

Analisis Komparasi Lintasan Terpendek dan Waktu Tempuh Minimum Rute Kuningan-Cirebon Menggunakan Representasi Graf Berbobot

Dara Maywanti¹, Aban Subanul Fahmi²

^{1,2} Fakultas Ilmu Tarbiyah dan Keguruan, UIN Siber Syekh Nurjati, Cirebon, Indonesia

Article Info

Article history:

Received Januari 1, 2026

Revised Januari 4, 2026

Accepted Januari 10, 2026

Kata Kunci:

Algoritma Dijkstra,
Teori Graf,
Lintasan Terpendek,
Optimasi Rute,
Kuningan-Cirebon

Keywords:

*Dijkstra Algorithm,
Graph Theory,
Shortest Path,
Route Optimization,
Kuningan-Cirebon*

ABSTRAK

Mobilitas mahasiswa pelaju (commuter) dari wilayah Kabupaten Kuningan menuju pusat pendidikan di Kota Cirebon dihadapkan pada dilema pemilihan rute transportasi yang efisien. Terdapat dua opsi jalur utama yang umum digunakan, yaitu jalur via Gronggong dan jalur via Plangon, yang memiliki karakteristik geometris dan kondisi lalu lintas yang berbeda. Penelitian ini bertujuan untuk memodelkan jaringan jalan tersebut kedalam struktur matematika dan menentukan rute paling optimal berdasarkan perbandingan dua variabel bobot: jarak tempuh minimum (km) dan waktu tempuh tercepat (menit). Metode penelitian yang digunakan adalah observasi lapangan (field observation) untuk menghimpun data bobot sisi, yang kemudian di representasikan sebagai Graf Berarah Berbobot (Directed Weighted Graph) dan diselesaikan menggunakan Algoritma Dijkstra. Hasil perhitungan menunjukkan bahwa rute via Gronggong yang dikombinasikan dengan jalan pintas Talun (Jembatan Merah) mendominasi efisiensi pada kedua parameter. Rute ini menghasilkan total jarak terpendek selama 62 menit. Sebaliknya, rute alternatif via Planggon terbukti kurang efisien dengan total jarak tempuh mencapai 50,7 km dan waktu tempuh 74 menit, akibat inefisiensi jarak pada segmen Mandirancan. Selain perhitungan manual, penelitian ini juga melakukan validasi menggunakan simulasi komputasi bahasa pemrograman Python, yang menghasilkan nilai luaran (output) yang identik. Hal ini membuktikan bahwa penggunaan jalan pintas Talun adalah solusi optimal (Pareto Optimal) bagi pengendara sepeda motor untuk memangkas jarak dan menghindari kemacetan kota.

ABSTRACT

The mobility of commuter students from Kuningan Regency to educational centers in Cirebon City is faced with the dilemma of choosing an efficient transportation route. There are two main route options used, namely the route via Gronggong and the route via Plangon, which have different geometric characteristics and traffic conditions. This study aims to model the road network into a mathematical structure and determine the most optimal route based on a comparison of two weight variables: minimum travel distance (km) and fastest travel time (minutes). The research method used is field observation to collect edge weight data, which is then represented as a Directed Weighted Graph and solved using the Dijkstra Algorithm. The calculation results show that the route via Gronggong combined with the Talun shortcut (Red Bridge) dominates efficiency in both parameters. This route produces the shortest total distance of 62 minutes. In contrast, the alternative route via Planggon is proven to be less efficient with a total distance of 50.7 km and a travel time of 74 minutes, due to distance inefficiency in the Mandirancan segment. In addition to manual calculations, this study also validated using computational simulations in the Python

programming language, which produced identical output values. This proves that using the Talun shortcut is a Pareto-optimal solution for motorcyclists to shorten distances and avoid city traffic jams.

This is an open access article under the [CC BY](#) license.



