



Penentuan Jalur Alternatif Menghindari Jalan Rawan Macet Di Kota Karawang Menggunakan Algoritma *Dijkstra*

Ilham Hidayatullah¹, Betha Nurina Sari²

^{1,2,3}Program Studi Teknik Informatika, Fakultas Ilmu Komputer,
Universitas Singaperbangsa Karawang

Abstract

Received: 11 Juli 2022

Revised: 14 Juli 2022

Accepted: 16 Juli 2022

Congestion is one of the society problems, especially in big cities. Congestion occurs because the number of vehicles with width of roads that are not comparable. Therefore an appropriate alternative route selection method is needed to reduce congestion problems. This study aims to determine alternative pathways that are more effective and efficient so as to reduce congestion in certain road segments by determining the smallest weights from each road segmen with Dijkstra Algorithm. This research produces several alternative routes that can be used to avoid certain road traffic jams, that people can choose different alternative routes according to their respective goals. The results of this study indicate that Dijkstra algorithm can determine the optimal path as an appropriate alternative route to reduce congestion problems in the city of karawang.

Keywords: *Dijkstra Algorithm, alternative paths, traffic congestion, smallest weights.*

(*) Corresponding Author: ilham.hdjuniol18025@student.unsika.ac.id¹,
betha.nurina@staff.unsika.ac.id²

How to Cite: Hidayatullah, I., & Sari, B. (2022). Penentuan Jalur Alternatif Menghindari Jalan Rawan Macet Di Kota Karawang Menggunakan Algoritma Dijkstra. *Jurnal Ilmiah Wahana Pendidikan*, 8(13), 190-199. <https://doi.org/10.5281/zenodo.6961635>.

PENDAHULUAN

Kemacetan lalu lintas di indonesia disebabkan oleh meningkatnya jumlah pengguna mobil pribadi. Jumlah kendaraan meningkat setiap tahun, namun belum sebanding dengan perluasan jalan, terutama di kota-kota besar yang banyak menggunakan kendaraan pribadi. Pada tahun 2018, jumlah kendaraan pribadi untuk roda 2 dan roda 4 di indonesia sudah mencapai 136 juta unit dengan peningkatan kurang lebih 10% kendaraan per tahun (bps.go.id, 2018). Kemacetan ini menjadi salah satu masalah yang sering dihadapi masyarakat serta menimbulkan banyak kerugian, salah satunya adalah kerugian waktu. Maka dari itu penyebabnya adalah tidak sebanding jumlah kendaraan dengan lebar atau ruas jalan raya yang dilalui masyarakat (Hanif Ilmi Mardlotillah, Amin Suyitno, 2014).

Kemacetan semakin hari semakin sering terjadi terutama di waktu-waktu ramai kendaraan seperti saat berangkat atau pulang sekolah, kantor, juga pada hari libur atau akhir pekan (Rifanti, 2017). Namun, waktu yang paling rawan kemacetan yang paling tinggi adalah di waktu pagi hari karena sebagian besar aktivitas masyarakat dimulai di pagi hari secara bersamaan. Saat ini, anak sekolah enggan menggunakan kendaraan umum untuk berangkat ke sekolah. Mereka lebih suka diantar jemput oleh orang tuanya hal ini berkontribusi pada kemacetan yang disebabkan oleh meningkatnya jumlah kendaraan pribadi setiap hari.

Ada banyak cara untuk menghindari kemacetan, salah satunya dengan menentukan jalur alternatif yang jumlah kendaraannya lebih sedikit dibandingkan jalur yang terjadi volume kemacetan. Maka dari itu, diperlukan metode pemilihan rute alternatif yang tepat untuk mengurangi masalah kemacetan. Ketika memilih rute alternatif dalam kasus ini perlu memperhitungkan jumlah kendaraan yang melewati segmen jalan terutama selama periode potensi kemacetan (Ismanto hadi & Iryanto, 2018).

Saat menentukan rute alternatif, yang perlu dipertimbangkan adalah kondisi jalan lain atau jumlah kendaraan di jalan lain yang mungkin digunakan sebagai rute alternatif. Penerapan graf merupakan suatu metode yang dapat digunakan untuk menentukan jalur alternatif yang paling sesuai. Ruas-ruas jalan yang rawan kemacetan dapat diinterpretasikan sebagai graf berbobot berarah yang membandingkan jumlah kendaraan yang melintas pada setiap ruas jalan, dan jumlah maksimum kendaraan yang dapat melintasi jalan tersebut dalam satu waktu. Prinsip yang digunakan untuk mencari rute alternatif adalah dengan menggunakan algoritma Dijkstra untuk menentukan rute alternatif dengan mempertimbangkan jumlah minimum kendaraan yang menuju ke beberapa rute alternatif.

