

Penggunaan Algoritma Dijkstra dalam Penentuan Jalur Alternatif untuk Mengurangi Kemacetan Lalu Lintas

Sri Umay Nur'aini Sholihah (13514007)

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

sriumayns@students.itb.ac.id

Abstrak—Kemacetan di jalan merupakan salah satu masalah dihadapi banyak kota besar. Permasalahan kemacetan dapat disebabkan oleh berbagai faktor, salah satunya tidak sebandingnya volume kendaraan dengan dengan ruas jalan. Permasalahan kemacetan tentu menimbulkan banyak kerugian antara lain waktu dan bahan bakar yang terbuang untuk menghadapi kemacetan. Salah satu upaya penanganan kemacetan untuk jangka pendek (saat kemacetan terjadi) adalah dengan pengalihan kendaraan ke jalur alternatif yang volume kendaraannya lebih kecil. Untuk itu diperlukan cara pemilihan jalur alternatif yang tepat untuk mengurai kemacetan agar penguraian kemacetan berjalan efektif dan efisien. Makalah ini akan membahas aplikasi dari graf berarah-berbobot beserta algoritma yang sesuai dalam teori graf untuk mencari jalur alternatif guna mengurai kemacetan dengan se-efektif dan se-efisien mungkin.

Kata Kunci—jalur alternatif, kemacetan, volume kendaraan, graf, algoritma Dijkstra.

I. PENDAHULUAN

Kemacetan lalu lintas merupakan salah satu permasalahan yang dihadapi banyak kota besar dan menjadi agenda yang harus diselesaikan oleh pihak berwenang karena semakin hari permasalahan kemacetan lalu lintas ini semakin mengkhawatirkan.^{[1][2]} Salah satu faktor utama penyebab kemacetan disuatu ruas jalan adalah tidak sebandingnya ruas jalan dengan volume kendaraan yang ingin melintasi jalan tersebut. Hal ini diperparah dengan semakin bertambahnya jumlah pengguna kendaraan seiring berjalannya waktu yang tidak dibarengi dengan pembangunan, penambahan, atau pelebaran ruas jalan. Akibatnya 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.

Penanganan permasalahan kemacetan memang bisa dibidang cukup sulit dan membutuhkan waktu yang tidak singkat, namun saat kemacetan terjadi tentu diperlukan suatu penanganan agar kemacetan tidak terjadi berlarut-larut sebab banyak kerugian yang diakibatkan dari kemacetan ini, antara lain waktu yang terbuang percuma untuk menghadapi kemacetan, juga pemborosan bahan

bakar kendaraan sebab meskipun kendaraan berhenti tapi mesin kendaraan tetap menyala, selain itu juga menimbulkan polusi yang lebih banyak.^{[1][2]} Dari penelitian yang dilakukan oleh Bappenas pada tahun 2006 didapatkan data bahwa kemacetan yang terjadi di Jakarta menimbulkan kerugian ekonomi sebesar Rp 7 Trilyun per tahun, padahal data ini hanya dihitung dari dua sektor sektor saja, yaitu dari sektor energi (Rp 5,57 Trilyun per tahun) dan dari sektor kesehatan (Rp 1,7 Trilyun per tahun).^[1]



Gambar 1.1 Kemacetan di Jalan Ir. Juanda, Bandung (Sabtu, 18 Februari 2015)

Sumber :

<http://www.pikiran-rakyat.com/foto/2015/02/28/318013/kemacetan-di-kota-bandung-rugikan-rp436-triliun-tahun> (Diakses pada 5 Desember 2015)

Oleh karena itu diperlukan suatu cara untuk mengurai kemacetan yang terjadi. Meskipun hanya jangka pendek, cara penguraian kemacetan dengan efektif dan efisien dapat mengurangi kerugian yang disebabkan oleh kemacetan. Salah satu cara penguraian kemacetan adalah mengalihkan kendaraan ke jalur alternatif agar volume kendaraan tidak menumpuk disalah satu ruas jalan tertentu.

