Aplikasi *Travelling Salesman Problem* dalam Penentuan Rute DAMRI Dipatiukur - Jatinangor

Ahmad Nadil - 13521024[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]*13521024@std.stei.itb.ac.id*

***Abstract*—Perkembangan ilmu pengetahuan serta teknologi telah membantu kehidupan manusia untuk dapat melakukan segala sesuatu menjadi lebih cepat dan efisien. Salah satunya adalah dalam penentuan rute perjalanan. Rute perjalanan yang optimal dapat membantu manusia agar dapat melakukan mobilisasi dari suatu tempat ke tempat lainnya dengan lebih efisien, baik dari segi waktu, tenaga, dan biaya. Pada makalah ini akan dibahas sebuah permasalahan graf, yaitu *Travelling Salesman Problem* (TSP), dengan algoritma *Branch and Bound* untuk mencari rute perjalanan damri paling efisien agar dapat mendatangi seluruh titik penjemputan tepat satu kali dan kembali ke titik awal.**

***Keywords*— Branch and Bound, Graf, Rute Terpendek, Travelling Salesman Problem.**

I. PENDAHULUAN

Djawatan Angkoetan Motor Repoeblik Indonesia (EYD : Jawatan Angkutan Motor Republik Indonesia) atau yang dikenal dengan DAMRI adalah salah moda transportasi umum yang kerap digunakan masyarakat. DAMRI menjadi salah satu pilihan masyarakat karena dapat memuat banyak penumpang, sehingga bisa digunakan untuk pergi berombongan, serta harganya yang cukup terjangkau. Salah satu rute DAMRI yang kerap memiliki banyak penumpang adalah DAMRI jurusan Dipatiukur – Jatinangor, dikarenakan kedua kota ini adalah kota mahasiswa yang saling terhubung satu sama lain. Contohnya adalah mahasiswa ITB Kampus Jatinangor yang kerap menggunakan DAMRI ini untuk melaksanakan kegiatan di ITB Kamus Ganesha.

Rute sebuah DAMRI tidak hanya dari titik A ke titik B, akan tetapi sebelum sampai di titik B, DAMRI tersebut akan melewati berbagai titik terlebih dahulu yang terdapat diantara kedua titik tersebut. Sehingga, perjalanan sebuah DAMRI bisa cukup panjang. Pada makalah ini akan dibahas bagaimana kita dapat menentukan rute paling efisien untuk dilalui sebuah DAMRI.

Travelling Salesman Problem atau yang lebih dikenal dengan TSP merupakan salah satu permasalahan yang sering dibahas dalam ilmu computer dan matematika. Permasalahan ini mencakup tentang bagaimana kita dapat menemukan rute terpendek yang dapat mengenjungi semua titik tujuan (diimplementasikan dalam bentukk graf) dan kembali ke titik awal dengan syarat hanya mengunjungi tiap titik tepat satu kali.

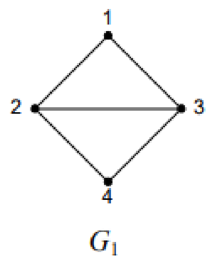
TSP kerap diterapkan di berbagai bidang kehidupan, misalnya permasalah logistik, pengiriman barang, serta perancanaan rute angkutan umum. Dengan menemukan solusi yang optimal untuk TSP ini, dapat membantu kita dalam mengurangi waktu serta biaya yang perlu dikeluarkan, serta dapat meningkatkan efisiensi serta keefektifan dalam berbagai aktivitas yang menggunakan rute perjalanan.

Makalah ini juga akan membahas tentang permasalahan TSP secara lebih detail pada kasus penentuan rute DAMRI serta metode yang dapat digunakan untuk menyelesaikan permasalah ini.

II. DASAR TEORI

A. *Definisi Graf*

Graf merupakan salah satu struktur data non linear yang mengandung *vertex* (simpul) dan *edge* (sisi). Simpul menjadi sebuah representasi objek pada sebuah graf, biasanya merupakan bilangan integer, sedangkan sisi merupakan representasi hubungan antara sebuah simpul ke simpul lainnya. Graf dapat ditulis dalam G = (V,E). V merupakan himpunan tidak-kosong dari simpul-simpul, dapat dinyatakan dengan V = $\{v_1, v_2, \dots, v_n\}$. Sedangkan E merupakan himpunan sisi yang menghubungkan sepasang simpul, dapat dinyatakan dengan E = $\{e_1, e_2, \dots, e_n\}$.



