  $A \rightarrow B \rightarrow E \rightarrow G \rightarrow H \rightarrow I$  dengan total jarak 25,80 km. Rute ini melewati Persimpangan Jl. PTP Ngobo, Jl. Diponegoro–Jl. Slamet Riyadi, Jl. Moh. Yamin–Jl. Ahmad Yani, hingga Jl. Gatot Subroto. Temuan ini membuktikan bahwa algoritma *Dijkstra* mampu menentukan rute optimal secara akurat dan terukur pada jaringan jalan nyata [18].

#### 3.2.3. Faktor-Faktor yang Mempengaruhi Pencarian Rute Terpendek

Terdapat beberapa faktor yang secara signifikan mempengaruhi kualitas hasil pencarian rute terpendek menggunakan algoritma *Dijkstra*. Pertama, akurasi data bobot jarak pada setiap sisi graf menjadi penentu utama validitas hasil komputasi, sebagaimana diungkapkan Suardinata bahwa sumber data dari *Google Maps* memberikan informasi jarak yang terukur dan dapat diverifikasi [19]. Kedua, kelengkapan pemodelan simpul dan sisi graf turut menentukan representasi jaringan jalan. Ardiansyah et al. menyatakan bahwa semakin komprehensif struktur graf yang dibangun, semakin tinggi tingkat akurasi jalur optimal yang dihasilkan [20].

Ketiga, kompleksitas topologi jaringan jalan, termasuk jumlah persimpangan dan alternatif jalur, berpengaruh terhadap efisiensi komputasi algoritma. Sipayung et al. [18] menambahkan bahwa pemilihan simpul sumber dan tujuan yang tepat sangat menentukan relevansi hasil akhir pencarian rute. Dalam konteks penelitian ini, ketiga faktor tersebut telah dikelola dengan baik melalui penggunaan data *Google Maps* yang akurat, pemodelan graf yang representatif, serta penetapan sembilan simpul yang mencakup seluruh koridor jalan utama antara Kampus Polimarin 1 Ungaran dan Kampus Polimarin 2 Bendan Duwur Semarang.

#### 3.2.4. Analisis Evaluasi Hasil

Analisis hasil pada penelitian ini dilakukan untuk mengevaluasi kinerja model graf berarah berbobot yang telah dibangun dalam merepresentasikan jaringan jalan serta dalam menghasilkan rute terpendek antar kampus. Evaluasi dilakukan dengan membandingkan hasil perhitungan berupa lintasan dan total jarak yang diperoleh melalui proses komputasi dengan data referensi dari *Google Maps* pada parameter yang sama.

Hasil evaluasi menunjukkan bahwa total jarak yang diperoleh melalui algoritma *Dijkstra* lebih pendek dibandingkan dengan jarak pada referensi *Google Maps*.

##### 1. Sensitivitas Bobot

Model yang digunakan dalam penelitian ini mengasumsikan bahwa bobot sisi bersifat tetap dan direpresentasikan oleh jarak. Namun, dalam praktiknya, nilai bobot tersebut memiliki sensitivitas terhadap perubahan kecil yang dapat mempengaruhi hasil akhir lintasan terpendek. Pada beberapa jalur dengan selisih jarak yang relatif kecil, perubahan bobot misalnya akibat pembaruan data peta atau perbedaan pengukuran dapat menyebabkan perubahan jalur optimal yang dipilih oleh algoritma *Dijkstra*. Hal ini menunjukkan bahwa stabilitas solusi sangat bergantung pada distribusi bobot dalam graf.

##### 2. Alternatif Rute

Graf yang dibangun menghasilkan beberapa alternatif lintasan (11 rute) dari titik asal ke tujuan. Keberadaan alternatif ini menunjukkan bahwa sistem tidak hanya memiliki satu solusi, melainkan sekumpulan solusi dengan tingkat optimalitas yang berbeda.

##### 3. Kondisi Lalu Lintas Nyata

Salah satu keterbatasan utama model adalah penggunaan bobot berbasis jarak statis yang tidak mempertimbangkan dinamika lalu lintas. Dalam sistem navigasi nyata seperti *Google Maps*, penentuan rute tidak hanya berdasarkan jarak, tetapi juga mempertimbangkan waktu tempuh, kepadatan lalu lintas, dan kondisi jalan secara *real-time*. Pengambilan data yang dilakukan pada satu waktu tertentu (22 Agustus 2024 pukul 09.00 WIB) hanya merepresentasikan kondisi lalu lintas pada saat itu, sehingga tidak mencerminkan variasi temporal seperti jam sibuk atau kondisi insidental seperti kemacetan atau kecelakaan.