Pemilihan jalur alternatif untuk membantu menguraikan kemacetan perlu dipertimbangkan dengan tepat agar pengalihan jalur dapat mengurai kemacetan di jalur yang dialihkan bukan malah menjadikan jalur alternatif menjadi

titik kemacetan baru, oleh karena itu diperlukan cara pemilihan jalur alternatif yang tepat untuk menentukan jalur alternatif ini berdasarkan banyaknya volume kendaraan yang melintas di ruas-ruas jalan yang bersangkutan. Dalam makalah ini akan dibahas cara pemilihan jalur alternatif guna menangani masalah kemacetan dengan menggunakan salah satu konsep dalam matematika diskrit, yaitu graf, juga algoritma Dijkstra yang berhubungan dengan kajian ini dalam teori graf untuk menyelesaikan permasalahan pemilihan jalur alternatif dengan tepat.

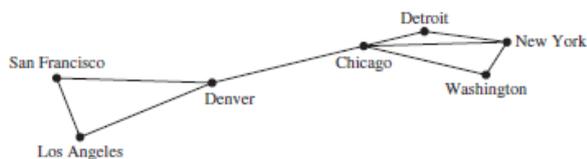
II. TEORI DASAR

A. Graf

1. Definisi Graf

Graf adalah struktur diskrit yang terdiri dari himpunan simpul (*vertices*) dan himpunan sisi (*edge/arcs*) yang merupakan penghubung antara satu simpul dengan simpul lainnya di dalam himpunan simpul.^[3]

Secara matematis graf dapat didefinisikan sebagai pasangan dari dua buah himpunan. Misal G adalah sebuah graf, maka graf G terdiri atas pasangan himpunan simpul (misal himpunan simpul adalah V) dan himpunan sisi (misal himpunan sisi adalah E), atau dapat dituliskan sebagai $G = (V, E)$. Himpunan simpul (V) tidak boleh kosong, jadi sebuah graf pasti memiliki setidaknya satu buah simpul. Sementara himpunan sisi (E) boleh kosong sehingga sebuah graf dapat memiliki hanya simpul saja namun tidak memiliki sisi.^[3]



Gambar 2.1 Contoh graf untuk merepresentasikan komputer network.

Sumber :

Rosen, Kenneth H., *Discrete Mathematics and Its Applications*, 7th edition. New York : McGraw-Hill, 2013, chapter 10, halaman 642.

2. Graf Berarah

Graf berarah $G = (V, E)$ terdiri dari himpunan simpul V dan himpunan sisi berarah E , dengan setiap sisi berarah menghubungkan sepasang simpul pada himpunan V , dimana untuk sisi berarah (u, v) berarti sisi berarah menghubungkan simpul u dan simpul v dengan u sebagai simpul awal dan v sebagai simpul akhir.^[3]



Gambar 2.2 Contoh graf berarah

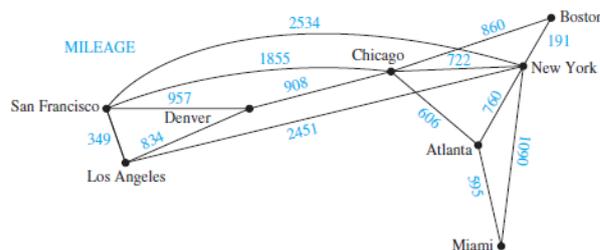
Sumber :

Rosen, Kenneth H., *Discrete Mathematics and Its Applications*, 7th edition. New York : McGraw-Hill, 2013, chapter 10, halaman 643.

3. Graf Berbobot

Graf berbobot adalah graf dengan setiap sisi dari graf tersebut memiliki sebuah bobot yang melambangkan suatu nilai tertentu.^[4] Bobot pada graf dapat menyatakan berbagai macam nilai, misal jarak antar dua kota, biaya perjalanan antar dua kota, waktu tempuh pesan dari sebuah simpul komunikasi ke simpul komunikasi lain, biaya produksi, dan sebagainya.^[4]

Dalam makalah ini bobot dari graf berbobot yang akan digunakan adalah perbandingan volume kendaraan yang melalui ruas jalan tertentu dengan volume kendaraan maksimal yang dapat melintas jalan tersebut dalam suatu satuan waktu.



Gambar 2.3 Contoh graf berbobot

Sumber :

Rosen, Kenneth H., *Discrete Mathematics and Its Applications*, 7th edition. New York : McGraw-Hill, 2013, chapter 10, halaman 708.