Masalah pencarian jalur alternatif ini sangat rumit apabila diterapkan dalam skala yang besar. Pemecahan masalah tersebut dibutuhkan suatu algoritma (Ishlakhuddin & SN, 2021). Beberapa algoritma untuk mencari jalur optimal diantaranya algoritma Dijkstra, algoritma semut (ant colony), algoritma Floyd Warshall, algoritma Bellman Ford, algoritma Distance Vector dan algoritma A-star. Salah satu algoritma yang terkenal dalam hal ini adalah algoritma Dijkstra. Hasil akhir dari algoritma Dijkstra adalah panjang jalur tercepat dari tempat asal ke tempat tujuan beserta rute perjalanannya.

Menurut (Hannants et al., 2012) Algoritma Dijkstra merupakan algoritma yang pencarian grafiknya akan digunakan untuk menyelesaikan masalah jalur terpendek dari suatu sumber dalam graf tersebut, permasalahan tersebut tidak akan memiliki biaya tepi negatif dan akan menghasilkan jalur terpendek. Kemudian akan dibuat jalur alternatif untuk menghindari kemacetan. Algoritma ini, menyelesaikan masalah dengan menghasilkan rute, dari satu lokasi awal sampai satu lokasi tujuan. Cara kerja algoritma Dijkstra memakai strategi greedy, dimana pada setiap langkah dipilih sisi dengan bobot terkecil yang menghubungkan sebuah simpul yang sudah terpilih dengan simpul lain yang belum terpilih. Algoritma Dijkstra membutuhkan parameter berupa tempat asal dan tempat tujuan (Kusuma & Agung, 2019).

Penelitian bertujuan untuk mempelajari ruas-ruas jalan dan menggambarkannya dalam bentuk graf berarah dan berbobot, sehingga setelah menjelaskan ruas-ruas jalan dalam bentuk graf, akan lebih mudah untuk menjelaskan jarak antara ruas jalan yang satu dengan ruas jalan yang lain. Penjelasan grafis dan kemudian menurut waktu tertentu jumlah kendaraan pada setiap bagian dari unit memberikan bobot pada bagian. Penelitian ini menerapkan algoritma Dijkstra penentuan jalur optimal.

Menurut (Intan Alifianu, Muhamad Aznar Abdillah, 2021) prinsip algoritma Dijkstra adalah pada setiap langkah kita memilih sisi yang berbobot minimum dan dimasukkan kedalam himpunan solusi. Misalkan graf berbobot dengan n buah titik dinyatakan dengan matriks ketetanggaan $M=[m_{ij}]$ dengan ketentuan m_{ij} dapat berisi bobot sisi (i,j) 0 (nol) atau tak hingga. Entri-entri matriks ketetanggaan yang ada kemudian dihitung menggunakan algoritma Dijkstra dan dilakukan pengecekan menggunakan metode telusur balik. Oleh karena itu, dibutuhkan sebuah analisis dan perhitungan matematis yang detail dalam proses pemilihan jalur optimal yang akan dipilih dari titik awal ke tempat tujuan sehingga sampai kepada jalur tercepat (Lestari et al., 2020).