## 4. Kesimpulan

Penelitian ini menunjukkan bahwa Algoritma *Dijkstra* dapat diterapkan untuk menentukan rute terpendek antara Kampus 1 Polimarin Ungaran dan Kampus 2 Polimarin Bendan Duwur Semarang melalui pemodelan graf berarah berbobot yang terdiri dari 9 simpul (A–I) dan 10 sisi dengan bobot jarak yang diperoleh dari *Google Maps*. Berdasarkan proses komputasi yang dilakukan, diperoleh lintasan  $A \rightarrow B \rightarrow E \rightarrow G \rightarrow H \rightarrow I$ , melewati Persimpangan Jl. PTP Ngo-bo, Jl. Diponegoro–Jl. Slamet Riyadi, Jl. Moh. Yamin–Jl. Ahmad Yani, dan Jl. Gatot Subroto, dengan total jarak 25,80 kilometer dari sejumlah alternatif rute yang dimodelkan. Hasil ini memberikan jarak yang lebih pendek dibandingkan dengan jarak yang dihasilkan oleh *Google Maps*, yaitu sebesar 26,20 kilometer.

Implikasi dari temuan ini menunjukkan bahwa pemodelan graf berarah berbobot yang dikombinasikan dengan Algoritma *Dijkstra* dapat digunakan sebagai pendekatan komputasional yang sistematis dalam mendukung penentuan rute terpendek. Namun demikian, hasil yang diperoleh bersifat kontekstual terhadap model graf yang digunakan dan tidak secara langsung merepresentasikan keseluruhan kondisi jaringan jalan yang sebenarnya. Penyederhanaan jumlah simpul dan sisi, penggunaan bobot statis berbasis jarak, serta tidak dipertimbangkannya dinamika lalu lintas menyebabkan hasil ini lebih tepat dipandang sebagai ilustrasi penerapan algoritma.

**Kontribusi Penulis.** Helena Devi Ariyani: Konseptualisasi, metodologi, pengembangan perangkat lunak, validasi, analisis formal, investigasi, visualisasi, penulisan draf awal. Khoirotun Nafillah: Kurasi data, analisis formal, investigasi. Kirtyana Nindita: Pengembangan perangkat lunak, implementasi dan perhitungan algoritma *Dijkstra*. Ngatmin: Penyuntingan naskah, supervisi. Sri Tutie Rahayu: Penyuntingan naskah, supervisi. Seluruh penulis telah membaca dan menyetujui versi akhir manuskrip yang diterbitkan.

**Ucapan Terima Kasih.** Para penulis mengucapkan terima kasih kepada Politeknik Maritim Negeri Indonesia karena telah membiayai penelitian ini. Kami juga menghargai editor dan reviewer atas masukan serta dukungannya dalam menyempurnakan karya ini.

**Pembiayaan.** Penelitian ini didanai oleh Politeknik Maritim Negeri Indonesia.

**Konflik Kepentingan.** Para penulis menyatakan tidak ada konflik kepentingan yang terkait dengan artikel ini.

**Ketersediaan Data.** Tidak tersedia.

## Referensi

- [1] M. K. Harahap and N. Khairina, "Pencarian Jalur Terpendek dengan Algoritma Dijkstra," *Sinkron*, vol. 2, no. 2, p. 18, 2019, doi: [10.33395/sinkron.v2i2.61](https://doi.org/10.33395/sinkron.v2i2.61).
- [2] A. D. Sabilla and A. Taufiq, "Penerapan Algoritma A\* Pada WebGIS Pencarian Rute Terpendek," *J. Inf. Syst. Comput.*, vol. 2, no. 2, pp. 32–35, 2022, doi: [10.34001/jister.v2i2.395](https://doi.org/10.34001/jister.v2i2.395).
- [3] S. Sanan, L. Jain, and B. Kapoor, "Shortest Path Algorithm," *Int. J. Appl. or Innov. Eng. Manag.*, vol. 2, no. 7, pp. 316–320, 2013.
- [4] B. Nugroho, *Aplikasi Pemrograman WEB Dinamis dengan PHP dan MySQL*. Gava Media, 2019.
- [5] R. D. Gunawan, R. Napianto, R. I. Borman, and I. Hanifah, "Implementation of Dijkstra's Algorithm in Determining the Shortest Path," *IJISCS*, vol. 3, no. 3, p. 98, 2019, doi: [10.56327/ijiscs.v3i3.768](https://doi.org/10.56327/ijiscs.v3i3.768).
- [6] Y. Yang, "Website Internal Link Optimization Strategy and SEO Effect Evaluation Based on Dijkstra Algorithm," *J. Comput. Signal Syst. Res.*, vol. 2, no. 3, p. 1, 2025, doi: [10.71222/151eah87](https://doi.org/10.71222/151eah87).
- [7] D. Wahyuningsih and E. Syahreza, "Shortest Path Search Futsal Field Location With Dijkstra Algorithm," *IJCCS*, vol. 12, no. 2, p. 161, 2018, doi: [10.22146/ijccs.34513](https://doi.org/10.22146/ijccs.34513).
- [8] I. P. Sari, M. F. Fahroza, M. I. Mufit, and I. F. Qathrunad, "Implementation of Dijkstra's Algorithm to Determine the Shortest Route in a City," *J. Comput. Sci. Inf. Technol. Telecommun. Eng.*, vol. 2, no. 1, pp. 134–138, 2021, doi: [10.30596/jcositte.v2i1.6503](https://doi.org/10.30596/jcositte.v2i1.6503).
- [9] V. N. C. Sebayang and I. Rosyida, "Implementations of Dijkstra Algorithm for Searching the Shortest Route," in *IC-MaGeStiC 2021*, 2022, pp. 76–84, doi: [10.2991/acsr.k.220202.016](https://doi.org/10.2991/acsr.k.220202.016).
- [10] A. Gusmao, S. H. Pramono, and S. Sunaryo, "Sistem Informasi Geografis Pariwisata Berbasis Web dan Pencarian Jalur Terpendek dengan Algoritma Dijkstra," *J. EECCIS*, vol. 7, no. 2, pp. 125–130, 2013, doi: [10.21776/jeeccis.v7i2.214](https://doi.org/10.21776/jeeccis.v7i2.214).
- [11] C. Ichsan, "Aplikasi Android Pencarian Mikrolet Kota Surabaya dengan Algoritma Dijkstra," *Ubiquitous Comput. its Appl. J.*, vol. 1, pp. 19–24, 2018.
- [12] V. N. C. Sebayang and I. Rosyida, "Implementations of Dijkstra Algorithm for Searching the Shortest Route," in *IC-MaGeStiC 2021*, vol. 96, pp. 76–84, 2022, doi: [10.2991/acsr.k.220202.016](https://doi.org/10.2991/acsr.k.220202.016).
- [13] M. Waruwu, "Pendekatan Penelitian Kualitatif: Konsep, Prosedur, Kelebihan dan Peran di Bidang Pen-