Gambar 1. Graf Sederhana
Sumber: Graf Bagian 1 oleh Rinaldi Munir

Diatas merupakan salah satu contoh dari sebuah graf. Graf tersebut dapat ditulis dalam G = (V,E), dengan V = {1,2,3,4} dan E = { (1,2), (1,3), (2,3), (2,4), (3,4) }.

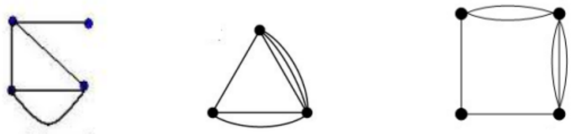
B. *Jenis Graf*

Berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graf, maka graf digolongkan menjadi dua jenis:

1. Graf sederhana (*simple graph*)

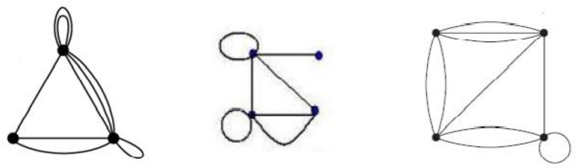
Graf yang tidak mengandung gelang maupun sisi ganda dinamakan graf sederhana. Contoh dari graf sederhana dapat dilihat pada Gambar 1.

2. Graf tak-sederhana (*unsimple-graph*)
   Graf yang mengandung sisi ganda atau gelang dinamakan graf-tak sederhana. Graf tak-sederhana dibedakan menjadi dua, yaitu:
   a. Graf ganda (*multi-graph*), yaitu graf yang mengandung sisi ganda.



Gambar 2. Graf Ganda
Sumber: Graf Bagian 1 oleh Rinaldi Munir

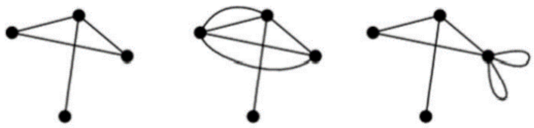
   b. Graf semu (*pseudo-graph*),yaitu graf yang mengandung sisi gelang (sisi yang mengarah pada simpul asalnya) .



Gambar 3. Graf Semu
Sumber: Graf Bagian 1 oleh Rinaldi Munir

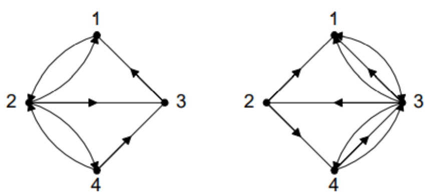
Berdasarkan orientasi arah pada sisi, graf dibedakan atas 2 jenis:

1. Graf tak-berarah (*undirected graph*)
   Graf yang sisinya tidak mempunyai orientasi arah disebut graf tak-berarah.



Gambar 4. Graf tak-berarah
Sumber: Graf Bagian 1 oleh Rinaldi Munir

2. Graf berarah (*directed graph atau digraph*)
   Graf yang setiap sisinya diberikan orientasi arah disebut sebagai graf berarah.



Gambar 5. Graf Berarah
Sumber: Graf Bagian 1 oleh Rinaldi Munir

C. Terminologi Graf

1. Ketetanggaan (*Adjacent*)
   Simpul yang bertetanggaan adalah simpul yang saling memiliki hubungan langsung.

2. Beririsan (*Incidency*)
   Sisi dikatakan beririsan jika simpul v_j dan v_k terdapat pada sisi $e = (v_j, v_k)$.

3. Simpul Terpencil (*Isolated Vertex*)
   Sebuah simpul dikatakan terpencil bila tidak mempunyai sisi yang bersisian dengannya.

