

# Algoritma Dijkstra untuk Menentukan Rute Termurah dan Rute Tercepat pada Taksi Konvensional

Restu Wahyu Kartiko - 13516155<sup>1</sup>  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
<sup>1</sup>rwk03@itb.ac.id

**Abstract**—Taksi konvensional merupakan salah satu aspek yang ikut andil dalam perkembangan moda transportasi di dunia. Berbeda dengan mode transportasi online yang sudah menetapkan harga diawal, taxi konvensional menggunakan sistem argometer sebagai alat pengukur tarif yang harus dibayar. Besar nilai tarif pada argometer dipengaruhi oleh dua factor utama yaitu waktu tunggu dan jarak tempuh. Oleh karena itu, besarnya tarif sangat ditentukan oleh pemilihan rute. Pemilihan rute yang tepat tidak hanya menghemat dari segi tarif, namun juga waktu. Makalah ini akan membahas tentang cara pemilihan rute taksi yang tepat sesuai kebutuhan konsumen dengan menggunakan algoritma Dijkstra.

**Keywords**— Algoritma Dijkstra, graf, taksi, rute, shortest path.

## I. PENDAHULUAN

Moda transportasi adalah alat yang digunakan untuk membantu manusia untuk berpindah dari suatu tempat ke tempat yang lain dengan waktu yang lebih cepat. Transportasi terbagi menjadi 3 sektor utama yaitu transportasi darat, laut, dan udara. Ada beragam jenis transportasi yang ada, baik itu umum maupun pribadi. Di kota besar seperti Jakarta, biasanya mempunyai moda transportasi yang lebih beragam. Misalnya seperti kereta api, busway, ojek, taksi, dan lain-lain.

Banyaknya jumlah penduduk membuat moda transportasi umum bersama seperti busway dan angkot, menjadi ramai pengguna. Hal tersebut terkadang menimbulkan situasi yang kurang nyaman dalam menggunakan transportasi umum karena harus berdesak-desakan dengan masyarakat lain yang juga ingin menggunakannya. Masyarakat menginginkan kenyamanan dalam menggunakan transportasi publik. Salah satu transportasi publik yang dianggap nyaman oleh masyarakat adalah taksi. Walaupun harga yang harus dibayar relatif lebih besar dibanding dengan moda transportasi umum bersama, namun kenyamanan yang diberikan sepadan. Terlebih lagi masyarakat tidak perlu mengantri panjang dan berdesak-desakan dan tidak dibatasi oleh rute-rute tertentu sehingga perjalanan menjadi lebih dinamis.

Layanan yang lebih baik dibanding dengan angkutan umum seperti angkot atau bis kota membuat tarif taksi cenderung lebih mahal. Sebagai konsumen, kita tentu menginginkan suatu cara agar tarif yang kita bayar menjadi seminimal mungkin. Terlebih lagi, terkadang ada pengemudi taksi yang kurang mengerti rute tujuan sehingga membuat waktu tempuh menjadi tidak optimal, padahal ada pertemuan atau kelas kuliah yang harus kita hadiri

atau saat jam-jam padat sehingga kemacetan ada dimana-mana.

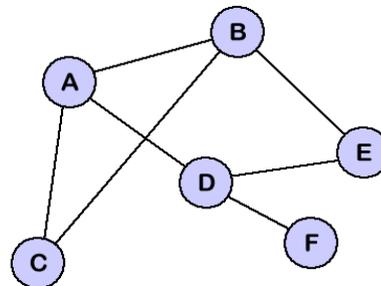
Dalam memilih rute taksi, tidak selalu rute terpendek merupakan rute tercepat. Dalam kondisi macet misalnya, mengambil jalan yang relatif lancar bisa membuat perjalanan menjadi lebih cepat walaupun jarak yang ditempuh lebih panjang. Begitu juga dengan tarif taksi, pemilihan rute terpendek belum tentu dapat memangkas tarif taksi. Maka, sebagai pengemudi ataupun pengguna taksi, kita hendaknya bisa menentukan mana jalur tercepat atau termurah yang harus kita lalui sesuai dengan kondisi kita.

