

Implementasi Algoritma Dijkstra dalam Menentukan Lintasan Terpendek Rute Alun-Alun Majalengka menuju UIN Siber Syekh Nurjati Cirebon

Ade Lia Rahmaningrum¹, Muhamad Fahri Fauzan²

^{1,2} UIN Siber Syekh Nurjati Cirebon

*Korespondensi: Email: adeliarahmaningrum46@mail.uinssc.ac.id,
muhamadfahrifauzan@gmail.com

Abstrak

Mobilitas perjalanan dari Kabupaten Majalengka menuju Kota Cirebon, khususnya ke UIN Siber Syekh Nurjati Cirebon, menghadapi kompleksitas jaringan jalan dengan beragam alternatif rute yang berpotensi menimbulkan ketidakefisienan jarak tempuh apabila tidak dianalisis secara matematis. Penelitian ini bertujuan untuk menentukan lintasan terpendek dari Alun-Alun Majalengka menuju UIN Siber Syekh Nurjati Cirebon dengan menerapkan teori graf dan algoritma Dijkstra sebagai pendekatan penyelesaian masalah optimasi. Metode penelitian menggunakan pendekatan kuantitatif dengan desain studi kasus, di mana jaringan transportasi dimodelkan sebagai graf berbobot tak berarah yang terdiri atas simpul-simpul lokasi strategis dan sisi-sisi yang merepresentasikan ruas jalan dengan bobot jarak dalam satuan kilometer yang diperoleh dari Google Maps. Proses perhitungan jalur terpendek dilakukan secara komputasional melalui implementasi algoritma Dijkstra menggunakan bahasa pemrograman Python. Hasil penelitian menunjukkan bahwa lintasan terpendek diperoleh melalui rute Alun-Alun Majalengka – Cigasong – Rajagaluh – Cikalahang – Sumber – Jalan Salada – UIN Siber Syekh Nurjati Cirebon dengan total jarak minimum sebesar 41,4 km dibandingkan jalur alternatif lainnya. Temuan ini membuktikan bahwa algoritma Dijkstra mampu menghasilkan solusi optimal pada graf berbobot dengan asumsi bobot sisi bernilai non-negatif. Penelitian ini dibatasi pada penggunaan jarak sebagai parameter bobot graf dan belum mempertimbangkan faktor waktu tempuh maupun kondisi lalu lintas.

Kata kunci: *Algoritma Dijkstra, Teori graf, Lintasan Terpendek, Jaringan Transportasi.*

PENDAHULUAN

Konektivitas antarwilayah di Provinsi Jawa Barat, khususnya antara kabupaten Majalengka dan Kota Cirebon, memegang peranan vital dalam mendukung dinamika sosial, ekonomi, dan pendidikan. Majalengka, sebagai wilayah penyangga yang sedang berkembang pesat, memiliki mobilitas penduduk yang tinggi menuju Cirebon yang berfungsi sebagai pusat aktivitas regional. Salah satu faktor pendorong utama mobilitas ini adalah sektor pendidikan tinggi, dengan keberadaan Universitas Islam Negeri (UIN) Siber Syekh Nurjati Cirebon sebagai destinasi utama bagi para pelajar dan mahasiswa dari wilayah sekitarnya.

Dalam melakukan perjalanan dari pusat kota Majalengka (Alun-Alun) menuju kampus UIN Siber Syekh Nurjati(UIN SSC), pengguna jalan dihadapkan pada sebuah jaringan transportasi yang kompleks dengan berbagai alternatif rute. Secara umum, terdapat dua jalan utama yang sering digunakan yaitu jalur utara yang melintasi distrik Jatiwangi dan Palimanan, serta jalur selatan yang melintasi kecamatan Rajagaluh dan Sumber. Setiap rute memiliki karakteristik topografi dan jarak tempuh yang berbeda-beda, yang pada akhirnya mempengaruhi efisiensi perjalanan.