4. Graf Kosong (*null graph* atau *empty graph*)
   Graf yang himpunan sisinya merupakan himpunan kosong (N_n).

5. Derajat (*Degree*)
   Derajat merupakan jumlah sisi yang bersisian dengan simpul tersebut.

6. Lintasan (*Path*)
   Lintasan yang panjangnya *n* dari simpul awal v_0 ke simpul tujuan v_n di dalam graf G ialah barisan berselang-seling simpul-simpul dan sisi-sisi yang berbentuk $v_o, e_1, v_1, e_2, v_2, v_{n-1}, e_n, v_n$ sedemikian sehingga $e_1 = (v_0, v_1), e_2 = (v_1, v_2), \dots, e_n = (v_{n-1}, v_n)$ adalah sisi-sisi dari graf G.

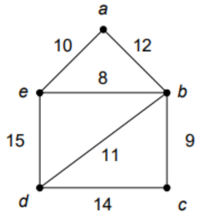
7. Siklus (*Cycle*) atau Sirkuit (*Circuit*)
   Lintasan yang berawal dan berakir pada simpul yang sama.

8. Keterhubungan (*Connected*)
   Dua buah simpul v_1 dan simpul v_2 terhubung jika terdapat lintasan dari v_1 ke v_2.
   Graf terhubung adalah graf yang memiliki simpul-simpul yang memiliki keterhubungan.

9. *Cut-set*
   Cut-set dari sebuah graf terhubung adalah himpunan sisi yang bila dibuang dari G menyebabkan G tidak terhubung.

10. Upagraf Merentang (*Spanning Subgraph*)
    Upagraf $G_1 = (V_1, E_1)$ dari $G = (V, E)$ dikatakan Upagraf Rentang jika $V_1 = V$ (yaitu G_1 mengandung semua simpul dari G.

11. Graf Berbobot (*Weighted Graph*)
    Merupakan graf yang setiap sisinya diberi sebuah nilai atau bobot.



Gambar 6. Graf Berbobot
Sumber: Graf Bagian 1 oleh Rinaldi Munir

12. Graf Lengkap (*Complete Graph*)
    Graf sederhana yang setiap simpulnya mempunyai sisi ke semua simpul lainnya. Sisi pada graf lengkap yang memiliki n buah simpul adalah $\frac{n(n-1)}{2}$.

13. Graf Lingkaran
    Graf sederhana yang setiap simpulnya berderajat dua, dilambangkan dengan C_n.

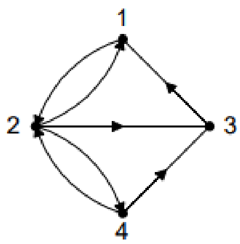
14. Graf Teratur (*Regular Graphs*)
    Merupakan graf yang setiap simpunya mempunyai derajat yang sama. Apabila derajat setiap simpul adalah r, maka jumlah sisi pada graf tersebut adalah $\frac{nr}{2}$.

D. *Representasi Graf*

Terdapat beberapa cara dalam merepresentasikan sebuah graf. Namun, dalam makalah ini akan dibahas representasi sebuah graf dalam bentuk matriks.

Matriks Ketetanggaan (*adjacency matrix*)
$A = [a_{ij}]$

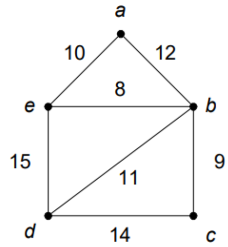
- Bernilai 1 jika simpul i dan j bertetanggaan
- Bernilai 0 jika tidak



$$\begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Gambar 7. Graf dan bentuk Matriks Ketetanggaan
Sumber: Graf Bagian 2 oleh Rinaldi Munir

Graf berbobot juga dapat direpresentasikan dalam bentuk matriks ketetanggaan. Pada matriks ketetanggan graf berbobot, jika sebuah simpul tidak terhubung terhadap sebuah simpul lain, maka nilai matriks ketetanggaannya bernilai ∞, dan jika terhubung akan bernilai sesuai dengan bobot.nya

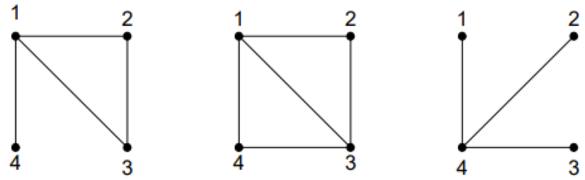


$$\begin{array}{c c} & \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} & \begin{bmatrix} \infty & 12 & \infty & \infty & 10 \\ 12 & \infty & 9 & 11 & 8 \\ \infty & 9 & \infty & 14 & \infty \\ \infty & 11 & 14 & \infty & 15 \\ 10 & 8 & \infty & 15 & \infty \end{bmatrix} \end{array}$$