## II. TEORI DASAR

Graf adalah kumpulan simpul(nodes) yang dihubungkan dengan satu sama lain melalui sisi/busur. Graf telah diaplikasikan dalam banyak hal seperti sistem penjadwalan ruang kelas, pembuatan drainase kota, penjaluran distribusi produk, hingga representasi senyawa kimia. Beberapa masalah menjadi lebih mudah diselesaikan dengan pemodelan graf.

### 2.1. Definisi Graf

Graf  $G$  didefinisikan sebagai pasangan himpunan  $(V, E)$ , ditulis dengan  $G = (V, E)$ , yang didalam hal ini  $V = \{v_1, v_2, v_3, \dots, v_n\}$  adalah himpunan tidak-kosong dari simpul-simpul dan  $E = \{e_1, e_2, e_3, \dots, e_m\}$  adalah simpunan sisi yang menghubungkan sepasang simpul yang dapat ditulis sebagai  $e = (v_1, v_2)[1]$ .



Gambar 2.1 : Graf dengan 6 simpul dan 7 sisi[1]

Sumber: <https://courses.cs.vt.edu/csonline/DataStructures/Lessons/Graphs/index.html>

### 2.2. Jenis-jenis Graf

Sisi ganda adalah sisi yang menghubungkannya dua buah simpul yang sama. Gelang adalah sisi yang memiliki simpul awal dan akhir yang sama. Berdasarkan ada tidaknya gelang atau sisi ganda, graf dapat digolongkan menjadi dua jenis yaitu[1]:

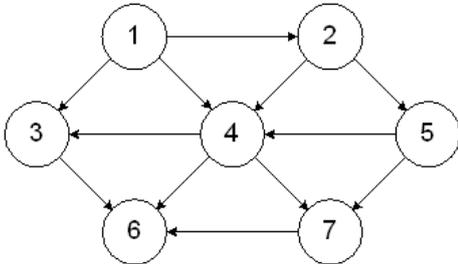
1. Graf sederhana (*simple graph*)  
Yaitu graf yang tidak mempunyai gelang maupun sisi ganda.
2. Graf tak-sederhana (*unsimple-graph*)  
Yaitu graf yang mempunyai sisi ganda atau gelang



Gambar 2.2 contoh graf tak sederhana dengan gelang  
Sumber: [mathworld.wolfram.com/SimpleGraph.html](http://mathworld.wolfram.com/SimpleGraph.html)

Berdasarkan orientasi arah pada sisi, graf dapat digolongkan menjadi dua jenis yaitu [1]:

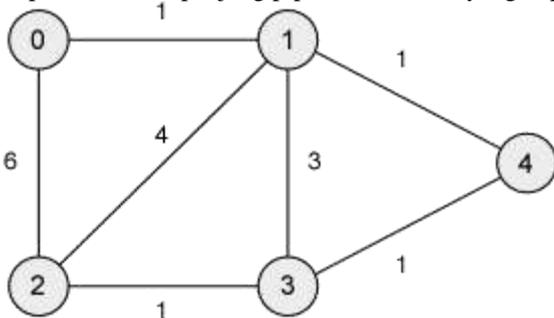
1. Graf tak-berarah (*undirected graph*)  
Yaitu graf yang sisinya tidak memiliki orientasi arah. Karena tidak memiliki orientasi arah, maka sisi  $(u, v) = (v, u)$ .
2. Graf berarah (*directed graph* atau *digraph*)  
Yaitu graf yang sisinya memiliki orientasi arah. Sisi pada graf ini direpresentasikan dengan  $e = (u, v)$  dengan  $u$  adalah simpul awal dan  $v$  adalah simpul tujuan. Sehingga  $(u, v) \neq (v, u)$ .