Permasalahan mendasar yang dihadapi oleh mahasiswa dan masyarakat umum adalah ketiadaan kepastian matematis dalam menentukan rute mana yang paling optimal. Seringkali, pemilihan rute hanya didasarkan pada kebiasaan (habit) atau intuisi subjektif yang tidak terukur. Padahal, selisih jarak tempuh sekecil apa pun akan berimplikasi langsung pada konsumsi bahan bakar, biaya operasional kendaraan, dan efisiensi waktu perjalanan. Oleh karena itu, diperlukan suatu pendekatan matematis yang mampu memodelkan permasalahan transportasi tersebut secara sistematis dan menghasilkan solusi optimal.

Matematika Diskrit menyediakan perangkat yang kuat untuk memodelkan persoalan ini melalui Teori Graf (Mamatha N dkk., 2024). Dalam perspektif teori graf, peta jalan raya dapat diabstraksikan sebagai sebuah graf terhubung di mana lokasi-lokasi strategis seperti persimpangan dan terminal direpresentasikan sebagai simpul, dan ruas jalan penghubung direpresentasikan sebagai sisi. Karena setiap ruas jalan memiliki jarak fisik tertentu, maka model yang digunakan termasuk ke dalam graf berbobot. Pendekatan ini memungkinkan permasalahan transportasi dianalisis secara kuantitatif dan terstruktur (Pham dkk., 2025).

Untuk menyelesaikan permasalahan pencarian lintasan terpendek pada graf berbobot dengan bobot non-negatif, algoritma Dijkstra merupakan salah satu metode yang paling banyak digunakan dan telah teruji keoptimalannya (Sunardi dkk., 2019; Zaki, 2017). Berbeda dengan algoritma *Breadth-First Search* yang hanya memperhitungkan jumlah langkah tanpa memperhatikan bobot, algoritma Dijkstra bekerja secara iteratif dengan prinsip *greedy* untuk menentukan jalur dengan total bobot minimum dari simpul asal ke simpul tujuan (Grujic & Grujic, 2025). Algoritma ini menjamin solusi yang optimal selama tidak terdapat sisi dengan bobot negatif, sebuah syarat yang terpenuhi dalam kasus jarak jalan raya.

Selain itu, penelitian fundamental menunjukkan bahwa struktur data heap baru dapat meningkatkan performa teoritis algoritma Dijkstra (Haeupler dkk., 2025). Penelitian ini memanfaatkan algoritma Dijkstra sebagai pendekatan dasar, sejalan dengan berbagai studi implementatif sebelumnya yang membahas penggunaan algoritma ini dalam sistem navigasi dan lingkungan kampus (Baharudin dkk., 2021; Manik dkk., 2024).

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk mengimplementasikan algoritma Dijkstra dalam menentukan lintasan terpendek dari alun-alun Majalengka menuju UIN Siber Syekh Nurjati Cirebon. Penelitian ini tidak hanya berfokus pada penentuan rute terpendek, tetapi juga pada pembuktian matematis efektivitas algoritma Dijkstra melalui pemodelan graf dan analisis jarak antar simpul. Hasil penelitian diharapkan dapat memberikan kontribusi praktis berupa rekomendasi rute yang efisien bagi masyarakat dan sivitas akademik, serta kontribusi teoritis dalam penerapan konsep matematika diskrit pada permasalahan transportasi nyata.

METODE

Pendekatan yang digunakan adalah pendekatan kuantitatif dengan desain studi kasus, dengan permasalahan transportasi nyata dimodelkan ke dalam bentuk matematis. Penelitian diawali melalui studi pustaka untuk mengkaji teori graf, struktur data graf berbobot, serta prinsip kerja algoritma Dijkstra sebagai landasan konseptual. Kajian pustaka mencakup teori graf dan struktur data graf berbobot, termasuk struktur data yang mendukung efisiensi analisis graf skala besar (Firmli & Chiadmi, 2023), serta perkembangan graf modern yang relevan dalam konteks optimasi jalur (Ding dkk., 2024). Tahap berikutnya adalah identifikasi titik-titik lokasi strategis yang direpresentasikan sebagai simpul graf serta pengukuran jarak fisik antar simpul menggunakan Google Maps sebagai sumber data geospasial. Jarak antar lokasi dinyatakan dalam satuan kilometer dan diasumsikan bernilai positif sesuai dengan syarat keberlakuan algoritma Dijkstra.