4. Lintasan

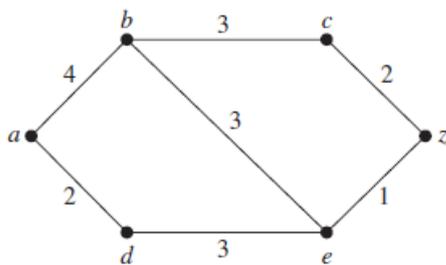
Lintasan adalah jarak yang yang ditempuh dari suatu simpul awal ke simpul tujuan dalam sebuah graf. Lintasan dari v_0 ke v_n , dengan v_0 sebagai simpul awal dan v_n sebagai simpul tujuan, pada graf G dituliskan sebagai barisan dari simpul dan sisi yang dilewati dalam lintasan, dengan lebih tepatnya bergantian antara simpul dan sisinya secara terurut, misal $v_0, e_1, v_1, e_2, v_2, \dots, v_{n-1}, e_n, v_n$ yang berarti sisi e_1 menghubungkan simpul v_0 dengan v_1 atau $e_1 = (v_0, v_1)$, sisi e_2 menghubungkan simpul v_1 dengan simpul v_2 , atau $e_2 = (v_1, v_2)$, dan seterusnya hingga sisi e_n menghubungkan simpul v_{n-1} dengan simpul v_n , atau $e_n = (v_{n-1}, v_n)$ dan setiap sisi e merupakan sisi dari graf G .^[4]

Dari satu titik ke titik yang lain dalam suatu graf dimungkinkan untuk memiliki lebih dari satu lintasan. Penulisan lintasan untuk graf sederhana, yaitu graf yang tidak memiliki sisi ganda dan sisi gelang, dapat dituliskan dengan menuliskan simpul-simpul yang dilewati saja. Contoh pada graf gambar 2.1, salah satu lintasan dari San Francisco ke Washington dapat dituliskan sebagai San Francisco, Denver, Chicago, Washington. Untuk graf

berarah, penentuan lintasan harus sesuai dengan arah dari sisinya. Misal untuk graf berarah pada gambar 2.2, dari ada lintasan dari Chicago menuju San Fransisco, sebab meskipun ada sisi yang bersisian dengan San Fransisco, arah dari sisi-sisi tersebut adalah keluar dari San Fransisco bukan menuju San Fransisco. Sementara pada graf yang tidak sederhana, penulisan diselang-seling antara simpul dan sisinya.

5. Lintasan Terpendek

Lintasan terpendek adalah lintasan dari suatu simpul graf (simpul awal) ke simpul tujuan pada suatu graf berbobot, dengan jumlah bobot dari sisi-sisi pada lintasannya seminimal mungkin.^[4] Misal untuk graf pada gambar 2.3, lintasan terpendek dari San Fransisco menuju Atlanta, dengan melalui lintasan San Fransisco, Chicago, Atlanta dengan jumlah bobot lintasannya 2461. Banyak cara untuk mengetahui lintasan terpendek dari suatu graf berbobot, cara yang paling sederhana adalah dengan menddata seluruh kemungkinan lintasan dari titik awak ke titik tujuan kemudian membandingkan bobot dari lintasan-lintasan tersebut, lintasan dengan bobot paling kecil adalah lintasan terpendek dari titik awak ke titik tujuan tersebut.



Gambar 2.4 Contoh graf berbobot sederhana

Sumber :

Rosen, Kenneth H., *Discrete Mathematics and Its Applications*, 7th edition. New York : McGraw-Hill, 2013, chapter 10, halaman 710.

Contoh untuk graf pada gambar 2.4, berikut cara untuk mencari lintasan terpendek dari simpul a menuju simpul z. Langkah pertama, data seluruh lintasan dari simpul a ke simpul z beserta bobot lintasannya.

Lintasan	Bobot
a, b, c, z	9
a, b, e, z	8
a, d, e, z	6
a, d, e, b, c, z	10

Tabel I. Data lintasan beserta bobotnya dari simpul a ke z pada gambar 2.4

Sumber : Olahan penulis

Langkah kedua, dari data tersebut, cari lintasan dengan bobot yang paling kecil. Untuk kasus ini, dapat dilihat dari tabel diatas lintasan yang memiliki bobot terkecil adalah

lintasan a,d,e,z maka lintasan terpendek dari simpul a ke simpul z adalah lintasan a,d,e,z. Namun cara ini tentu saja bukan cara yang paling efektif untuk menentukan lintasan terpendek. Banyak algoritma untuk mencari lintasan terpendek yang lebih efektif.