Gambar 2.3 contoh graf berarah  
Sumber: <http://faculty.cs.niu.edu/~freedman/340/340notes/340graph.htm>

### 2.3. Graf Berbobot (*Weighted graph*)

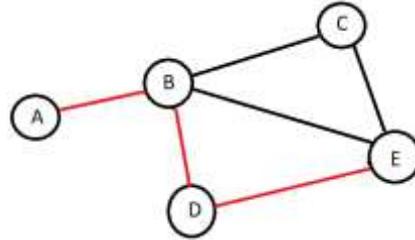
Graf berbobot adalah graf yang setiap sisinya mempunyai nilai/bobot tertentu. Biasanya, bobot tiap sisi langsung dituliskan di sisi tersebut. Nilai yang ada di setiap sisi dapat mewakili apapun tergantung pada masalah yang dimodelkan. Dalam pembangunan sistem pipa air, simpul dapat merepresentasikan bangunan, dan bobot pada sisinya dapat merepresentasikan panjang pipa dan debit air yang dapat lewat.



Gambar 2.4 contoh graf berbobot.  
Sumber: <http://pages.cpsc.ucalgary.ca/~jacobs/Courses/cpsc331/F08/tutorials/tutorial14.html>

### 2.4. Lintasan

Lintasan Merupakan barisan berselang-seling simpul dan sisi yang terbentuk  $v_0, e_1, v_1, e_2, v_2, \dots, v_{n-1}, e_n, v_n$  sedemikian hingga  $e_1 = (v_0, v_1)$ ,  $e_2 = (v_1, v_2)$  dst adalah sisi dari graf  $G[1]$ .



Gambar 2.5 sisi warna merah adalah contoh lintasan pada graf.  
Sumber : buatan penulis  
Lintasan  $A \rightarrow B \rightarrow D \rightarrow E$  merupakan salah satu lintasan yang dapat digunakan dari simpul A ke simpul E.

### 2.5. Lintasan Terpendek Satu Sumber

Diberikan sebuah graf  $G$  berbobot dan simpul awal  $S$ , berapakah lintasan terpendek dari  $S$  ke semua simpul yang ada?. Hal tersebut merupakan masalah lintasan terpendek satu sumber (*single source shortest path / SSSP*) yang merupakan permasalahan klasik di dunia graf. Ada banyak masalah yang bisa dimodelkan dari SSSP ini. Contoh paling sederhana adalah kita dapat memodelkan sebuah kota menjadi sebuah graf dimana simpulnya mewakili persimpangan jalan dan sisinya mewakili jalan. Waktu yang dibutuhkan untuk melalui suatu jalan adalah bobot dari sisi. Berapa waktu paling minimum untuk berpindah dari suatu simpul ke simpul lain? [4].

### 2.6. Algoritma Dijkstra

Algoritma Dijkstra dikemukakan oleh Edsger Wybe Dijkstra. Algoritma ini dipublikasikan pada tahun 1959 jurnal *Numerische Mathematik* yang berjudul "A Note on Two Problems in Connexion with Graphs". Dijkstra merupakan salah satu dari banyak algoritma yang dapat memecahkan masalah optimasi pencarian lintasan terpendek. Algoritma ini menentukan bobot minimum dari suatu simpul ke semua simpul yang ada pada graf dengan pendekatan *greedy*. Bobot yang ada pada graf harus bernilai positif, karena jika ada bobot yang bernilai negatif, maka akan tak-hingga solusi yang mungkin.

Secara garis besar, urutan logika algoritma Dijkstra pada graf berbobot adalah sebagai berikut [5]:

1. Inisialisasi nilai simpul awal dengan 0 dan nilai tak hingga untuk simpul lain
2. Set semua simpul sebagai simpul yang belum dikunjungi dan set simpul awal sebagai "Simpul keberangkatan"
3. Beri tanda *false* untuk semua simpul yang belum pernah dikunjungi dan *true* untuk simpul keberangkatan (awal).
4. Dari simpul keberangkatan, pertimbangkan simpul tetangga yang belum dikunjungi dan hitung jaraknya dari simpul keberangkatan. Jika jaraknya lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya) hapus data lama, simpan ulang data jarak dengan jarak yang baru
5. Setelah selesai mempertimbangkan setiap jarak terhadap simpul tetangga, tandai simpul yang telah dikunjungi dengan *true* (sudah dikunjungi). Simpul yang telah dikunjungi tidak akan dikunjungi lagi, jarak yang

disimpan adalah jarak terakhir dan yang paling minimal bobotnya.