Permasalahan pencarian rute dimodelkan sebagai graf tak berarah berbobot yang secara formal dinyatakan sebagai $G = (V, E)$. Himpunan simpul V merepresentasikan lokasi-lokasi yang terlibat dalam penelitian, sedangkan himpunan sisi E merepresentasikan ruas jalan yang menghubungkan antar lokasi tersebut. Dalam penelitian ini, $V = \{v_1, v_2, \dots, v_8\}$ menyatakan lokasi dari alun-alun Majalengka sebagai simpul tujuan. Setiap sisi $e \in E$ diasosiasikan dengan fungsi bobot $w(e)$ yang memetakan sisi ke bilangan real positif. Nilai bobot tersebut menyatakan jarak tempuh antar simpul yang saling terhubung. Metode ini juga sesuai dengan formulasi lintasan terpendek yang telah dibahas pada kajian teoritis sebelumnya (Pradhana, t.t.).

Penentuan lintasan terpendek dilakukan menggunakan algoritma Dijkstra yang termasuk ke dalam algoritma *greedy* untuk menyelesaikan permasalahan *single-source shortest path* pada graf berbobot non-negatif. Proses algoritma diawali dengan analisis jarak simpul asal bernilai nol, sedangkan jarak simpul lainnya ditetapkan sebagai tak hingga. Pada setiap iterasi, algoritma memilih simpul dengan jarak sementara minimum dari himpunan simpul yang belum diproses, kemudian melakukan relaksasi terhadap simpul-simpul tetangganya untuk memperbarui nilai jarak terpendek. Proses iteratif tersebut berlanjut hingga simpul tujuan diproses dan jarak minimum ditetapkan. Penggunaan antrian prioritas dan komputasi graf ini mengikuti praktik pada penelitian optimasi jaringan (Narnavre, 2024).

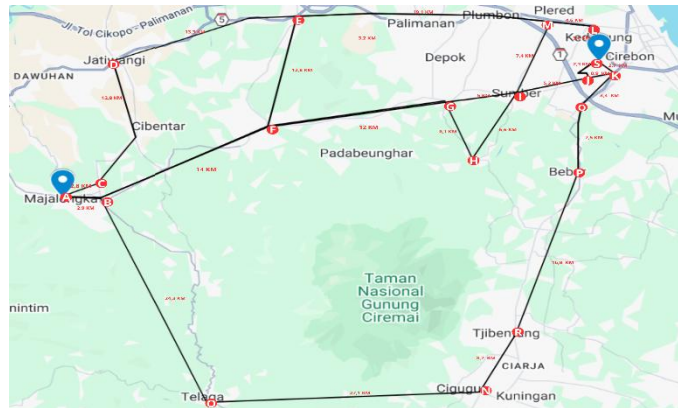
Implementasi Dijkstra dilakukan secara komputasional menggunakan bahasa pemrograman Python dengan memanfaatkan struktur data graf berbobot dan antrian prioritas untuk meningkatkan efisiensi perhitungan. Selain menghitung jarak minimum, algoritma mencatat simpul pendahulu pada setiap pembaruan jarak sehingga lintasan terpendek dapat direkonstruksi melalui proses pelacakan balik dari simpul tujuan menuju simpul asal. Prosedur ini menjamin bahwa solusi yang diperoleh merupakan solusi optimum global karena tidak terdapat lintasan lain dengan total bobot yang lebih kecil dibandingkan lintasan yang dihasilkan.

HASIL PENELITIAN DAN PEMBAHASAN

Penentuan rute penelitian ditentukan dengan mempertimbangkan jalur yang secara empiris paling sering digunakan oleh pengendara motor maupun mobil. Pemilihan rute ini bertujuan untuk memastikan bahwa model yang dibangun memiliki relevansi praktis dan merepresentasikan kondisi transportasi nyata di lapangan. Jaringan jalan yang dievaluasi dalam penelitian ini difokuskan pada rute arteri atau jalan nasional atau provinsi non-tol. Hal ini didasarkan pada pertimbangan untuk mengakomodasi mobilitas harian yang dapat diakses oleh seluruh jenis kendaraan, baik roda dua maupun roda empat. Dengan pendekatan ini, hasil analisis yang diperoleh diharapkan tidak hanya bersifat teoritis, tetapi juga aplikatif dalam konteks jaringan transportasi. Pemodelan jaringan jalan seperti ini konsisten dengan praktik implementasi Dijkstra dalam distribusi barang maupun jaringan transportasi digital (Baharudin dkk., 2021)