- didikan,” *Afeksi J. Penelit. dan Eval. Pendidik.*, vol. 5, no. 2, pp. 198–211, 2024, doi: [10.59698/afeksi.v5i2.236](https://doi.org/10.59698/afeksi.v5i2.236).
- [14] M. C. Bunaen, H. Pratiwi, and Y. F. Riti, “Penerapan Algoritma Dijkstra Untuk Menentukan Rute Terpendek,” *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 4, no. 1, pp. 213–223, 2022, doi: [10.47233/jteksis.v4i1.407](https://doi.org/10.47233/jteksis.v4i1.407).
- [15] S. Yuliani, M. R. Istambul, and E. A. Laksana, “Dijkstra’s Algorithm to Find Shortest Path of Tourist Destination in Bandung,” *Turkish J. Comput. Math. Educ.*, vol. 12, no. 8, pp. 1163–1168, 2021.
- [16] A. Lusiani et al., “Determination of the Fastest Path on Logistics Distribution by Using Dijkstra Algorithm,” in *ISSAT 2021*, pp. 246–250, 2021, doi: [10.2991/aer.k.211106.039](https://doi.org/10.2991/aer.k.211106.039).
- [17] L. Jabbar, E. Abass, and S. Hasan, “A Modification of Shortest Path Algorithm According to Adjustable Weights Based on Dijkstra Algorithm,” *Eng. Technol. J.*, vol. 41, no. 2, pp. 1–16, 2022, doi: [10.30684/etj.2022.136107.1296](https://doi.org/10.30684/etj.2022.136107.1296).
- [18] L. Y. Sipayung, C. R. Sinaga, and A. C. Sagala, “Application of Dijkstra’s Algorithm to Determine the Shortest Route,” *J. Comput. Networks Archit. High Perform. Comput.*, vol. 5, no. 2, pp. 648–655, 2023, doi: [10.47709/cnahpc.v5i2.2699](https://doi.org/10.47709/cnahpc.v5i2.2699).
- [19] Suardinata, R. Rusmi, and M. A. Lubis, “Determining Travel Time and Fastest Route Using Dijkstra Algorithm and Google Map,” *J. Sist. Inf.*, vol. 11, no. 1, pp. 496–505, 2022, doi: [10.32520/stmsi.v11i2.1836](https://doi.org/10.32520/stmsi.v11i2.1836).
- [20] A. Ardiansyah, A. M. Nasution, and M. Iqbal, “Comparative Analysis of Dijkstra and A\* Algorithms,” *bit-Tech*, vol. 8, no. 2, pp. 2974–2983, 2025, doi: [10.32877/bt.v8i2.3474](https://doi.org/10.32877/bt.v8i2.3474).
- [21] B. He, “Application of Dijkstra algorithm in finding the shortest path,” *J. Phys. Conf. Ser.*, vol. 2181, no. 1, 2022, doi: [10.1088/1742-6596/2181/1/012005](https://doi.org/10.1088/1742-6596/2181/1/012005).