- Atur Simpul yang belum dikunjungi dengan jarak terkecil(dari simpul keberangkatan) sebagai simpul keberangkatan yang baru. Ulangi langkah 4 dan 6 hingga semua simpul bertanda *true*(sudah dikunjungi)

### Psuedocode algoritma Dijkstra

**function** Dijkstra (Graph, source):

```

for each vertex v in Graph :
    dist[v] := infinity ;    (inisialisasi nilai jarak)
    previous[v] := undefined;
end for

dist[source] := 0;
Q := the set of all nodes in Graph;
While Q is not empty :
    u := vertex in Q with smallest distance in dist[];
    remove u from Q;

    if dist[u] = infinity:
        break;
    end if

    for each neighbor v of u:
        alt := dist[u] + dist_between(u,v);
        if alt < dist[v] :
            dist[v] := alt;
            previous[v] := u;
            decrease-key v in Q;
        end if
    end for
end while
return dist;

```

## III. TAKSI

### 3.1. Definisi Taksi

Taksi adalah sebuah moda transportasi umum yang berlisensi untuk mengantarkan penumpang dari tempat asal ke tempat yang dituju dengan imbalan sesuai dengan yang tertera di argometer.

### 3.2. Argometer

Argometer adalah alat ukur banyaknya uang yang harus dibayar penumpang taksi berdasarkan jarak tempuh dan waktu penggunaan. Argometer menggunakan puls listrik yang menempel pada transmisi taksi untuk mengukur jarak dan waktu tempuh. Sensor mengirim puls listrik ke meteran setiap kali taksi menempuh jarak tertentu. Dalam meteran tersebut juga terdapat timer yang mengirim informasi saat sejumlah waktu berlalu tanpa adanya pergereakan/berhenti.



Gambar 3.1 Argometer saat taksi mengangkut penumpang dan berjalan

Sumber: <http://www.alamy.com/stock-photo/taxi-meter.html>



Gambar 3.2 Argometer saat taksi mengangkut penumpang, namun dalam keadaan berhenti

Sumber: <https://play.google.com/store/apps/details?id=com.planetco.ops.android.taximeter>

### 3.3. Tarif Batas Atas dan Batas Bawah Taksi

Ada 3 macam jenis tarif pada taksi konvensional yang yaitu[2]:

- Tarif buka pintu : Rp 6.500  
Tarif buka pintu akan berlaku hingga 1km pertama atau 5 menit pertama
- Tarif per km : Rp 3.500  
Tarif per km akan berlaku untuk kilometer ke dua dan seterusnya
- Tarif waktu tunggu : Rp 42.000 per jam atau Rp 700 per menit

Melihat tabel diatas, terdapat 2 macam argo yaitu argo berjalan(per km) dan argo waktu(per menit). Ada Batasan kecepatan dimana argo berjalan yang dipakai dan juga ada batasan dimana argo waktu yang dipakai.

Batas yang dipakai adalah sebagai berikut[3]:

$$\text{KPM} * \text{APK} = \text{APM}$$

$$\text{KPM} = \text{APM} / \text{APK}$$

$$\text{BATAS\_KECEPATAN} = 60 \text{ MENIT} * \text{APM} / \text{APK}$$

Keterangan:

KPM = Kecepatan per menit

APK = Argo per Km

APM = Argo per menit

$$\text{BATAS\_KECEPATAN} = 60 * 700 / 3500 = 12 \text{ KM/JAM}$$

Jadi, jika taksi melaju dengan kecepatan dibawah 12 km/jam, maka akan dipakai argo waktu. Sebaliknya, jika taksi melaju dengan kecepatan 12 km/jam atau lebih, akan dikenakan tarif berjalan.