Setelah jalur utama ditetapkan, proses visualisasi dilakukan dengan menelusuri setiap rute dan menggambarkannya dalam bentuk garis lintasan pada peta digital. Tahapan ini bertujuan untuk memperoleh representasi jaringan transportasi dari Alun-alun Majalengka menuju UIN Siber Syekh Nurjati Cirebon, sehingga struktur rute dapat dianalisis secara sistematis. Visualisasi lintasan tersebut menjadi dasar konstruksi graf berbobot yang digunakan dalam penelitian ini.

Selanjutnya, setiap titik simpul direpresentasikan dengan huruf-huruf yang melambangkan lokasi berbeda di sepanjang rute. Untuk memperkuat validitas data, jarak antar simpul ditentukan menggunakan satuan kilometer (Km) yang diperoleh melalui Google Maps. Dengan demikian setiap sisi dalam graf memiliki bobot yang jelas dan terukur, sehingga dapat digunakan secara konsisten dalam penerapan algoritma pencarian jalur terpendek.



Gambar 1. Visualisasi rute Majalengka-Cirebon

Visualisasi hasil konstruksi simpul dan ruas jalan penelitian ditunjukkan pada Gambar 1. Gambar ini memperlihatkan jaringan rute dari Alun-alun Majalengka menuju UIN Siber Syekh Nurjati Cirebon, yang setiap simpulnya direpresentasikan dengan huruf-huruf dan setiap sisinya diberi bobot jarak dalam kilometer.

Untuk memperjelas identifikasi simpul, disusun tabel yang memuat daftar huruf beserta nama lokasi yang diwakilinya. Setiap huruf merepresentasikan simpul yang telah ditentukan pada tahap konstruksi graf, sehingga hubungan antara simbol graf dan lokasi nyata dapat ditampilkan secara sistematis. Selain itu karena simpul terhubung melalui ruas jalan dengan bobot berupa jarak tempuh antar simpul.

Tabel 1. Definisi Simpul (Lokasi)

Kode	Nama Lokasi	Kode	Nama Lokasi
A	Alun-alun Majalengka (Start)	K	Kalitanjung
B	Bunderan Cigasong	L	Kedawung
C	Bunderan Tonjong	M	Plered
D	Jatiwangi	N	Cigugur
E	Panjalín Kidul	O	Telaga
F	Rajagaluh	P	Beber
G	Singdangjawa	Q	Ciperna
H	Mandirancan	R	Cibentang
I	Sumber	S	UIN SSC (Tujuan Akhir)
J	Jl. Saladara		

Tabel 2. Bobot jarak antar simpul dalam jaringan rute
Majalengka–Cirebon

Vertex Sumber	Vertex Tujuan	Jarak/ Km	Vertex Sumber	Vertex Tujuan	Jarak/Km
A	B	2.9	I	M	7.4
A	C	2.8	J	K	0.9
B	F	14	J	S	2.3
B	O	24.3	K	Q	3.4
C	D	12.8	K	S	2.7
D	E	13.3	L	S	3.2
E	F	12.8	M	L	4.6
E	M	19.1	Q	P	7.5
F	G	12	P	R	16.8
G	H	6.1	R	N	8.7
G	I	5.0	N	O	27.1
H	I	6.6	A	B	2.8
I	J	5.2			