6. Matriks Ketetanggan (*Adjacency Matrix*)

Salah satu cara merepresentasikan graf adalah dengan menggunakan matriks, contoh matriks yang dapat digunakan untuk merepresentasikan graf adalah matriks ketetanggaan. Matriks ketetanggaan berbentuk matriks dua dimensi berukuran $n \times n$, dengan n merupakan jumlah simpul pada matriks dan $n \geq 1$. Misal matriks ketetanggan diberi nama M dengan elemen m_{ij} , i sebagai indeks baris dan j sebagai indeks kolom.^[4] Maka isi dari m_{ij} bergantung pada jenis grafnya sebagai berikut.

a. Pada graf tidak berbobot

$m_{ij} = 0$, jika tidak ada lintasan dari simpul i menuju simpul j^{[3][4]}

$m_{ij} = 1$, jika ada lintasan dari simpul i menuju simpul j^{[3][4]}

b. Pada graf berbobot

m_{ij} = bobot dari sisi dari simpul i ke simpul j, jika ada lintasan dari i ke j^{[3][4]}

$m_{ij} = \infty$, jika tidak ada lintasan dari simpul i ke simpul j^{[3][4]}

	a	b	c	d	e	z
a	∞	4	∞	2	∞	∞
b	4	∞	3	∞	3	∞
c	∞	3	∞	∞	∞	2
d	2	∞	∞	∞	3	∞
e	∞	3	∞	3	∞	1
z	∞	∞	2	∞	1	∞

Gambar 2.5 Contoh matriks ketetanggaan dari graf pada gambar 2.4

Sumber : Olahan penulis

B. Algoritma Dijkstra

Algoritma Dijkstra adalah salah satu algoritma yang paling terkenal untuk pencarian lintasan terpendek (*shortest path*) dengan pendekatan *greedy algorithm*.^[5] *Greedy algorithm* merupakan algoritma yang setiap tahapannya ditentukan berdasarkan pemilihan solusi atau nilai/bobot terbaik (tergantung konteks dari penggunaannya, bisa nilai terkecil atau terbesar) sementara disetiap tahap yang dilalui.^[6] Algoritma ini diaplikasikan pada sebuah graf berarah dan berbobot, namun algoritma Dijkstra juga dapat digunakan untuk mencari lintasan terpendek pada graf berbobot namun tidak berarah.^[4]

Cara kerja Algoritma Dijkstra adalah sebagai berikut.

1. Pada setiap langkah dari satu simpul ke simpul yang lain, simpul yang dipilih adalah simpul yang berbobot paling kecil.
2. Langkah yang dipilih dimasukkan dalam himpunan solusi.

Misal untuk sebuah graf berbobot dengan n buah simpul, untuk mencari lintasan terpendek dari suatu simpul a ke simpul i maka akan dibutuhkan :

- Matriks ketetanggaan $M = [m_{ij}]$ dari graf tersebut dengan,
 m_{ij} = bobot dari sisi dari simpul i ke simpul j , jika ada lintasan dari i ke j
 $m_{ij} = 0$, jika $i = j$ (menuju dirinya sendiri)
 $m_{ij} = \infty$, jika tidak ada lintasan dari simpul i ke simpul j
- Larik $S = [s_i]$ dengan,
 $s_i = 1$ apabila simpul i termasuk simpul yang dilalui lintasan terpendek
 $s_i = 0$ apabila simpul i tidak termasuk simpul yang dilalui lintasan terpendek
- Larik $D = [d_i]$ dengan,
 d_i = panjang lintasan dari suatu simpul awal ke simpul tujuan i .

```
procedure Dijkstra (input : matriks, a : simpul awal)
{ Mencari lintasan terpendek dari simpul awal a ke
  semua simpul lainnya
  Masukan : matriks ketetanggaan (m) dari graf berbobot G
  dan simpul awal a
  Keluaran : lintasan terpendek dari a ke semua simpul
  lainnya
}

Deklarasi
s1,s2, ..., sn : integer { larik integer }
d1,d2, ..., dn : integer { larik integer }
i : integer

Algoritma
{ Langkah 0 (inisialisasi)
  for i ← 1 to n do
    si ← 0
    di ← mai
  endfor
{ Langkah 1 }
sa ← 1 {karean simpul a adalah simpul asal lintasan
  terpendek, jadi simpul a sudah pasti terpilih
  dalam lintasn terpendek}
da ← ∞ {tidak ada lintasan terpendek dari simpul
  a ke a}

{ Langkah 2,3, ..., n-1 }
for i←2 to n-1 do
  cari j sedemikian sehingga sj = 0 dan
  dj = min(d1,d2,...,dn)
  sj = 1 {simpul j sudah terpilih ke dalam lintasan
  terpendek}
  perbarui di, untuk i = 1, 2, 3, ..., n dengan :
  di (baru) = min(di (lama), dj + mji)
endfor
```