Corresponding Author:

Dara Maywanti
Fakultas Ilmu Tarbiyah dan Keguruan, UIN Siber Syekh Nurjati,
Cirebon, Indonesia
Email: daramaywanti123@gmail.com

1. PENDAHULUAN

Mobilitas masyarakat antar daerah merupakan hal penting dalam mendukung aktivitas pendidikan dan ekonomi, terutama dijalur penghubung antara Kabupaten Kuningan dan Kota Cirebon. Bagi mahasiswa yang tinggal diwilayah Kuningan seperti Desa Cihideung Hilir, dan menempuh pendidikan dikampus UIN Siber Syekh Nurjati Cirebon, efisiensi dalam perjalanan menjadi hal utama. Memilih rute perjalanan tidak hanya melihat jarak terdekat, tetapi juga mempertimbangkan estimasi waktu tempuh yang dipengaruhi oleh kondisi jalan. Dalam praktiknya, pengguna jalan menghadapi sebagai pilihan rute dengan karakteristik berbeda, yaitu jalur utama melalui Jalan Baru-Gronggong dan jalur alternatif melalui Mandicaran-Plangon.

Masalah muncul ketika pengguna jalan harus memilih rute yang paling efisien. Jalur utama melalui Gronggong secara geometris lebih pendek, tetapi sering mengalami hambatan berupa kemacetan karena aktivitas proyek dan volume kendaraan yang padat, serta kondisi jalan yang curam. Sebaliknya, jalur alternatif melalui Plagon-Sumber memiliki jarak lebih panjang namun arus lalu lintas lebih lancar. Terkadang, keputusan memilih rute hanya didasarkan pada perasaan subjektif tanpa bantuan perhitungan matematis yang tepat. Hal ini sesuai dengan penelitian Gede dan Santiari [1], yang menyatakan bahwa jarak terpendek belum tentu menjamin waktu tempuh tercepat karena adanya hambatan lalu lintas yang tidak pasti. Senada dengan hal tersebut, Ratnasari [2] juga menekankan pentingnya melakukan estimasi waktu tempuh sebagai variabel pembanding agar hasil penentuan rute menjadi lebih akurat. Untuk mengatasi ketidakpastian ini, dibutuhkan pendekatan pemodelan matematika yang dapat merepresentasikan jaringan jalan dalam bentuk struktur yang bisa dianalisis secara kuantitatif.

Teori Graf merupakan cabang matematika yang cocok untuk menyelesaikan permasalahan optimasi ini. Dimana persimpangan jalan diperbandingkan sebagai himpunan simpul dan ruas jalan sebagai sisi dengan bobot tertentu. Salah satu algoritma pencarian lintasan yang paling efisien dan umum digunakan untuk graf berbobot positif adalah Algoritma Dijkstra. Penelitian Awalloedin [3] menunjukkan bahwa Algoritma Dijkstra sangat akurat dalam menentukan rute optimal dengan memproses bobot setiap sisi secara sistematis. Berdasarkan penjelasan diatas, penelitian ini bertujuan untuk menerapkan Algoritma Dijkstra dalam melakukan analisis perbandingan antara jalur terpendek berdasarkan jarak (km) dan jalur tercepat berdasarkan waktu (menit) pada rute Cihideung Hilir ke Kampus UIN Siber Syekh Nurjati. Penelitian ini akan membandingkan efisiensi antara rute utama dan rute alternatif dengan menggunakan data primer dari hasil observasi lapangan dengan moda sepeda motor. Hasil analisis diharapkan dapat memberikan rekomendasi rute yang valid secara matematis, sehingga bisa menjadi referensi bagi mahasiswa maupun masyarakat umum dalam mengefisiensikan waktu perjalanan sehari-hari.

2. METODE

2.1 Pendekatan penelitian

Penelitian ini menggunakan pendekatan kuantitatif terapan (applied mathematics). Masalah pencarian rute transportasi dimodelkan secara matematis menggunakan struktur data Graf, kemudian diselesaikan menggunakan metode komputasi deterministik. Pendekatan ini dipilih karena mampu memberikan solusi pasti (eksak) dalam menentukan nilai minimum dari himpunan rute yang tersedia [4].

2.2 Sumber dan Teknik Pengumpulan Data

Data digunakan dalam penelitian ini di klasifikasikan sebagai data primer. Teknik pengumpulan data dilakukan melalui metode observasi langsung (field observation) dengan cara menelusuri rute secara nyata menggunakan moda transportasi sepeda motor. Pengambilan data dilakukan dengan mencatat dua variabel utama pada setiap ruas jalan:

1. Jarak Geometris (s) : Diukur menggunakan instrumen odometer digital pada kendaraan dalam satuan kilometer (km).
2. Waktu Tempuh (t) : Diukur menggunakan alat pencatat waktu (stopwatch) dalam satuan menit.