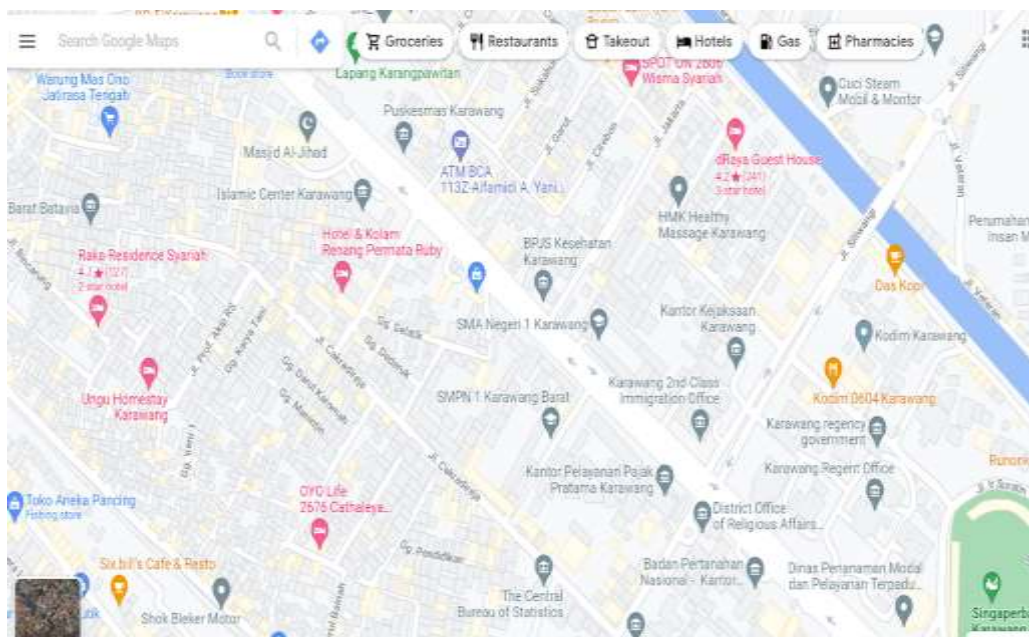
METODOLOGI PENELITIAN

Penelitian dilakukan di Kota Karawang Provinsi Jawa Barat yang terletak antara $107^{\circ}02'$ - $107^{\circ}40'$ BT dan $5^{\circ}56'$ - $6^{\circ}34'$ LS, merupakan data ruas jalan yang sering macet karena terdapat beberapa sekolah di daerah tersebut. Metode yang digunakan pada penelitian ini adalah studi pustaka, pengumpulan data, pemecahan masalah, dan penarikan kesimpulan. Studi pustaka dilakukan dengan mengumpulkan sumber pustaka berupa buku teks, karya ilmiah, dan lain sebagainya. Pengumpulan data dilakukan dengan cara mengambil data sekunder untuk data peta jalan di kota karawang beserta jarak tiap ruas jalannya. Jarak akan dijadikan bobot untuk kemudian dianalisis dan dihitung menggunakan algoritma Dijkstra. Daftar jalan yang dilalui di kota Karawang dapat dilihat pada Tabel 1.

Tabel 1. Daftar Jalan di Kota Karawang

1	Jl. Jenderal Ahmad Yani (SMA Negeri 1 Karawang)
2	Jl. Jakarta (SMP 6 Karawang Barat)
3	Jl. Sukarja Jayalaksana (SMP 1 Karawang Barat)
4	Jl. Jend. A. Yani By Pass No.23 (SMA Negeri 4 Karawang)
5	Jl. Banten No.5 (SMK Negeri 2 Karawang)
6	Jl. Banten By Pass No.5 (SMA Negeri 3 Karawang)
7	Jl. Veteran, KW 6 (SMA Negeri 6 Karawang)
8	Jl. Bogor (SMP Negeri 8 Karawang Barat)
9	Jl. Siliwangi (Sekolah Terpadu Al-Madinah)
10	Jl. Husni Hamid (SDN Nagasari)
11	Jl. Cakradireja (SMP 5 Karawang Barat)
12	Jl. Pangkal Perjuangan (SMK Negeri 1 Karawang)

Berikut ini Gambar 1 dan Gambar 2 menunjukkan peta jalan di kota Karawang yang rawan kemacetan. Gambar 1 menunjukkan jalan di wilayah Desa Nagasari kecamatan Karawang Barat.