Pada tahap selanjutnya, penentuan jalur terpendek dilakukan dengan menerapkan algoritma Dijkstra yang diimplementasikan menggunakan bahasa pemrograman Python. Algoritma ini dirancang untuk menentukan jarak minimum dari satu simpul asal ke seluruh simpul lainnya dalam suatu graf berbobot, dengan asumsi bahwa seluruh bobot sisi bernilai non-negatif. Asumsi tersebut menjadi syarat utama agar algoritma Dijkstra dapat menjamin optimalisasi solusi yang dihasilkan, sebagaimana dijelaskan dalam teori graf pada Matematika Diskrit. Adapun ke-19 titik simpul yang ditentukan (Simpul A sampai S) dipilih karena merupakan persimpangan jalan strategis utama atau titik transit penting yang memungkinkan pengendara untuk beralih rute antar kecamatan di sepanjang jalur Majalengka menuju Cirebon

```
1 import networkx as nx
2 import matplotlib.pyplot as plt
3
4 G = nx.Graph()
5
6 pos = {
7     'A': (1, 6),      # Alun-alun
8     'B': (2, 5.5),    # Cigasong
9     'C': (2, 6.5),    # Tonjong
10    'D': (3, 9),       # Jatiwangi
11    'E': (6, 9.5),     # Panjalin
12    'M': (9, 9.2),     # Plered
13    'L': (10, 8.8),    # Kedawung
14    'F': (5, 7),       # Rajagaluh
15    'G': (7, 7.2),     # Cikalahang
16    'H': (7.5, 6),     # Mandirancan
17    'I': (8.5, 7.5),   # Sumber
18    'J': (9.5, 7.8),   #
19    'S': (10.5, 8.4),  #
20    'K': (11.5, 7.5),  #
21    'O': (4, 1),       # Telaga
22    'N': (8, 1.5),     # Kuningan
23    'R': (9, 3),       # Cibentang
24    'P': (10.5, 5),    # Beber
25    'Q': (11, 6.5)     # Ciperna
26 }
27
28 labels_map = {
29     'A': 'A\\nMajalengka', 'B': 'B\\nCigasong', 'C': 'C\\nBunderan Tonjong',
30     'D': 'D\\nJatiwangi', 'E': 'E\\nPanjalin', 'F': 'F\\nRajagaluh',
31     'G': 'G\\nSindangjawa', 'H': 'H\\nMandirancan', 'I': 'I\\nSumber',
32     'J': 'J\\nJl.Saladara', 'K': 'K\\nKalitanjung', 'L': 'L\\nKedawung',
33     'M': 'M\\nPlered', 'N': 'N\\nKuningan', 'O': 'O\\nTelaga',
34     'P': 'P', 'Q': 'Q', 'R': 'R',
35     'S': 'S\\nUIN (Tujuan)'
36 }
37
38 edges = [
39     ('A', 'B', 2.9), ('A', 'C', 2.8),
40     ('C', 'D', 12.8), ('D', 'E', 13.3), ('E', 'M', 19.1), ('M', 'L', 4.6),
41     ('B', 'F', 14.0), ('E', 'F', 12.8),
42     ('F', 'G', 12.0), ('G', 'H', 6.1),
43     ('H', 'I', 6.6), ('G', 'I', 5.0),
44     ('I', 'M', 7.4), ('I', 'J', 5.2),
45     ('J', 'S', 2.3), ('K', 'J', 0.9),
46     ('K', 'S', 2.7), ('L', 'S', 3.2),
47     ('K', 'Q', 3.4),
48     ('B', 'O', 24.3), ('O', 'N', 27.1), ('N', 'R', 8.7),
49     ('R', 'P', 16.8), ('P', 'Q', 7.5)
50 ]
51
52 for u, v, w in edges:
53     G.add_edge(u, v, weight=w)
```

Gambar 2. Potongan Kode Pemodelan Graf Jaringan Jalan Berbobot

Potongan kode pada Gambar 2 memperlihatkan tahap pada pemodelan jaringan jalan ke dalam bentuk graf berbobot menggunakan Pustaka NetworkX. Graf dibangun sebagai graf tak berarah, dengan setiap simpul merepresentasikan Lokasi atau titik penting pada jaringan jalan. Struktur data *pos* digunakan untuk menentukan koordinat masing-masing simpul yang berfungsi sebagai tata letak visualisasi graf, sedangkan *labels_map* berperan dalam pemberian label nama Lokasi pada setiap simpul agar mudah diidentifikasi.