Pengukuran dilakukan pada kondisi lalu lintas normal untuk menghindari bias data akibat cuaca ekstrem atau kecelakaan tak terduga.

2.3 Pemodelan Matematika (Representasi Graf)

Sistem jaringan jalan dari Desa Cihideung Hilir menuju Kampus UIN Siber Syekh Nurjati direpresentasikan sebagai Graf Berarah Berbobot (Directed Weighted Graph), yang dinotasikan dengan $G = (V, E)$ [5].

2.3.1 Definisi Simpul (Vertex)

Simpul (V) merepresentasikan titik-titik lokasi penting, seperti persimpangan jalan, titik percabangan, atau landmark wilayah. Berdasarkan rute observasi, ditetapkan himpunan simpul $V = \{v_1, v_2, \dots, v_{11}\}$ sebagai berikut:

- v_1 : Desa Cihideung Hilir (Titik Awal)
- v_2 : Ciawigebang
- v_3 : Pertigaan Jalan Baru (Cipicung)
- v_4 : Taman Caracas (Titik Percabangan Utama)
- v_5 : Kawasan Gronggong (Puncak Jalur Utama)
- v_6 : Talun (via Jembatan Merah)
- v_7 : Pertigaan Mandirancan
- v_8 : Wisata Plangon
- v_9 : Sumber (Pusat Kota Kabupaten)
- v_{10} : Majasem (Titik Temu)
- v_{11} : Kampus UIN Siber Syekh Nurjati (Titik Tujuan)

2.3.2 Definisi Sisi (Edge) dan Bobot (weight)

Sisi (E) merepresentasikan ruas jalan yang menghubungkan dua simpul. Karena penelitian ini membandingkan dua parameter efisiensi, maka setiap sisi e_{ij} (jalan dari simpul i ke j) memiliki dua jenis bobot, yaitu:

1. Bobot Jarak (W_d)
2. Bobot Waktu (W_t)

Fungsi bobot dinotasikan sebagai:

$$W(e_{ij}) = \{d_{ij}, t_{ij}\}$$

Penelitian ini akan melakukan dua kali proses perhitungan terpisah untuk meminimalkan $\sum d$ (jarak) dan $\sum t$ (waktu) [3].

2.4 Prosedur Algoritma Dijkstra

Untuk menyelesaikan permasalahan lintasan terpendek (*shortest path problem*), digunakan Algoritma Dijkstra. Algoritma ini bekerja dengan prinsip *Greedy*, yaitu selalu memilih simpul dengan bobot terkecil yang belum dikunjungi pada setiap langkahnya [6].

Langkah-langkah prosedural algoritma adalah sebagai berikut:

1. Inisialisasi (Initialization): Tetapkan nilai jarak ke titik awal (V_1) sama dengan 0 ($d[v_1] = 0$), dan jarak ke semua titik lain sama dengan tak hingga ($d[v] = \infty$).
2. Seleksi (Selection): Pilih simpul dengan estimasi bobot terkecil dari himpunan simpul yang belum dikunjungi. Sebut simpul ini sebagai u .
3. Relaksasi (Relaxation): Periksa semua tetangga v dari simpul u . Hitung total bobot dari titik awal melalui u . Jika bobot baru ini lebih kecil daripada bobot yang tersimpan sebelumnya di v , maka perbarui nilai v . Rumus Relaksasi:

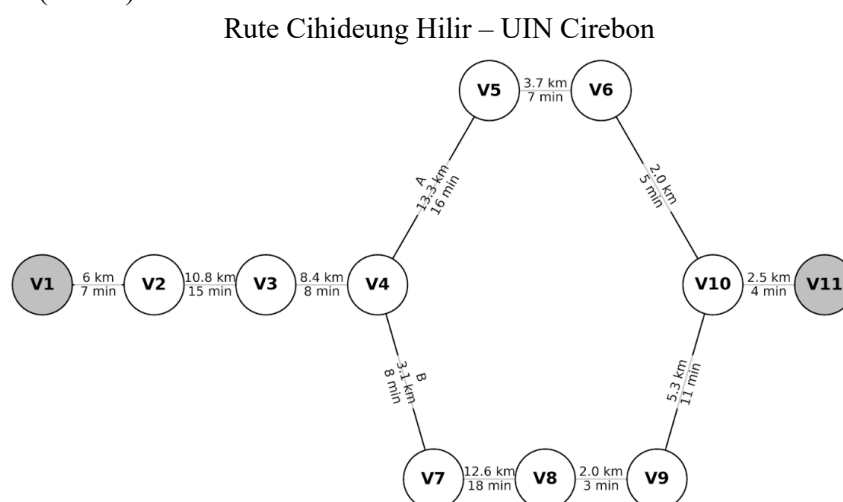
$$if (d[u] + w(u, v) < d[v]) then d[v] = d[u] + w(u, v)$$
4. Penandaan (Marking): Tandai simpul u sebagai "sudah dikunjungi" (visited). Simpul ini tidak akan dicek lagi.
5. Iterasi (Looping): Ulangi langkah 2 sampai 4 hingga simpul tujuan (v_{11}) tercapai.

3. HASIL DAN PEMBAHASAN

3.1 Visualisasi Model Graf

Langkah pertama dalam analisis ini adalah memodelkan rute perjalanan nyata dari Desa Cihideung Hilir menuju Kampus UIN Siber Syekh Nurjati menjadi sebuah model matematika yang disebut graf. Pendekatan ini memungkinkan struktur jalan yang kompleks disederhanakan menjadi simpul dan sisi agar dapat dihitung secara komputasi [7].

Gambar 1 dibawah ini adalah visualisasi graf berarah (*directed graph*) yang dihasilkan menggunakan simulasi komputer. Lingkaran merepresentasikan lokasi (Simpul V) dan garis panah merepresentasikan jalan (Sisi E). Kotak kecil pada setiap garis menunjukkan bobot ganda, yaitu Jarak (km) dan Waktu (menit).



Gambar 1. Visualisasi Graf Berarah Berbobot

Penjelasan Gambar:

1. Titik Awal (V_1) : Desa Cihideung Hilir
2. Titik Percabangan (V_4) : Taman Caracas. Disini graf terpecah menjadi dua Jalur (Atas dan Bawah)
 - Jalur A (Atas) : Melalui V_5 (Gronggong) $\rightarrow$ V_6 (Talun / Jembatan)
 - Jalur B (Bawah) : Melalui V_7 (Mandirancan) $\rightarrow$ V_8 (Plangon) $\rightarrow$ V_9 (Sumber).
3. Titik Temu (V_{10}) : Majasem
4. Titik Akhir (V_{11}) : Kampus UIN Cirebon

3.2 Data Bobot Sisi (Tabel Observasi)

Data yang ditampilkan pada graf diatas berasal dari hasil observasi lapangan menggunakan odometer motor dan stopwatch. Penggunaan data primer dari lapangan sangat penting untuk meminimalkan bias dan meningkatkan akurasi model graf terhadap kondisi nyata [8]. Rincian lengkap data tersebut disajikan dalam Tabel 1 berikut. Tabel ini menjadi acuan utama (input) untuk perhitungan algoritma.

Tabel 1. Data Bobot Sisi (Jarak dan Waktu)

Kode Sisi	Rute (Dari Simpul $\rightarrow$ Ke Simpul)	Jarak (km)	Waktu (menit)	Keterangan Kondisi Jalan
SEGMENT AWAL	(Jalur Tunggal)			
e_1	V_1 (Cihideung) $\rightarrow$ V_2 (Ciawigebang)	6,0	7	Jalan Desa
e_2	V_2 (Ciawigebang) $\rightarrow$ V_3 (Jalan Baru)	10,8	15	Jalan Kabupaten
e_3	V_3 (Jalan Baru) $\rightarrow$ V_4 (Caracas)	8,4	8	Jalan Provinsi
OPSI A	(VIA GRONGGONG)			
e_4	V_4 (Caracas) $\rightarrow$ V_5 (Gronggong)	13,3	16	Menanjak/Padat
e_5	V_5 (Gronggong) $\rightarrow$ V_6 (Talun)	3,7	7	Turunan/Jembatan Merah
e_6	V_6 (Talun) $\rightarrow$ V_{10} (Majasem)	2,0	5	Jalan Tikus/Pintas
OPSI B	(VIA PANGON)			
e_7	V_4 (Caracas) $\rightarrow$ V_7 (Mandirancan)	3,1	8	Aspal Lancar
e_8	V_7 (Mandirancan) $\rightarrow$ V_8 (Plangon)	12,6	18	Rute Panjang
e_9	V_8 (Plangon) $\rightarrow$ V_9 (Sumber)	2,0	3	Kawasan Hutan
e_{10}	V_9 (Sumber) $\rightarrow$ V_{10} (Majasem)	5,3	11	Pusat Kota
SEGMENT AKHIR	(Menuju Kampus)			
e_{11}	V_{10} (Majasem) $\rightarrow$ V_{11} (Kampus UIN)	2,5	4	Jalan Bypass