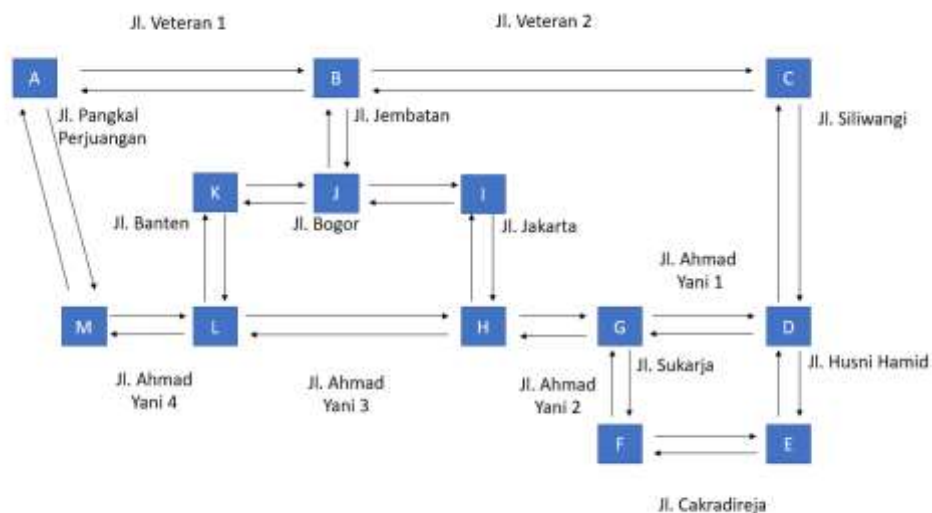
Gambar 1. Jalan di wilayah Desa Nagasari Karawang Barat



Gambar 2. Jalan di wilayah Desa Karangpawitan Karawang Barat

Penentuan Graf Berarah dan Berbobot

Peta pada Gambar 1 dan 2 dapat diartikan sebagai graf berarah berbobot, di mana simpul mewakili perpotongan dan sisi mewakili jalan yang menghubungkan persimpangan. Tepi ini dua arah dan berarah pada gambar di bawah ini menunjukkan bahwa ada beberapa jalan yang hanya satu arah (verboden). Berat tepi grafik menjelaskan volume kendaraan di segmen jalan. Di bawah ini adalah grafik berarah yang menjelaskan peta pada Gambar 1 dan 2, seperti yang ditunjukkan pada Gambar 3.



Gambar 3. Graf berarah dari wilayah rawan kemacetan

Untuk menentukan berat setiap ruas jalan, misalkan kapasitas maksimum ruas jalan per satuan waktu adalah 100 kendaraan, termasuk mobil, sepeda motor atau kendaraan tradisional seperti becak. Apabila jumlah kendaraan yang melintasi ruas jalan tersebut dalam satu satuan waktu tertentu melebihi 100 maka dapat ditentukan akan

terjadi kemacetan lalu lintas pada ruas jalan tersebut, sehingga tidak dapat dipilih sebagai jalur alternatif.

Algoritma Penentuan Jalur Optimal

Pada materi graf mencari jalur optimal dari satu titik ke titik lainnya merupakan masalah yang sering dijumpai dalam kehidupan sehari-hari. Algoritma Dijkstra adalah salah satu cara terbaik dalam mencari jalur optimal (Tyagi, 2017). Algoritma Dijkstra ditemukan oleh ilmuwan komputer dari Belanda yang bernama Edsger Dijkstra tahun 1956 dan dipublikasikan tahun 1959. Algoritma ini hanya cocok untuk graf berbobot positif dan biasanya digunakan untuk mencari jalur terpendek dan optimal dengan bobot terendah. Algoritma Dijkstra merupakan salah satu algoritma pencarian jalur terpendek dengan metode greedy.

Menurut (Gusmão & Pramono, 2013) Algoritma Greedy merupakan algoritma yang pada tahapannya ditentukan berdasarkan pemilihan bobot terbaik. Pada setiap tahapan yang dilalui, algoritma Dijkstra menentukan jalur terpendek dari titik satu ke titik lainnya, secara teru-menerus sehingga menemukan jarak terpendek dari titik pertama ke titik terakhir. Algoritma berkaitan pada serangkaian iterasi pada kumpulan titik yang berbeda dengan menambahkan titik di setiap iterasi. Proses penandaan dilakukan pada setiap iterasi, dimulai dari penandaan a , penandaan dengan 0, dan penandaan titik lainnya. Karena lintasan dari suatu titik ke suatu titik berbeda dengan titik yang ada maka panjang lintasan terpendek antara a dan titik 0. Ilustrasi Algoritma Dijkstra dalam notasi *Pseudo Code* dapat dilihat pada Gambar 4.

```

Procedure Dijkstra (input : matriks, a : titikawal)
Deklarasi
t1, t2 ... , tn : integer
d1 ,d2 ... , dn : integer
i : integer
Algoritma
{ Langkah 0: (inisialisasi)}
Fori ← 1 to n do
ti ← ∞
di ← main
endfor
{ Langkah 1}
ta ← 1
da ← ∞
{ Langkah 2, 3, ... , n-1 }
Fori ← 2 to n-1 do
    Cari j sedemikian sehingga tj = 0 dan
    dj = min(d1, d2, ... , dn)
    tj = 1
    Perbarui di, untuk i = 1, 2, 3, ... , n dengan :
    di (baru) = min(di lama), dj + mij )
endfor
    
```