Hubungan antar lokasi direpresentasikan melalui himpunan sisi yang disimpan dalam struktur *edges*, dengan setiap sisi dilengkapi bobot yang menyatakan jarak tempuh antar Lokasi dalam satuan kilometer. Proses penambahan sisi ke dalam graf dilakukan secara iteratif sehingga seluruh jaringan jalan dapat dimodelkan secara terstruktur. Pemodelan graf berbobot ini menjadi dasar dalam penerapan algoritma Dijkstra pada tahap selanjutnya untuk menentukan rute terpendek dari titik awal menuju titik tujuan.


```
55 source = 'A'
56 target = 'S'
57
58 try:
59     path = nx.dijkstra_path(G, source, target, weight='weight')
60     dist = nx.dijkstra_path_length(G, source, target, weight='weight')
61     print(f"Rute Terpendek: {path}")
62     print(f"Total Jarak: {dist} Km")
63 except:
64     print("Jalur tidak ditemukan")
65
```

Gambar 3. Potongan Kode Implementasi Algoritma Dijkstra

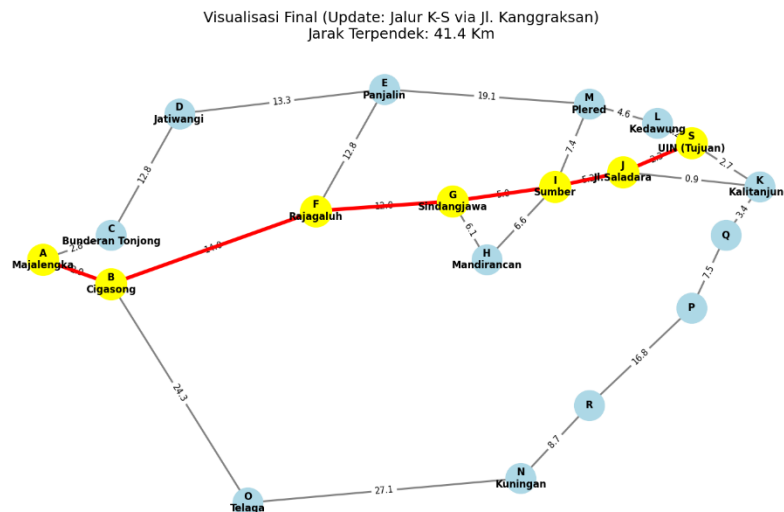
Potongan kode pada Gambar 3 menunjukkan implementasi algoritma Dijkstra untuk menentukan lintasan terpendek pada graf berbobot yang telah dibentuk sebelumnya. Simpul *source* ditetapkan sebagai titik awal dan simpul *target* sebagai titik tujuan pencarian rute. Fungsi *nx.dijkstra_path* digunakan untuk memperoleh urutan simpul yang membentuk lintasan terpendek, sedangkan *nx.dijkstra_path_length* digunakan untuk menghitung total jarak lintasan berdasarkan bobot sisi.

Struktur *try-except* diterapkan untuk menangani kondisi apabila lintasan antara titik awal dan titik tujuan tidak tersedia dalam graf. Hasil perhitungan berupa rute terpendek dan total jarak selanjutnya digunakan sebagai dasar analisis dan divisualisasikan pada tahap berikutnya.

```
[Running] python -u "d:\HOBBY\Datapeta.py"
Rute Terpendek: ['A', 'B', 'F', 'G', 'I', 'J', 'S']
Total Jarak: 41.4 Km
```

Gambar 4. Hasil Keluaran Program Penentuan Rute Terpendek
Menggunakan Algoritma Dijkstra

Gambar 4 menunjukkan hasil keluaran program setelah penerapan algoritma Dijkstra pada graf berbobot. Berdasarkan hasil tersebut, diperoleh lintasan terpendek dari simpul A menuju simpul S dengan urutan simpul A – B – F – G – I – J – S. Total jarak lintasan terpendek yang dihasilkan adalah sebesar 41,4 km. Hal ini sejalan dengan karakteristik algoritma Dijkstra yang terbukti optimal dalam kondisi bobot non-negatif (Zaki, 2017), dan selanjutnya divisualisasikan dalam bentuk graf untuk memperjelas jalur yang diperoleh.