3.3 Analisis Perhitungan Manual (Algoritma Dijkstra)

Pada bagian ini, kita akan menguraikan bagaimana Algoritma Dijkstra bekerja langkah demi langkah untuk menentukan rute terbaik berdasarkan data pada tabel 3.2. Metode ini bekerja dengan menelusuri simpul-simpul yang memiliki akumulasi bobot terkecil secara berurutan [9].

3.3.1 Perhitungan Jarak Terpendek (*km*)

Algoritma mencari rute dengan total kilometer paling sedikit.

1. Tahap awal: Dari Cihideung (V_1) sampai ke percabangan Caracas (V_4), jarak dijumlahkan langsung:

$$6,0+10,8+8,4 = 25,2 \text{ km}$$

2. Tahap Percabangan (Kompetisi): Di Caracas (V_4), terdapat dua opsi menuju Majasem (V_{10}):

Jalur Gronggong: $25,2 + 13,3 + 3,7 + 2,0 = 44,2 \text{ km}$

Jalur Plangon: $25,2 + 3,1 + 12,6 + 2,0 + 5,3 = 48,2 \text{ km}$

3. Keputusan: Karena $44,2 < 48,2$, algoritma memilih Jalur Gronggong.

4. Tahap akhir: Ditambah jarak ke kampus (2,5 km)

Total Jarak Minimum = 46,7 km

3.3.2 Perhitungan Waktu Tercepat (*menit*)

Algoritma mencari rute dengan durasi perjalanan paling singkat. Analisis waktu seringkali menghasilkan rute yang berbeda dengan analisis jarak karena adanya faktor hambatan lalu lintas yang mempengaruhi kecepatan rata-rata kendaraan [10].

1. Tahap Awal: Waktu tempuh dari Cihideung ke Caracas:

$$7 + 15 + 8 = 30 \text{ menit}$$

2. Tahap Percabangan (kompetisi):

Jalur Gronggong: $30 + 16 + 7 + 5 = 58 \text{ menit}$. (*Meskipun Gronggong macet/nanjak, jalan tikus Talun sangat cepat*).

Jalur Plangon: $30 + 8 + 18 + 3 + 11 = 70 \text{ menit}$. (*Segmen Mandirancan-Plangon memakan waktu 18 menit karena jaraknya jauh*).

3. Keputusan: karena $58 < 70$, algoritma memilih Jalur Gronggong.

4. Tahap Akhir: Ditambah waktu ke kampus (4 menit).

Total Waktu Minimum = 62 menit.

3.4 Validasi Komputasi (Hasil Python)

Untuk memastikan keakuratan perhitungan manual di atas, penelitian ini melakukan validasi menggunakan program komputer berbasis bahasa Python. Validasi berbasis kode pemrograman dinilai efektif untuk menguji konsistensi logika algoritma pada dataset yang kompleks[11]. Program ini memproses struktur data graf dan menjalankan fungsi `dijkstra_shortest_path`.