Gambar 2.5 Ilustrasi algoritma Dijkstra dalam notasi pseudo-code

Sumber :

Munir, Rinaldi, *Diktat Kuliah IF2120 Matematika Diskrit*, Bandung : Teknik Infomatika Institut Teknologi Bandung, 2003.

III. ANALISIS

A. Hubungan Kemacetan, Kondisi Jalan, dan Volume Kendaraan yang Melintas

Seperti yang telah disebutkan di bab sebelumnya, salah satu penyebab utama terjadinya kemacetan adalah tidak seimbangny volume kendaraan yang melintas dengan ruas jalan yang dilalui. Misal suatu ruas jalan memiliki kapasitas untuk dilalui oleh 500 kendaraan untuk setiap satuan waktu, maka apabila dalam suatu satuan waktu tersebut jumlah kendaraan yang ingin melintas melebihi kapasitas tersebut dimungkinkan akan terjadi kemacetan. Faktor yang mempengaruhi kapasitas jalan untuk dilalui kendaraan antara lain lebar jalan, bentuk jalan, dan banyaknya persimpangan yang menggunakan lampu lalu lintas pada ruas jalan tersebut (yang berarti akan ada waktu dimana kendaraan berhenti, sehingga apabila tidak seimbang dengan volume kendaraan yang ingin melewati jalur tersebut akan terjadi penumpukan kendaraan).

B. Analisis Permasalahan

Mengatasi kemacetan saat kemacetan itu terjadi merupakan hal yang harus dilakukan dengan tepat agar kemacetan tidak terjadi berlarut-larut. Salah satu caranya adalah dengan mengalihkan kendaraan ke jalur alternatif agar kendaraan tidak menumpuk di satu jalur saja. Dari satu titik ke satu titik yang lain dimungkinkan memiliki banyak variasi jalur yang dapat dipilih. Namun untuk mengurai kemacetan tentu tidak semua jalur dapat digunakan, ada beberapa faktor yang perlu diperhatikan salah satunya volume kendaraan di jalur lain. Karena itulah diperlukan cara pemilihan jalur alternatif yang tepat agar jalur alternatif yang dipilih tidak menjadi titik kemacetan baru dan kemacetan dapat teratasi dengan lebih cepat.

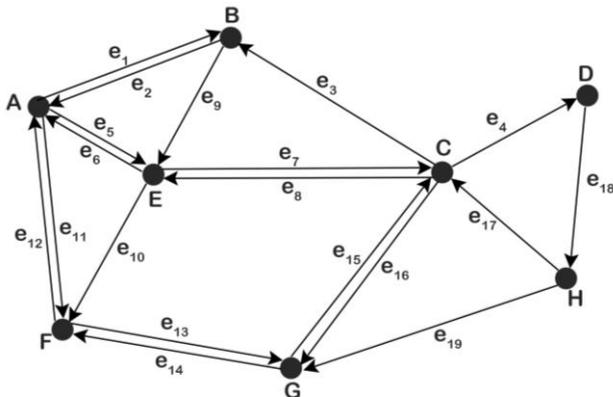
IV. PENCARIAN JALUR ALTERNATIF DENGAN KONSEP LINTASAN TERPENDEK MENGGUNAKAN ALGORITMA DIJKSTRA

Dalam pencarian jalur alternatif ini akan digunakan algoritma Dijkstra pada graf yang merepresentasikan suatu jalur lalu lintas. Graf yang digunakan adalah graf berarah dan berbobot, dengan simpul graf merupakan suatu titik/daerah tertentu, sisi melambangkan jalan dengan arahnya merupakan arah dari arus kendaraan, dan bobot dari sisi melambangkan perbandingan antara volume kendaraan yang lewat dengan volume kendaraan maksimal dalam suatu satuan waktu yang dapat melintasi jalan tersebut. Volume kendaraan maksimal merupakan nilai maksimal yang apabila volume kendaraan yang lewat melebihi volume maksimal maka dijalan tersebut akan terjadi kemacetan. Contohnya jika suatu jalan memiliki volume kendaraan maksimal 100 kendaraan dalam suatu satuan waktu dan volume kendaraan yang lewat 70 dalam suatu satuan waktu maka bobot dari sisi tersebut adalah

70/100.