### 3.4. Faktor Luar yang Mempengaruhi Tarif Taksi

Tarif taksi dipengaruhi oleh dua parameter utama yaitu waktu tunggu dan jarak tempuh. Waktu tunggu dan jarak tempuh

sendiri dipengaruhi oleh beberapa faktor eksternal seperti kondisi lalu lintas, adanya pemberhentian, dan pemilihan rute.

Saat kondisi lalu lintas normal, maka tarif taksi murni hanya ditentukan dari tarif per km. Namun saat kondisi macet, taksi menjadi sering berhenti. Maka, selain tarif per km, total tarif taksi juga dipengaruhi tarif waktu tunggu saat taksi berhenti.

Terkadang, kita meminta untuk berhenti di minimarket untuk membeli suatu barang. Pada saat menunggu, argo taksi terus berjalan sehingga tarifnya bertambah.

Pemilihan rute yang tepat dapat meminimalkan nilai nominal tarif yang ada pada taksi. Pemilihan rute tersebut akan dijelaskan lebih lanjut pada bab berikutnya.

#### IV. ANALISIS RUTE TERCEPAT VS RUTE TERMURAH

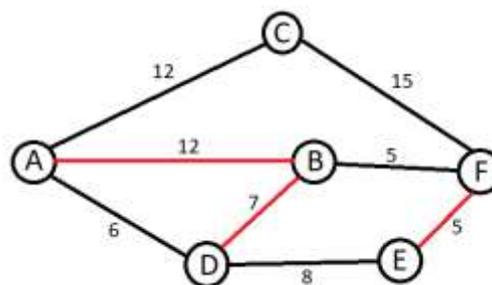
Pemilihan rute taksi bergantung pada kondisi penumpang. Jika dalam kondisi tergesa-gesa, penumpang cenderung memilih rute tercepat walaupun rutenya menjadi lebih jauh. Dalam keadaan normal, penumpang cenderung memilih rute termurah.

Waktu tempuh suatu perjalanan dipengaruhi oleh beberapa hal seperti kondisi jalan yang rusak, adanya kemacetan, kecepatan kendaraan, dan jarak yang ditempuh. Menurut data transportasi, di Jakarta, kecepatan kendaraan hanya berkisar 5 km/jam saat macet. Tentunya angkanya tidak akan sebesar itu dikota lain.

Pada analisis ini, rute perjalanan seseorang dengan menggunakan taksi akan dimodelkan menjadi sebuah graf berbobot dan tidak berarah. Dengan simpul melambangkan sebuah persimpangan jalan, kecuali dua simpul dengan simbol A yang menandakan titik keberangkatan kita, dan F yang menandakan lokasi tujuan. Graf tidak berarah menunjukkan bahwa sisi tersebut mewakili jalan dua arah. Sementara untuk bobot pada tiap sisi akan dibagi menjadi dua jenis, yaitu bobot yang merepresentasikan waktu tempuh dari suatu simpul ke simpul lain, yang kedua adalah bobot yang merepresentasikan tarif tempuh dari suatu simpul ke simpul lain. Sisi graf akan diberi dua warna yaitu merah yang melambangkan jalan sedang macet dan hitam jika kondisi lalu lintas normal.

Misalkan kita ingin menggunakan jasa taksi dari lokasi awal A ke lokasi tujuan F. Berikut adalah data hubungan simpul dan sisi yang ada:

Jalan(sisi)	Jarak(bobot)	Kondisi Jalan
A,B	12	Macet
A,C	12	Normal
A,D	6	Normal
B,F	5	Normal
B,D	7	Macet
C,F	15	Normal
D,E	8	Normal
E,F	5	Macet



Gambar 4.1 model graf dari data yang diberikan dengan bobot merepresentasikan jarak  
Sumber : buatan penulis

Jika dilihat secara sekilas, rute yang akan dipilih oleh supir taksi adalah A→B→F dengan total jarak yang minimum yaitu hanya 17 km. Namun, dengan beberapa pertimbangan yang akan disampaikan, terdapat rute yang lebih efisien dari segi harga, ataupun dari segi waktu.