Gambar 4. *Pseudo Code* Algoritma Dijkstra

Keterangan

- m = Matriks ketetanggaan graf, dengan
- m_{ij} = bobot dari sisi titik i ke titik j jika ada lintasan dari i ke j
- m_{ij} = ∞ , jika tidak ada lintasan dari titik i ke titik j
- t = titik pada graf dengan
- t_i = 1 apabila titik i termasuk titik yang dilalui lintasan terpendek
- t_i = 0 apabila titik i tidak termasuk titik yang dilalui lintasan terpendek
- d_i = bobot total dari titik awal ke titik i

Jalur Terpendek (Shortest Path Problem)

Proses penghitungan ruter terpendek adalah proses mencari jarak terpendek atau biaya terkecil suatu rute dari node awal ke node tujuan dalam sebuah jaringan. Pada proses penghitungan rute terpendek terdapat dua macam proses yaitu proses pemberian label dan proses pemeriksaan node. Metode pemberian label adalah metode untuk memberikan identifikasi pada setiap node dalam jaringan. Pada sebagian besar algoritma penghitungan rute terpendek, terdapat 3 label informasi yang dikelola untuk setiap node i pada proses pemberian label yaitu, label jarak d_i , parent node p_i dan status node s_i . (Yulia et al., 2015)

HASIL DAN PEMBAHASAN

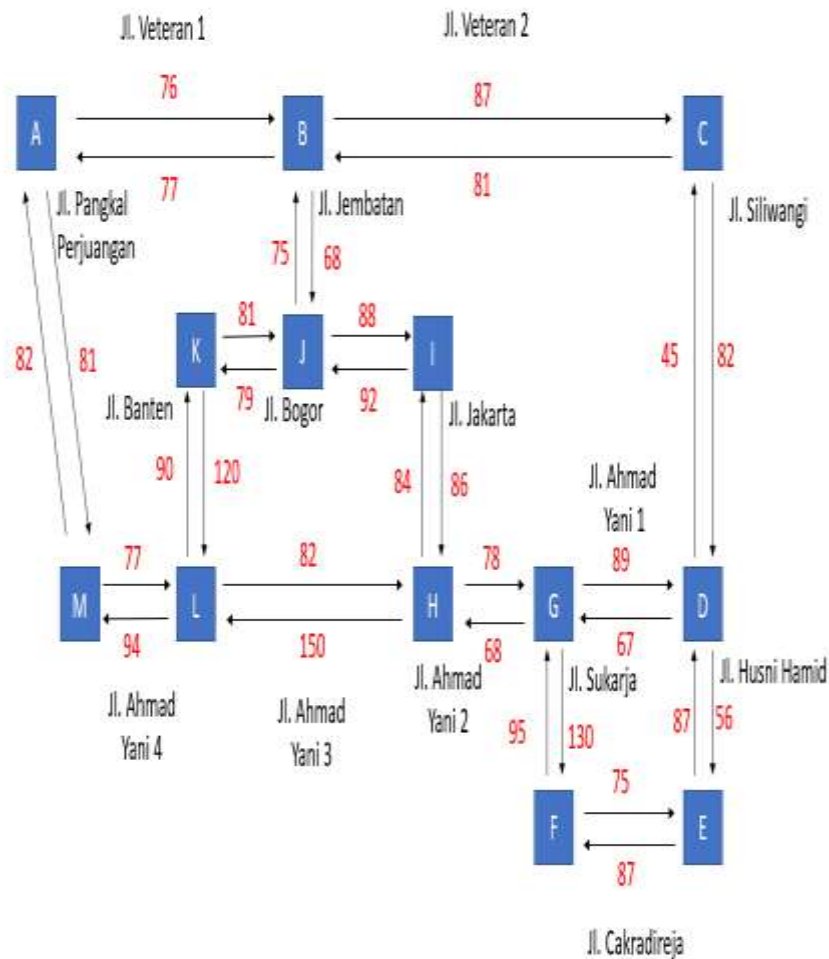
Pada penelitian ini hanya algoritma Dijkstra yang digunakan untuk menentukan rute alternatif yang dipilih untuk menghindari kemacetan di karawang. Graf yang digunakan di sini adalah graf berbobot berarah dengan simpul-simpulnya, sehingga merupakan suatu perpotongan dan sisi-sisi dari graf tersebut mewakili suatu ruas jalan dengan arah dan bobot tertentu. Berat masing-masing sisi merupakan perbandingan volume kendaraan yang melewati suatu satuan waktu dengan volume kendaraan terbesar yang dapat melewati jalan tersebut. Oleh karena itu, jika lebih dari 100 kendaraan melewati jalan tersebut dalam satuan waktu tertentu maka kemacetan dapat terjadi. Penulis mengamati setiap ruas jalan dengan menghitung jumlah kendaraan yang sering melewati ruas jalan yang rawan lalu lintas pada pukul 06:00 08:00 WIB. Hasil data yang bisa dilihat pada graf berarah dan berbobot pada tabel dibawah berikut.