Gambar 5. Visualisasi hasil penentuan lintasan terpendek menggunakan algoritma Dijkstra

Hasil jalur terpendek yang diperoleh dari eksekusi program ditampilkan pada Gambar 5, diperoleh lintasan optimal yang menghubungkan Alun-alun Majalengka sebagai simpul asal hingga UIN Siber Syekh Nurjati Cirebon sebagai simpul tujuan akhir. Berdasarkan hasil komputasi algoritma Dijkstra, rute paling efisien memiliki total akumulasi jarak sebesar 41,4 km dan terbentuk dari rangkaian simpul $A \rightarrow B \rightarrow F \rightarrow G \rightarrow I \rightarrow J \rightarrow S$, yang secara geografis merepresentasikan lintasan Alun-alun Majalengka – Cigasong – Rajagaluh – Sindangjawa – Sumber – Jalan Saladara – UIN SSC. Rincian segmen jalan yang dilalui meliputi Alun-alun Majalengka – Cigasong sejauh 2.9 km, Cigasong – Rajagaluh sejauh 14 km, Rajagaluh – Sindangjawa, Sindangjawa – Sumber sejauh 5 km, Sumber – Jalan Saladara sejauh 5.2 km, Jalan Saladara – UIN SSC sejauh 2.3 km. Lintasan ini dipilih karena memiliki akumulasi bobot jarak paling kecil dibandingkan dengan jalur alternatif lain yang terdapat dalam jaringan graf.

Hasil perhitungan menggunakan algoritma Dijkstra kemudian dibandingkan dengan rute default yang direkomendasikan oleh Google Maps. Hasil perbandingan menunjukkan bahwa lintasan yang diperoleh melalui algoritma Dijkstra sama dengan rute yang direkomendasikan Google maps, yaitu melalui Alun-alun Majalengka – Cigasong – Rajagaluh – Sindangjawa – sumber – Jalan Saladara – UIN Siber Syekh Nurjati Cirebon. Perbedaan hanya terdapat pada estimasi jarak, yaitu 41,4 km berdasarkan hasil komputasi dan sekitar 41 km pada Google Maps, yang disebabkan oleh perbedaan metode pengukuran dan pembulatan jarak.

Berdasarkan jaringan graf yang dibangun, masih terdapat jalur alternatif yang dapat menghubungkan titik asal dan tujuan, salah satunya melalui jatiwangi dengan lintasan A – C – D – E – M – L – S. Jalur tersebut memiliki total jarak 55,8 km, lebih besar dibandingkan lintasan hasil algoritma Dijkstra yang memiliki total jarak 41,4 km. Perbandingan ini menunjukkan bahwa algoritma Dijkstra memilih lintasan dengan akumulasi bobot minimum diantara alternatif rute yang tersedia.

Pemilihan jalur tersebut menunjukkan bahwa algoritma Dijkstra secara konsisten mengutamakan efisiensi jarak total pada setiap iterasi perhitungan. Meskipun terdapat rute lain yang dapat menghubungkan titik asal dan tujuan, kombinasi ruas jalan pada jalur terpilih menghasilkan bobot kumulatif yang lebih rendah. Hal ini menegaskan bahwa lintasan terpendek dalam graf berbobot tidak ditentukan oleh jumlah simpul yang dilewati, melainkan total bobot sisi yang membentuk jalur tersebut.

Secara konseptual, lintasan Alun-alun Majalengka – Cigasong – Rajagaluh – Sindangjawa – Sumber – Jalan Saladara – UIN SSC telah memenuhi prinsip lintasan terpendek dalam teori graf, dimana setiap simpul antara dipilih melalui mekanisme seleksi *greedy* dan relaksasi jarak. Dengan demikian, hasil ini menunjukkan bahwa penerapan algoritma Dijkstra pada jaringan transportasi yang dimodelkan mampu menghasilkan rute perjalanan yang efisien dan relevan dengan kondisi nyata di lapangan.

KESIMPULAN DAN SARAN

Penelitian ini menunjukkan bahwa algoritma Dijkstra dapat diterapkan secara efektif untuk menentukan lintasan terpendek dari Alun-alun Majalengka menuju UIN Siber Syekh Nurjati Cirebon. Dengan memodelkan jaringan jalan sebagai graf tak berarah berbobot, diperoleh rute optimal melalui Alun-alun Majalengka – Cigasong – Rajagaluh – Sindangjawa – Sumber – Jalan Saladara – UIN Siber Syekh Nurjati Cirebon dengan total jarak minimum sebesar 41.4 km, yang lebih efisien dibandingkan jalur alternatif lainnya.