Dalam memodelkan jalur kali ini, penulis akan membuat beberapa asumsi:

1. Dalam keadaan lancar, kecepatan kendaraan adalah 40 km/jam. Sementara dalam keadaan macet, kecepatan kendaraan adalah 6 km/jam mengingat kemacetan di Jakarta membuat laju kendaraan hanya 5 km/jam
2. Tarif taksi yang berlaku adalah tarif seperti yang sudah dijelaskan di poin 3.3

Dari data diatas, kita akan mencari rute yang tercepat(waktu) dan rute dengan tarif termurah.

##### 4.1. Rute Tercepat

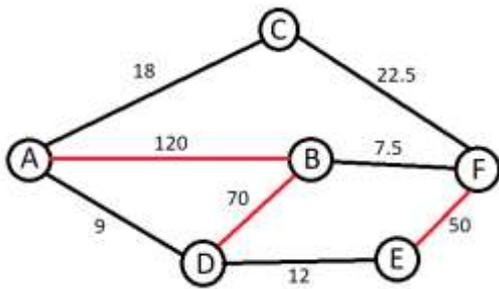
Dalam mencari rute tercepat, kita akan mengubah jenis bobot pada graf awal. Pada data awal, bobot pada sisi merepresentasikan jarak dari suatu simpul ke simpul lain. Disini kita akan merubahnya menjadi representasi dari waktu dari suatu simpul ke simpul tetangganya.

Waktu yang dibutuhkan untuk menelusuri suatu jalan bergantung pada kondisi jalan tersebut. Ada dua cara untuk menentukan waktu tempuh yaitu:

1. Jika jalan macet, maka kita dapat mencari waktu tempuhnya dengan rumus  
Waktu = (jarak / 6) x 60 menit  
(6 adalah kecepatan saat macet)
2. Jika kondisi jalan normal, kita dapat mencari waktu tempuhnya dengan rumu  
Waktu = (jarak / 40) x 60 menit  
(40 adalah kecepatan saat jalan normal)

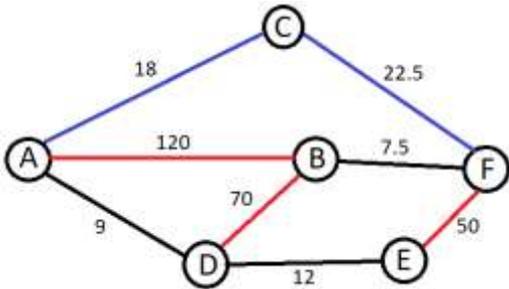
Dengan aturan diatas, maka data tabel berubah menjadi seperti berikut.

Jalan(sisi)	Waktu(menit)	Kondisi Jalan
A,B	120	Macet
A,C	18	Normal
A,D	9	Normal
B,F	7.5	Normal
B,D	70	Macet
C,F	22.5	Normal
D,E	12	Normal
E,F	50	Macet



Gambar 4.2 model graf dengan bobot sisi merepresentasikan waktu  
 Sumber : buatan penulis

Dengan menerapkan algoritman Dijkstra, kita dapat menentukan waktu tercepat dari tempat asal(A) ke tempat tujuan(F).



Gambar 4.3 rute tercepat menggunakan algoritma Dijkstra  
 Sumber : buatan penulis

Graf diatas adalah graf setelah algoritma Dijkstra dijalankan. Sisi dengan warna biru menandakan rute hasil dari algoritman Dijkstra. Rute tercepat darim A ke F adalah  $A \rightarrow C \rightarrow F$  dengan total jarak tempuh sebesar  $18 + 22.5 = 40.5$  menit. Total jarak yang ditempuh dengan rute tersebut adalah 27 km. Karena rute yang dilewati adalah rute normal, maka perhitungan tarif akan menggunakan tarif per km.

$$\text{Tarif} = \text{jarak} * 3500$$

$$\text{Tarif} = 27 * 3500 = \text{Rp } 94.500$$

#### 4.2 Rute Termurah

Dalam mencari rute termurah, kita akan mengubah jenis bobotnya dari data awal yang merepresentasikan jarak menjadi representasi dari tarif taksi.