1. Konfigurasi Input Program

Berikut adalah representasi data Tabel 1 ke dalam struktur data *Dictionary* pada Python yang digunakan sebagai input:

```
1 import heapq
2
3 # 1. Representasi Graf (Data dari Observasi Anda)
4 # Format: { 'Node_Asal': { 'Node_Tujuan': { 'jarak': km, 'waktu': menit, ... } } }
5 graph = {
6     'V1': { 'V2': { 'jarak': 6.0, 'waktu': 7 } },
7     'V2': { 'V3': { 'jarak': 10.8, 'waktu': 15 } },
8     'V3': { 'V4': { 'jarak': 8.4, 'waktu': 8 } },
9
10    # Percabangan di V4 (Caracas)
11    'V4': {
12        'V5': { 'jarak': 13.3, 'waktu': 16 }, # ke Gronggong
13        'V7': { 'jarak': 3.1, 'waktu': 8 }    # ke Mandirancan
14    },
15
16    # Jalur Atas (Gronggong - Talun)
17    'V5': { 'V6': { 'jarak': 3.7, 'waktu': 7 } },
18    'V6': { 'V10': { 'jarak': 2.0, 'waktu': 5 } },
19
20    # Jalur Bawah (Piangan - Sumber)
21    'V7': { 'V8': { 'jarak': 12.6, 'waktu': 18 } },
22    'V8': { 'V9': { 'jarak': 2.0, 'waktu': 3 } },
23    'V9': { 'V10': { 'jarak': 5.3, 'waktu': 11 } },
24
25    # Titik Temu ke Finish
26    'V10': { 'V11': { 'jarak': 2.5, 'waktu': 4 } },
27    'V11': {} # Finish
28 }
29
30 # 2. Fungsi Algoritma Dijkstra
31 def dijkstra_solver(graph, start, end, weight_type):
32     # Inisialisasi jarak infinity
33     distances = {node: float('inf') for node in graph}
34     distances[start] = 0
35     pq = [(0, start)] # Priority Queue
36     predecessors = {node: None for node in graph} # untuk melacak rute
37
38     while pq:
39         current_weight, current_node = heapq.heappop(pq)
40
41         if current_weight > distances[current_node]:
42             continue
43
44         if current_node == end:
45             break
46
47     # Cek Tetangga
48     if current_node in graph:
49         for neighbor, weights in graph[current_node].items():
50             weight = weights[weight_type]
51             new_weight = current_weight + weight
52
53             if new_weight < distances[neighbor]:
54                 distances[neighbor] = new_weight
55                 predecessors[neighbor] = current_node
56                 heapq.heappush(pq, (new_weight, neighbor))
```

Gambar 1. Konfigurasi Input Program

```
57
58 # Rekonstruksi Jalur
59 path = []
60 curr = end
61 while curr:
62     path.append(curr)
63     curr = predecessors[curr]
64 path.reverse()
65
66 return distances[end], path
67
68 # 3. Eksekusi Program
69 # Hitung Jarak Terpendek (km)
70 min_dist, path_dist = dijkstra_solver(graph, 'V1', 'V11', 'jarak')
71 print(f"--- HASIL OPTIMASI JARAK ---")
72 print(f"Total Jarak Minimum: {min_dist} km")
73 print(f"Rute: {' -> '.join(path_dist)}")
74
75 # Hitung Waktu Tercepat (menit)
76 min_time, path_time = dijkstra_solver(graph, 'V1', 'V11', 'waktu')
77 print(f"\n--- HASIL OPTIMASI WAKTU ---")
78 print(f"Total Waktu Minimum: {min_time} menit")
79 print(f"Rute: {' -> '.join(path_time)}")
```

Gambar 2. Konfigurasi Input Program (2)

2. Hasil Output Program

```
PS C:\Users\DARA MAYWANTI\OneDrive\文档\Phyton> & "C:/Users/DARA MAYWANTI/AppData/Local/Programs/Python/Python31
3/python.exe" "c:/Users/DARA MAYWANTI/OneDrive\文档\Phyton/main.py/UAS MD.py"
--- HASIL OPTIMASI JARAK ---
Total Jarak Minimum: 46.7 km
Rute: V1 -> V2 -> V3 -> V4 -> V5 -> V6 -> V10 -> V11

--- HASIL OPTIMASI WAKTU ---
Total Waktu Minimum: 62 menit
Rute: V1 -> V2 -> V3 -> V4 -> V5 -> V6 -> V10 -> V11
PS C:\Users\DARA MAYWANTI\OneDrive\文档\Phyton>
```