Data dari graf akan dituliskan dalam sebuah matriks ketetanggaan, dengan satu aturan tambahan yaitu apabila bobot dari sisi ≥ 1 , yang berarti volume kendaraan yang lewat melebihi volume kendaraan maksimal atau sedang terjadi kemacetan di sisi/jalan tersebut, maka bobot dari sisi tersebut akan ditulis sebagai ∞ , yang artinya sisi diasumsikan dihilangkan karena tidak bisa dilewati atau sedang macet.

Berikut akan dibahas cara pencarian jalur alternatif dengan algoritma Dijkstra. Misal ada graf yang merepresentasikan jalur lalu lintas di suatu kawasan seperti gambar berikut.



Gambar 4.1 Graf yang merepresentasikan jalur lalu lintas
Sumber : Buatan penulis

Dan data dari graf tersebut adalah sebagai berikut.

Nama Jalan	Simpul (x,y)	Sisi	Bobot
Jalan Mawar	(A,B)	e_1	53/100
	(B,A)	e_2	71/100
Jalan Flamboyan	(C,B)	e_3	30/100
Jalan Sedap Malam	(C,D)	e_4	44/100
Jalan Kamboja	(A,E)	e_5	121/100
	(E,A)	e_6	60/100
Jalan Anggrek	(E,C)	e_7	75/100
	(C,E)	e_8	51/100
Jalan Begonia	(B,E)	e_9	46/100
Jalan Pakis	(E,F)	e_{10}	52/100
Jalan Melati	(A,F)	e_{11}	77/100
	(F,A)	e_{12}	63/100
Jalan Aster	(F,G)	e_{13}	54/100
	(G,F)	e_{14}	103/100
Jalan Alamanda	(G,C)	e_{15}	65/100
	(C,G)	e_{16}	70/100
Jalan Raflessia	(H,C)	e_{17}	82/100
Jalan Seruni	(D,H)	e_{18}	49/100
Jalan Aster	(H,G)	e_{19}	34/100

Tabel II. Data dari graf pada gambar 4.1

Sumber : Olahan penulis

Maka matriks ketetanggaan dari graf dapat dibuat sebagai berikut.

	A	B	C	D	E	F	G	H
A	0	53	∞	∞	∞	77	∞	∞
B	71	0	∞	∞	46	∞	∞	∞
C	∞	30	0	44	51	∞	70	∞
D	∞	∞	∞	0	∞	∞	∞	49
E	60	∞	75	∞	0	52	∞	∞
F	63	∞	∞	∞	∞	0	54	∞
G	∞	∞	65	∞	∞	∞	0	∞
H	∞	∞	82	∞	∞	∞	34	0

Gambar 4.2 Matriks ketetanggaan graf pada gambar 4.1
Sumber : Olahan penulis

Bobot dari matriks ketetanggaan tersebut hanya dituliskan volume kendaraan yang melintas saja, sebab volume maksimalnya sama untuk semua sisi. Apabila volume maksimalnya tidak sama, dapat disamakan terlebih dahulu perbandingannya sehingga lebih mudah untuk diolah.

Pada matriks ketetanggaan tersebut dapat dilihat bahwa sisi e_5 dan e_{14} dianggap tidak ada sebab perbandingan bobotnya ≥ 1 , yang berarti sedang terjadi kemacetan disisi-sisi tersebut sehingga tidak memungkinkan untuk dilewati.

Selanjutnya pembahasan dari kasus yang mungkin timbul dari permasalahan lalu lintas dalam graf pada gambar 4.1 adalah sebagai berikut.

Kasus :

Sedang terjadi kemacetan di Jalan Kamboja yang menuju titik E (sisi e_5), manakah jalur yang sebaiknya dipilih oleh polisi yang berada pada posisi A untuk mengalihkan kendaraan yang berada pada posisi A yang ingin melintasi sisi Jalan Kamboja dengan kemungkinan kendaraan ingin menuju salah satu titik di dalam graf tersebut?