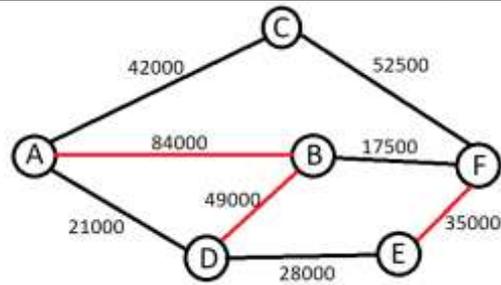
Total tarif taksi yang dibutuhkan untuk menelusuri suatu jalan bergantung pada kondisi macet tidaknya jalan tersebut. Berikut cara perhitungannya :

1. Jika kondisi jalan macet, asumsi laju taksi adalah 6 km/jam. Karena laju taksi berada dibawah batas kecepatan minimum, maka berlaku tarif waktu tunggu.  
 $\text{Tarif} = \text{waktu} \times 700$
2. Jika kondisi jalan normal, asumsi laju taksi adalah 40 km/jam. Karena berada diatas batas kecepatan minimum, maka berlaku tarif jarak per km.  
 $\text{Tarif} = \text{jarak} \times 3500$

Dengan aturan diatas, maka data tabel berubah menjadi seperti berikut

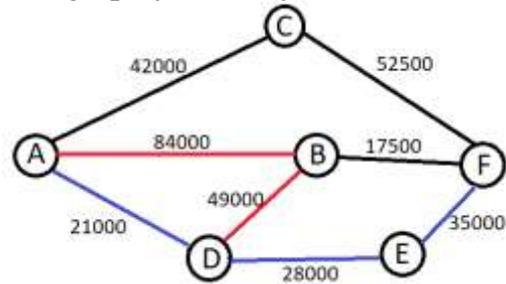
Jalan(sisi)	Tarif (Rupiah)	Kondisi Jalan
A,B	84000	Macet
A,C	42000	Normal
A,D	21000	Normal
B,F	17500	Normal
B,D	49000	Macet

C,F	52500	Normal
D,E	28000	Normal
E,F	35000	Macet



Gambar 4.4 model graf dengan bobot sisi merepresentasikan tarif taksi  
 Sumber : buatan penulis

Dengan algoritma Dijkstra, kita akan menentukan rute termurah yang dapat kita tempuh dari A ke F.



Gambar 4.5 rute termurah dengan algoritman Dijkstra  
 Sumber : buatan penulis

Graf diatas adalah representasi graf setelah algoritma Dijkstra dijalankan. Sisi yang berwarna biru menandakan rute Dijkstra yang dalam hal ini berarti rute dengan tarif termurah dari A ke F. Rute tersebut adalah  $A \rightarrow D \rightarrow E \rightarrow F$  dengan total biaya  $21000 + 28000 + 35000 = \text{Rp } 84000$ . Total jarak yang ditempuh dengan rute tersebut adalah 19 km. Sementara total waktu yang dibutuhkan adalah 71 menit.

Dari hasil analisis data diatas, pemilihan rute taksi sangat bergantung pada keinginan seseorang saat sedang menggunakan taksi. Jika kita ingin perjalanan yang cepat, kita dapat memilih rute dari  $A \rightarrow C \rightarrow F$  dengan total waktu yang paling singkat dibandingkan rute lain yaitu hanya 40.5 menit. Pada kondisi normal, seseorang akan memilih rute  $A \rightarrow D \rightarrow E \rightarrow F$  yang memiliki tarif paling murah disbanding rute lain yaitu sebesar Rp 84000.