Gambar 3. Output Program

Berdasarkan dokumentasi visual yang dilampirkan, proses validasi komputasi diawali dengan konfigurasi parameter masukan (*input*) yang bersumber dari model graf visual. Sistem menerima struktur topologi jaringan yang terdiri dari 11 simpul utama, dinotasikan sebagai V_1 hingga V_{11} , di mana setiap sisi penghubung memiliki variabel bobot ganda berupa jarak geometris dan durasi tempuh. Secara spesifik, algoritma memproses logika percabangan di simpul V_4 (Taman Caracas) yang memisahkan jalur menjadi dua opsi, yaitu Jalur A menuju V_5 dengan bobot 13,3 km (16 menit) dan Jalur B menuju V_7 dengan bobot 3,1 km (8 menit). Seluruh data bobot pada visualisasi graf tersebut diterjemahkan menjadi matriks data yang menjadi acuan dasar bagi algoritma dalam melakukan iterasi pencarian rute terbaik.

Selanjutnya, berdasarkan tangkapan layar terminal eksekusi (*output*), algoritma menghasilkan keputusan optimal untuk parameter jarak dengan menampilkan luaran teks "Total Jarak Minimum: 46.7

km". Angka ini diperoleh dari akumulasi bobot sisi pada rute terpilih, yakni urutan lintasan $V_1 \rightarrow V_2 \rightarrow V_3 \rightarrow V_4 \rightarrow V_5 \rightarrow V_6 \rightarrow V_{10} \rightarrow V_{11}$. Hasil luaran ini mengindikasikan bahwa secara matematis, lintasan yang melalui simpul V_5 dan V_6 (Jalur Gronggong-Talun) terbukti sebagai rute terpendek, mengalahkan opsi jalur bawah via Plangon yang memiliki total akumulasi jarak lebih besar. Hal ini mengonfirmasi bahwa algoritma telah berhasil mengidentifikasi efisiensi jarak pada segmen jalan tikus Tahun meskipun harus melewati jalur menanjak sebelumnya.

Pada parameter optimasi waktu, sistem komputasi menampilkan hasil luaran "Total Waktu Minimum: 62 menit". Analisis terhadap hasil ini menunjukkan bahwa meskipun terdapat segmen dengan bobot waktu yang cukup tinggi pada ruas $V_4 \rightarrow V_5$ (16 menit), algoritma tetap merekomendasikan jalur ini sebagai solusi tercepat. Keputusan komputasi ini didasari oleh adanya kompensasi efisiensi waktu yang signifikan pada segmen selanjutnya, yaitu $V_5 \rightarrow V_6$, sehingga total durasi perjalanan tetap lebih rendah dibandingkan jalur alternatif via Planggon yang terdeteksi mencapai 74 menit. Dengan demikian, korelasi antara visualisasi graf sebagai input dan hasil terminal sebagai output menunjukkan konsistensi penuh, membuktikan bahwa model algoritma bekerja akurat dalam memvalidasi rute Via Gronggong-Talun sebagai solusi yang paling optimal baik dari segi jarak maupun waktu.

4. KESIMPULAN

Berdasarkan hasil penelitian dan analisis komputasi yang telah dilakukan, dapat disimpulkan bahwa jaringan jalan dari Desa Cihideung Hilir menuju Kampus UIN Siber Syekh Nurjati Cirebon berhasil dimodelkan secara presisi menggunakan representasi Graf Berarah Berbobot dengan 11 simpul dan 11 sisi. Penerapan Algoritma Dijkstra pada model tersebut membuktikan bahwa rute jalur utama yang dikombinasikan dengan jalan pintas, yakni rute via Gronggong – Talun (Jembatan Merah), merupakan solusi yang paling optimal (Pareto Optimal) bagi pengendara sepeda motor. Rute ini mendominasi efisiensi pada kedua variabel sekaligus, menghasilkan total jarak terpendek sebesar 46,7 km dan estimasi waktu tempuh tercepat selama 62 menit. Hasil ini sekaligus mematahkan asumsi umum yang menyatakan bahwa jalur alternatif via Plangon selalu lebih cepat; faktanya, jalur tersebut memiliki inefisiensi geometris pada segmen Mandirancan yang menyebabkan pembengkakan jarak dan waktu tempuh hingga 74 menit.