Pembahasan :

Pertama, cari lintasan terpendek dari A ke semua titik selain A pada graf dengan menggunakan algoritma Dijkstra. Data yang di dapat adalah seperti pada tabel berikut.

Simpul awal adalah simpul A.

Tahap	Simpul terpilih	Lintasan	Larik	Simpul									
				A	B	C	D	E	F	G	H		
Inisial	-	-	S	0	0	0	0	0	0	0	0	0	0
			D	0	53	∞	∞	∞	77	∞	∞		
1	A	A	S	1	0	0	0	0	0	0	0	0	
			D	∞	53	∞	∞	∞	77	∞	∞		
2	B	A,B	S	1	1	0	0	0	0	0	0	0	
			D	∞	53	∞	∞	∞	77	∞	∞		
3	E	A,B,E	S	1	1	0	0	1	0	0	0	0	
			D	∞	53	∞	∞	99	77	∞	∞		
4	F	A,F	S	1	1	0	0	1	1	0	0	0	
			D	∞	53	∞	∞	99	77	∞	∞		
5	G	A,F,G	S	1	1	0	0	1	1	1	0	0	
			D	∞	53	∞	∞	99	77	131	∞		
6	C	A,B,E,C	S	1	1	1	0	1	1	1	0	0	
			D	∞	53	174	∞	99	77	131	∞		
7	D	A,B,E,C,D	S	1	1	1	1	1	1	1	0	0	
			D	∞	53	174	218	99	77	131	∞		
8	H	A,B,E,C,D,H	S	1	1	1	1	1	1	1	1	0	
			D	∞	53	174	218	99	77	131	267		

Tabel III Hasil pencarian lintasan terpendek dengan algoritma Dijkstra dari graf pada gambar 4.1 dengan simpul awal A

Sumber : Olahan penulis

Dari tabel tersebut telah diketahui seluruh lintasan terpendek dari titik A ke semua titik lain yaitu sebagai berikut.

1. Dari A ke B : melalui lintasan A,B
Melalui Jalan Mawar.
2. Dari A ke C : melalui lintasan A,B,E,C
Melalui Jalan Mawar, Jalan Begonia, dan Jalan Anggrek.
3. Dari A ke D : melalui lintasan A,B,E,C,D
Melalui Jalan Mawar, Jalan Begonia, Jalan Anggrek, dan Jalan Sedap Malam.
4. Dari A ke E : melalui lintasan A,B,E
Melalui Jalan Mawar dan Jalan Begonia.
5. Dari A ke F : melalui lintasan A,F
Melalui Jalan Melati.
6. Dari A ke G : melalui lintasan A,F,G
Melalui Jalan Melati dan Jalan Aster.
7. Dari A ke H : melalui lintasan A,B,E,C,D,H
8. Melalui Jalan Mawar, Jalan Begonia, Jalan Anggrek, Jalan Sedap Malam, dan Jalan Seruni.

Dari data diatas sekarang sudah dapat ditentukan kemana kendaraan harus dialihkan agar pengalihan tidak menambah kemacetan baru. Kendaraan yang ingin menuju titik B, C, D, E, dan H dapat dialihkan ke Jalan Mawar (sisi e_5), sementara kendaraan yang ingin menuju titik F dan G adapat dialihkan ke Jalan Melati (sisi e_{12}).

Pemilihan jalur pengalihan ini penting sebab pemilihan jalur pengalihan yang menambah kemacetan di jalur lain. Misalkan ada kendaraan X di titik A yang ingin menuju titik C melalui Jalan Kamboja (sisi e_5), namun karena Jalan Kamboja macet, polisi lalu lintas yang ada di titik A mengalihkan kendaraan ke Jalan Melati (sisi e_{12}), akibatnya kendaraan X tersebut harus melewati Jalan Melati (A ke F), kemudian ke Jalan Aster (F ke G), dan selanjutnya ke Jalan Alamanda (F ke C) karena itu satu-satunya jalur yang dapat ditempuh dari Jalan Melati untuk menuju titik C. Namun pada kenyataannya di Jalan Alamanda yang menuju titik C juga sedang terjadi macet, sehingga kendaraan X juga ikut terjebak macet dan memperparah kemacetan di Jalan Alamanda. Itu artinya pengalihan kemacetan yang dilakukan polisi di titik A tadi tidak berhasil dan justru memperparah kemacetan di ruas jalan yang lain. Oleh karena itulah pemilihan jalur alternatif yang tepat adalah penting.