Dari segi waktu tempuh, rute  $A \rightarrow C \rightarrow F$  memiliki waktu tempuh tersingkat. Namun dari segi tarif, rute  $A \rightarrow D \rightarrow E \rightarrow F$  memiliki tarif termurah. Pada kenyataannya, rute  $A \rightarrow C \rightarrow F$  memiliki rute yang lebih panjang jika dibandingkan dengan rute lain seperti  $A \rightarrow B \rightarrow F$  maupun  $A \rightarrow D \rightarrow E \rightarrow F$ . Akan tetapi rute  $A \rightarrow C \rightarrow F$  justru menjadi rute tercepat. Hal ini membuktikan bahwa panjang rute tidak selalu mengakibatkan waktu tempuh menjadi lebih lama. Inilah pentingnya strategi pemilihan rute agar waktu dan uang yang penumpang keluarkan menjadi se-efisien mungkin.

#### V. KESIMPULAN

Teori graf memiliki banyak penerapan didunia nyata baik yang langsung maupun tak langsung. Contoh penerapann secara langsung yang setiap orang dapat lakukan adalah memodelkan

rute taksi. Pengguna taksi pasti menginginkan rute perjalanan yang optimal baik dari segi waktu maupun biaya. Namun, seringkali rute yang dipilih adalah rute dengan jarak terpendek dengan maksud agar bisa sampai lebih cepat dan murah. Padahal, rute yang pendek tidak menjamin bahwa waktu yang dibutuhkan menjadi lebih singkat seperti hal yang telah dijabarkan sebelumnya. Pemilihan rute yang tepat dapat mengoptimalkan waktu dan biaya yang akan dikeluarkan.

Pemodelan rute taksi kali ini menggunakan graf berbobot dan tak-berarah. Simpul pada graf mewakili persimpangan jalan kecuali simpul awal keberangkatan dan tujuan. Sisi graf mewakili jalan antar persimpangan. Ada dua macam sisi yaitu sisi warna merah yang menandakan ada kemacetan di jalan tersebut, dan sisi warna biru yang menandakan kondisi lalu lintas yang normal. Bobot pada sisi mewakili tiga hal yaitu jarak untuk mencari rute terpendek, waktu tempuh untuk mencari rute tersingkat, dan total tarif untuk mencari rute termurah.

Algoritma Dijkstra berguna untuk menentukan rute mana yang paling optimal dari segi waktu atau dari segi tarif. Penentuan rute ini sangat penting untuk memangkas waktu perjalanan dan meminimalkan total tarif yang harus dibayar.

## VI. UCAPAN TERIMA KASIH

Penulis mengucapkan terimakasih kepada Allah SWT atas berkat dan rahmat-Nya, penulis dapat menyelesaikan makalah ini dengan sebaik mungkin. Penulis juga ingin mengucapkan terima kasih kepada kedua orang tua penulis yang telah memberikan dukungan kepada penulis dalam menjalankan perkuliahan. Selain itu penulis mengucapkan terimakasih kepada Bapak Rinaldi Munir dan Ibu Harlili selaku dosen mata kuliah Matematika Diskrit untuk semua ilmu yang mereka berikan yang sangat berguna dalam pembuatan makalah ini. Terakhir penulis ingin mengucapkan kepada semua pihak yang telah memberi dukungan dan inspirasi dalam membuat makalah ini.

## REFERENCES

- [1] Munir, Rinaldi, "Matematika Diskrit edisi keempat", Bandung : Penerbit Informatika, 2010. Hlm 356-376.
- [2] <https://harga.web.id/info-tarif-taksi-blue-bird-tahun-2017.info>. Diakses pada 2 Desember 2017.
- [3] [https://www.kompasiana.com/abietanpan/mengira-tarif-taksi\\_552c2a186ea834517f8b4580](https://www.kompasiana.com/abietanpan/mengira-tarif-taksi_552c2a186ea834517f8b4580). Diakses pada 2 Desember 2017.
- [4] Halim, Steven, & Felix, Steven, "Competitive Programming 3 : The New Lower Bound of Programming Contests", Singapore : lulu 2013. hlm 146-148.
- [5] <http://mti.binus.ac.id/2017/11/28/algorithm-dijkstra/>. Diakses pada 3 Desember 2017.

## 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, 3 Desember 2017



Restu Wahyu Kartiko 13516155