Lebih lanjut, analisis mendalam menunjukkan bahwa faktor kunci keberhasilan rute terpilih bukanlah terletak pada kelancaran jalan utama Gronggong semata, melainkan pada keputusan strategis pengambilan rute pintas di simpul Talun (Jembatan Merah). Penggunaan ruas jalan ini terbukti signifikan dalam memangkas waktu perjalanan dengan cara menghindari titik – titik kemacetan lampu merah di wilayah Kota Cirebon. Validitas seluruh temuan ini telah dikonfirmasi melalui simulasi komputasi menggunakan bahasa pemrograman Python, di mana hasil luaran program menunjukkan nilai yang identik dengan perhitungan manual, sehingga tingkat akurasi penelitian ini dapat dipertanggungjawabkan secara ilmiah.

Berdasarkan temuan tersebut, penulis merekomendasikan kepada mahasiswa pelajar (commuter) yang menggunakan sepeda motor untuk memprioritaskan rute Gronggong dengan memanfaatkan jalan pintas Jembatan Merah Talun guna menghemat bahan bakar dan mengoptimalkan waktu perjalanan harian. Jalur alternatif via Planggon sebaiknya hanya dijadikan opsi cadangan apabila terjadi kondisi stagnasi lalu lintas yang parah di jalur utama. Untuk pengembangan penelitian dimasa mendatang, disarankan agar model graf dapat diperluas dengan mengintegrasikan variabel bobot yang lebih dinamis, seperti indeks kerusakan jalan (road roughness indeks) atau data kemacetan waktu nyata (real-time-traffic data), sehingga model dapat memberikan rekomendasi yang lebih adaptif terhadap perubahan kondisi lapangan.

REFERENSI

[1] S. R. I. Gede and N. P. L. Santiari, "Penentuan Rute Evakuasi Bencana Kebakaran Menggunakan

- Algoritma Dijkstra (Studi Kasus: Mempertimbangkan Kemacetan),” *J. Sist. Dan Inform.*, vol. 14, no. 2, pp. 105–112, 2020.
- [2] A. Ratnasari, “Penentuan Jarak Terpendek dan Jarak Terpendek Alternatif Menggunakan Algoritma Dijkstra Serta Estimasi Waktu Tempuh,” in *Prosiding Seminar Nasional Teknologi Informasi & Komunikasi Terapan (Semantik)*, 2013.
- [3] N. Awalloedin, W. Gata, and N. Qomariyah, “Algoritma Dijkstra dalam Penentuan Rute Terpendek pada Jalan Raya Antar Kota Jakarta - Tangerang,” *J. Sist. Informasi, Teknol. Inform. dan Komput.*, vol. 12, no. 2, pp. 1–8, 2022.
- [4] R. Munir, *Matematika Diskrit*. Bandung: Informatika, 2010.
- [5] K. H. Rosen, *Discrete Mathematics and Its Applications*, 8th ed. New York: McGraw-Hill Education, 2019.
- [6] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. Cambridge, MA: MIT Press, 2009.
- [7] Y. Rahmanto, J. Alfian, and D. Damayanti, “Determination of the Shortest Route Distribution of LPG Gas Using Dijkstra Algorithm,” *J. Tekno Kompak*, vol. 15, no. 1, pp. 12–23, 2021.
- [8] R. A. Saputra and A. Yuliani, “Analisis Optimasi Rute Distribusi Logistik Menggunakan Pendekatan Graf Berarah pada Kawasan Industri Jawa Barat,” *J. Mat. Terap. dan Komputasi*, vol. 8, no. 1, pp. 45–52, 2024.
- [9] F. S. P. Samosir, I. P. Sari, and B. S. Negara, “Analisis Algoritma Dijkstra Dalam Menentukan Lintasan Terpendek Pada Rute Distribusi Barang,” *J. Media Inform. Budidarma*, vol. 7, no. 1, pp. 202–211, 2023.
- [10] R. D. Ramadhani and M. A. R. Anshori, “Implementasi Algoritma Dijkstra Untuk Mencari Rute Tercepat Pada Aplikasi Navigasi Kampus,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 6, pp. 1215–1224, 2020.
- [11] D. A. P. Putri and R. W. Arifin, “Implementasi Algoritma Dijkstra Untuk Menentukan Rute Terpendek Lokasi Wisata Berbasis Python,” *J. Ilm. Inform. Glob.*, vol. 13, no. 1, pp. 30–36, 2022.