Tabel 2. Data Bobot Ruas Jalan

Nama Jalan	Titik (x,y)	Sisi	Bobot
Jl. Veteran 1	(A,B)	e_1	40/100
	(B,A)	e_2	77/100
Jl. Veteran 2	(B,C)	e_3	87/100
	(C,B)	e_4	81/100
Jl. Siliwangi	(C,D)	e_5	82/100
	(D,C)	e_6	45/100
Jl. Husni Hamid	(D,E)	e_7	56/100
	(E,D)	e_8	87/200
Jl. Cakradireja	(F,E)	e_9	75/100
	(E,F)	e_{10}	87/100
Jl. Sukarja	(G,F)	e_{11}	130/100
	(F,G)	e_{12}	95/100
Jl. Jenderal Ahmad Yani 1	(G,D)	e_{13}	89/100
	(D,G)	e_{14}	67/200
Jl. Jenderal Ahmad Yani 2	(H,G)	e_{15}	78/100
	(G,H)	e_{16}	68/100
Jl. Jenderal Ahmad Yani 3	(L,H)	e_{17}	82/100
	(H,L)	e_{18}	150/100
Jl. Jenderal Ahmad Yani 4	(M,L)	e_{19}	77/100
	(L,M)	e_{20}	94/100
Jl. Jakarta	(I,H)	e_{21}	86/100

	(H,I)	e_{22}	84/100
Jl. Bogor 1	(J,I)	e_{23}	88/100
	(I,J)	e_{24}	92/100
Jl. Jembatan	(B,J)	e_{25}	68/100
	(J,B)	e_{26}	75/100
Jl. Bogor 2	(K,J)	e_{27}	81/100
	(J,K)	e_{28}	79/100
Jl. Banten	(K,L)	e_{29}	120/100
	(L,K)	e_{30}	90/100
Jl. Pangkal Perjuangan	(A,M)	e_{31}	81/100
	(M,A)	e_{32}	82/100

Dari Tabel 2 menjelaskan tentang nilai bobot yang telah didata ketika penelitian pada setiap ruas jalan di wilayah karawang, maka grafik berarah yang menjelaskan tabel 2 dapat dilihat pada Gambar 5.



Gambar 5. Graf berarah dan berbobot ruas jalan

Dari grafik yang ditunjukkan pada Gambar 5 dimana bobot masing-masing segmen sudah diketahui, maka dari itu tentukan matriks ketetanggaan. Menurut definisi matriks ketetanggaan m , yaitu m_{ij} = berat dari titik i ke titik j terdapat jalur dari titik i ke titik j , dan tidak ada jalur dari titik i ke titik j , maka matriks *adjacency* dari grafik yang pada Gambar 6 diperoleh sebagai berikut.