V. KESIMPULAN

Kemacetan lalu lintas merupakan salah satu masalah yang menimbulkan banyak kerugian, karena itu saat kemacetan terjadi perlu dilakukan tindakan untuk mengurangi kemacetan. Salah satu cara mengurangi

kemacetan adalah dengan mengalihkan kendaraan ke jalur alternatif, yaitu jalur lain selain jalur yang sedang terjadi macet. Namun pengalihan ke jalur alternatif ini perlu diperhatikan sebab dapat terjadi dua kemungkinan yaitu mengurangi kemacetan dengan teralihkannya kendaraan ke jalur yang lebih lancar, atau justru memperparah kemacetan di ruas jalan lain atau bahkan membuat titik kemacetan baru. Untuk itu pemilihan jalur alternatif yang tepat untuk pengalihan jalur saat terjadi kemacetan sangat diperlukan. Hal yang perlu diperhatikan untuk menentukan jalur alternatif adalah kondisi ruas jalan lain atau volume kendaraan di ruas jalan lain yang mungkin menjadi jalur alternatif. Seperti yang telah dibahas dalam makalah ini, salah satu cara yang dapat digunakan dalam menentukan jalur alternatif yang tepat untuk mengurangi kemacetan adalah dengan mengaplikasikan teori graf, dengan graf yang digunakan adalah graf berarah yang merepresentasikan jalur lalu lintas di suatu wilayah dan memiliki bobot berupa perbandingan volume kendaraan yang sedang lewat dimasing-masing ruas jalan dengan volume maksimal kendaraan yang dapat melewati jalan tersebut pada suatu satuan waktu. Prinsip yang digunakan pada pencarian jalur alternatif ini adalah lintasan terpendek dan pencariannya menggunakan Algoritma Dijkstra.

VI. PENUTUP

Puji syukur saya panjatkan kepada Allah SWT karena atas rahmatNya saya diberikan kekuatan untuk menyelesaikan makalah ini. Saya juga mengucapkan terima kasih kepada orang tua yang selalu mendukung segala kegiatan yang saya lakukan, baik dengan bantuan moral maupun material. Selanjutnya saya juga mengucapkan terima kasih kepada Ibu Harlili dan Bapak Rinaldi Munir selaku dosen mata kuliah Matematika Diskrit, yang didalamnya juga memuat teori graf serta algoritma Dijkstra yang saya gunakan untuk menulis makalah ini. Terima kasih juga saya ucapkan kepada semua pihak yang membantu terselesaikannya makalah ini, baik secara langsung maupun tidak langsung, yang tidak dapat saya sebutkan satu persatu.

Penulis menyadari masih banyak kekurangan dalam makalah ini, karena itu kritik dan saran sangat disilakan.

REFERENCES

- [1] Small, Kenneth A. (2008). "Urban Transportation". Di David R. Henderson (ed.). *Concise Encyclopedia of Economics* (2nd ed.). Indianapolis: Library of Economics and Liberty. Diakses pada 9 Desember 2015.
- [2] <http://www.pu.go.id/isustrategis/view/24>. Diakses pada 9 Desember 2015.
- [3] Rosen, Kenneth H., *Global Edition Discrete Mathematics and Its Applications*, 7th edition. New York : McGraw-Hill, 2013.
- [4] Munir, Rinaldi, *Diktat Kuliah IF2120 Matematika Diskrit*. Bandung : Teknik Informatika Institut Teknologi Bandung, 2003.
- [5] Sniedovich, M. (2006). "Dijkstra's algorithm revisited: the dynamic programming connexion" (PDF). *Journal of Control and*

Cybernetics. Online version of the paper with interactive computational modules. Diakses pada 9 Desember 2015.

Link : <http://matwbn.icm.edu.pl/ksiazki/cc/cc35/cc3536.pdf>

- [6] Paul E. Black, "greedy algorithm", in *Dictionary of Algorithms and Data Structures* [online], Vreda Pieterse and Paul E. Black, eds. 2 February 2005. Diakses pada 9 Desember 2015.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 8 Desember 2015



Sri Umay Nur'aini Sholihah
(13514007)