Hasil tersebut membuktikan bahwa algoritma Dijkstra mampu menghasilkan solusi optimal pada graf berbobot non-negatif melalui mekanisme seleksi *greedy* dan relaksasi jarak. Penelitian ini masih terbatas pada penggunaan jarak sebagai satu-satunya parameter bobot graf dan belum mempertimbangkan faktor waktu tempuh, kondisi lalu lintas, maupun kepadatan kendaraan. Oleh karena itu, penelitian selanjutnya dapat dikembangkan dengan menambahkan variabel-variabel tersebut atau mengintegrasikan algoritma lain untuk perbandingan performa. Meskipun demikian, penelitian ini memberikan kontribusi akademik dalam penerapan konsep matematika diskrit dan algoritma graf, serta memiliki nilai praktis sebagai dasar pengembangan sistem penentuan rute berbasis optimasi graf

bagi masyarakat dan sivitas akademik.

REFERENSI

- Baharudin, I., Purwanto, A. J., Budiman, T. R., & Fauzi, M. (2021). Implementasi Algoritma Dijkstra Untuk Menentukan Jalur Terpendek Dalam Distribusi Barang. *Jurnal Lebesgue: Jurnal Ilmiah Pendidikan Matematika, Matematika Dan Statistika*, 2(2), 194–203. <https://doi.org/10.46306/Lb.V2i2.74>
- Ding, K., Liu, Y., Zhang, C., & Wang, J. (2024). *Data-Efficient Graph Learning: Problems, Progress, And Prospects*. *Ai Magazine*, 45(4), 549–560. <https://doi.org/10.1002/Aai.12200>
- Firmli, S., & Chiadmi, D. (2023). *A Scalable Data Structure For Efficient Graph Analytics And In-Place Mutations*. *Data*, 8(11), 166. <https://doi.org/10.3390/Data8110166>
- Grujic, Z., & Grujic, B. (2025). Optimal Routing In Urban Road Networks: A Graph-Based Approach Using Dijkstra's Algorithm. *Applied Sciences*, 15(8), 4162. <https://doi.org/10.3390/App15084162>
- Haeupler, B., Hladík, R., Rozhoň, V., Tarjan, R. E., & Tětek, J. (2025). *Universal Optimality Of Dijkstra Via Beyond-Worst-Case Heaps* (No. Arxiv:2311.11793). Arxiv. <https://doi.org/10.48550/Arxiv.2311.11793>
- Mamatha N, Sunitha S.S, & Shivakumar M D. (2024). Graph Theory And Its Role In Social Network Analysis. *World Journal Of Advanced Research And Reviews*, 21(2), 2088–2093. <https://doi.org/10.30574/Wjarr.2024.21.2.0249>
- Manik, A. R., Purba, J., Akbar, M. B., & Harliana, P. (2024). Implementasi Algoritma Dijkstra Dalam Pencarian Rute Terpendek Antar Fakultas Di Universitas Negeri Medan. *Journal Of Education Transportation And Business*, 1(2), 466–478. <https://doi.org/10.57235/Jetbus.V1i2.3905>
- Narnavre, S. (2024). Optimized Implementation Of Dijkstra's Algorithm For Efficient Shortest Path Finding In Networked Systems. *International Journal For Research In Applied Science And Engineering Technology*, 12(11), 1705–1712. <https://doi.org/10.22214/Ijrasat.2024.65444>
- 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. <https://doi.org/10.3389/Ffut.2025.1524232>
- Pradhana, B. A. (T.T.). *Studi Dan Implementasi Persoalan Lintasan Terpendek Suatu Graf Dengan Algoritma Dijkstra Dan Algoritma Bellman-Ford*.

- Sunardi, S., Yudhana, A., & Kadim, A. A. (2019). Implementasi Algoritma Dijkstra Untuk Analisis Rute Transportasi Umum Transjogja Berbasis Android. *Jurnal Sistem Informasi Bisnis*, 9(1), 32. <https://doi.org/10.21456/Vol9iss1pp32-38>
- Zaki, A. (2017). Algoritma Dijkstra: Teori Dan Aplikasinya. *Jurnal Matematika Unand*, 6(4), 1–8. <https://doi.org/10.25077/jmu.6.4.1-8.2017>