	A	B	C	D	E	F	G	H	I	J	K	L	M
A	0	76	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	81
B	77	0	87	∞	∞	∞	∞	∞	∞	68	∞	∞	∞
C	∞	81	0	82	∞	∞	∞	∞	∞	∞	∞	∞	∞
D	∞	∞	45	0	56	∞	67	∞	∞	∞	∞	∞	∞
E	∞	∞	∞	87	0	87	∞	∞	∞	∞	∞	∞	∞
F	∞	∞	∞	∞	75	0	95	∞	∞	∞	∞	∞	∞
G	∞	∞	∞	89	∞	130	0	68	∞	∞	∞	∞	∞
H	∞	∞	∞	∞	∞	∞	78	0	84	∞	∞	150	∞
I	∞	∞	∞	∞	∞	∞	∞	86	0	92	∞	∞	∞
J	∞	75	∞	∞	∞	∞	∞	∞	88	0	79	∞	∞
K	∞	∞	∞	∞	∞	∞	∞	∞	∞	81	0	120	∞
L	∞	∞	∞	∞	∞	∞	∞	82	∞	∞	90	0	94
M	82	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	77	0

Gambar 6. Matriks adjacency

Matriks yang berdekatan menunjukkan jumlah kendaraan yang melewati segmen jalan pada waktu tertentu. Misalnya, nilai pada baris pertama dan kolom kedua adalah 76, yang berarti ada sekitar 766 mobil di jalan AB (sisi e1). Dengan kata lain, tidak terjadi kemacetan buruk di jalan AB (Jl Veteran 1) karena volume kendaraan yang padat. Jumlah kendaraan yang lewat masih kurang dari 100. Kemudian perhatikan bahwa nilai yang ditampilkan pada baris ketujuh dan kolom keenam adalah 130, yang berarti ada sekitar 130 kendaraan di bagian GF (sisi e11). Dengan kata lain, jalan GF (Jl. Sukaraja) terkena macet karena lebih dari 100 kendaraan yang lewat, secara umum jika nilai $m_{ij} \geq$ pada matriks lingkungan adalah 100, maka terjadi kemacetan pada ruas jalan tersebut, sehingga tidak dapat digunakan sebagai jalur alternatif untuk menghindari kemacetan jalan baru. Kemudian cari bobot dengan nilai terkecil dari titik A ke semua titik selain A pada Gambar 5 dengan menggunakan Algoritma Dijkstra.

Tabel 3. Bobot total dari titik A ke titik lainnya

Tahap	Titik terpilih	B	C	D	E	F	G	H	I	J	K	L	M
1	A	76	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	81
2	B	76	163	∞	∞	∞	∞	∞	∞	146	∞	∞	81
3	M	76	163	∞	∞	∞	∞	∞	∞	146	∞	158	81
4	C	76	163	245	∞	∞	∞	∞	∞	146	∞	158	81

5	L	76	163	245	∞	∞	∞	240	∞	146	248	158	81
6	D	76	163	245	301	∞	312	240	∞	146	248	158	81
7	K	76	163	245	301	∞	312	240	∞	329	248	158	81
8	J	76	163	245	301	∞	312	240	232	146	223	158	81
9	I	76	163	245	301	∞	312	318	232	146	223	158	81
10	H	76	163	245	301	∞	312	318	232	146	223	158	81
11	G	76	163	407	301	442	312	318	232	146	223	158	81
12	F	76	163	407	513	388	312	318	232	146	223	158	81

Berdasarkan data pada tabel 3 dapat disimpulkan jalur optimal yang akan dipilih untuk menghindari kemacetan yaitu sebagai berikut.

1. Dari titik A ke arah titik B melalui ruas jalan A-B (Jl Veteran 1) dengan bobot total 76.
2. Dari titik A ke arah titik M melalui ruas jalan A-M (Jl Pangkal Perjuangan) dengan bobot total 81.
3. Dari titik A ke arah titik C melalui ruas jalan A-B-C (Jl Veteran 1-Jl Veteran 2) dengan bobot total 163.
4. Dari titik A ke arah titik L melalui ruas jalan A-M-L (Jl Pangkal Perjuangan – Jl Ahmad Yani 4) dengan bobot total 158.
5. Dari titik A ke arah titik J melalui ruas jalan A-B-J (Jl Veteran 1 – Jl Jembatan) dengan bobot total 144.
6. Dari titik A ke arah titik D melalui ruas jalan A-B-C-D (Jl Veteran 1 – Jl veteran 2 – Jl Siliwangi) dengan bobot total 245.
7. Dari titik A ke arah titik K melalui ruas jalan A-M-L-K (Jl Pangkal Perjuangan – Jl Bogor 2) dengan bobot total 232.
8. Dari titik A ke arah titik I melalui ruas jalan A-B-J-I (Jl Veteran 1 – Jl Jembatan – Jl Bogor 2 – Jl Jakarta) dengan bobot total 318.
9. Dari titik A ke arah titik G melalui ruas jalan A-B-C-D-G-E-F (Jl Veteran 1 – Jl Veteran 2 – Jl Siliwangi – Jl Ahmad Yani 1) dengan bobot total 312.
10. Dari titik A ke arah titik F melalui ruas jalan A-B-C-D-E-F (Jl Veteran 1 – Jl Veteran 2 – Jl Siliwangi – Jl Husni Hamid) dengan bobot total 388.

KESIMPULAN

Hasil penelitian ini menghasilkan beberapa alternatif rute yang dapat digunakan untuk menghindari kemacetan lalu lintas jalan tertentu, terutama yang memiliki bobot lebih dari 100, sehingga masyarakat dapat memilih jalur alternatif yang berbeda sesuai dengan tujuannya masing masing. Hasil penelitian ini menunjukkan bahwa algoritma Dijkstra dapat menentukan jalur optimal sebagai rute alternatif yang tepat untuk mengurangi masalah kemacetan di kota Karawang.

DAFTAR PUSTAKA

- bps.go.id. (2018). Perkembangan Jumlah Kendaraan Bermotor Menurut Jenis. In *Badan Pusat Statistik* (pp. 1–1). <https://www.bps.go.id/linkTableDinamis/view/id/1133>
- Gusmão, A., & Pramono, S. H. (2013). *Sistem Informasi Geografis Pariwisata Berbasis Web Dan Pencarian Jalur Terpendek Dengan Algoritma Dijkstra*. 7(2), 125–130.
- Hanif Ilmi Mardlotillah, Amin Suyitno, F. Y. A. (2014). Simulasi Algoritma Dijkstra Dalam Menangani Masalah Lintasan Terpendek Pada Graf Menggunakan Visual

- Basic. *Unnes Journal of Mathematics*, 4(2), 3–8.
- Hannants, M., Ichsan, H., Yudaningtyas, E., & Muslim, M. A. (2012). *Solusi Optimal Pencarian Jalur Tercepat dengan Algoritma Hybrid Fuzzy-Dijkstra*. 6(2), 155–160.
- Intan Alifianu, Muhamad Aznar Abdillah, I. S. (2021). *Solusi Optimal Pencarian Jalur Tercepat Menggunakan Algoritma Dijkstra Untuk Mencari*. 8(2), 140–148.
- Ishlakhuddin, F., & SN, A. (2021). Ontology-based Chatbot to Support Monitoring of Server Performance and Security By Rule-base. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 15(2), 131. <https://doi.org/10.22146/ijccs.58588>
- Ismanto hadi, E., & Iryanto, I. (2018). Penerapan Algoritma Dijkstra untuk Penentuan Jalur Terbaik Evakuasi Tsunami – Studi Kasus: Kelurahan Sanur Bali. *JTT (Jurnal Teknologi Terapan)*, 4(2), 72–78. <https://doi.org/10.31884/jtt.v4i2.79>
- Kusuma, E., & Agung, H. (2019). Aplikasi Perhitungan Dan Visualisasi Jarak Terpendek Berdasarkan Data Coordinate Dengan Algoritma Dijkstra Dalam Kasus Pengantaran Barang Di Kawasan Jabodetabek. *Aplikasi Perhitungan Dan Visualisasi Jarak Terpendek Berdasarkan Data Coordinate Dengan Algoritma Dijkstra Dalam Kasus Pengantaran Barang Di Kawasan Jabodetabek*, 08(1), 14–23.
- Lestari, S. L., Ardiansyah, A., Giovani, A. P., & Dwijayanti, D. (2020). a Dijkstra Algorithm Implementation in Determining Shortest Route To Mosque in Residential Citra Indah City. *Jurnal Pilar Nusa Mandiri*, 16(1), 65–70. <https://doi.org/10.33480/pilar.v16i1.1199>
- Rifanti, U. M. (2017). *Pemilihan Rute Terbaik Menggunakan Algoritma Dijkstra Untuk Mengurangi (Best Route Selection Use Dijkstra Algorithm To Reduce*. 2(2), 90–99.
- Tyagi, K. A. (2017). *Bellman Ford Shortest Path Algorithm using Global Positioning System IRJET-Power Management in Wireless Sensor Network*. 4.
- Yulia, W. E., Istiadi, D., & Roqib, A. (2015). *Pencarian spbu terdekat dan penentuan jarak terpendek menggunakan algoritma dijkstra*. 1, 89–93.