Gambar 8. Graf Berbobot dan bentuk Matriks Ketetanggan
Sumber: Graf Bagian 2 oleh Rinaldi Munir

E. *Lintasan dan Sirkuit Hamilton*

Lintasan Hamilton merupakan sebuah lintasan yang melalui tiap simpul pada sebuah graf hanya tepat satu kali. Sirkuit Hamilton merupakan sirkuit yang melalui tiap simpul pada sebuah graf hanya tepat satu kali, kecuali simpul asal yang akan dianggap sebagai simpul akhir akan dilalui sebanyak tepat dua kali. Graf Hamilton merupakan sebuah graf yang memiliki Sirkuit Hamilton, sedangkan graf yang hanya memiliki Lintasan Hamilton disebut Graf Semi-Hamilton.

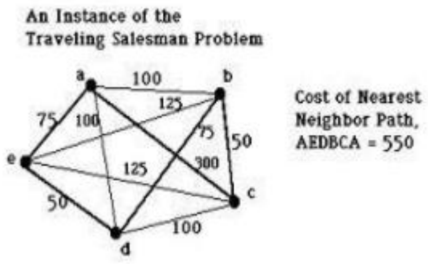


(a) (b) (c)

Gambar 9. (a) Graf yang memiliki lintasan Hamilton (3,2,1,4)
(b). Graf yang memiliki sirkuit Hamilton (1,2,3,4,1)
(c). Graf yang tidak memiliki lintasan atauapun sirkuit Hamilton
Sumber: Graf Bagian 2 oleh Rinaldi Munir

F. *Travelling Salesman Problem (TSP)*

Merupakan sebuah masalah optimasi yang dapat digunakan untuk mencari rute terpendek yang harus ditempuh oleh seorang *sales* yang harus mengunjungi sejumlah kota. Masalah ini dikenal sebagai salah satu masalah yang sulit dipecahkan, karena jika semakin banyak kota yang harus dikunjungi, maka akan semakin sulit masalah ini untuk dipecahkan secara efisien. Dalam masalah TSP ini, tujuan utamanya adalah untuk mencari rute terpedek yang dapat menghubungkan semua kota yang harus dikunjungi tepat satu kali dengan cara yang paling efisien. Graf pada TSP direpresentasikan dalam graf berbobot. Bobot pada graf dapat dinyatakan dalam jarak, biaya, ataupun waktu. Pada sebuah graf yang lengkap yang terdiri atas n buah simpul, terdapat $\frac{(n-1)!}{2}$ sirkuit Hamilton, yang berarti terdapat $\frac{(n-1)!}{2}$ rute yang berbeda untuk mengunjungi tiap kota tepat satu kali dan kembali ke kota asal. Permasalahan TSP ini tidak hanya terbatas pada graf lengkap, asalkan graf tersebut memiliki sirkuit Hamilton.



Gambar 9. Contoh permasalahan *Travelling Salesman Problem*
Sumber: Graf Bagian 3 oleh Rinali Munir

G. *Algoritma Branch and Bound*

Algortima *Branch and Bound* adalah salah satu metode yang dapat digunakan untuk menyelesaikan masalah optimasi yang sulit, misalkan *Travelling Salesman Problem* (TSP). Dengan menggunakan metode ini, masalah yang ada akan dipecah menjadi beberapa submasalah yang lebih kecil, dan kemudian mencari solusi optimal dari setiap submasalah tersebut. Setelah solusi optimal dari setiap submasalah ditemukan, maka akan dilakukan penggabungan kembali solusi-solusi tersebut menjadi satu solusi yang terbaik untuk masalah yang ada.

Algoritma ini dapat digunakan untuk menyelesaikan permasalahan *Travelling Salesman Problem* (TSP) dengan menentukan rute paling efisien dengan bobot yang minimum untuk tiap sisi (rute) yang dilalui dalam sebuah graf berbobot.

Berikut adalah Langkah-langkah yang diperlukan untuk menentukan rute yang paling optimal dalam *Travelling Salesman Problem* dengan menggunakan algoritma *branch and bound* :

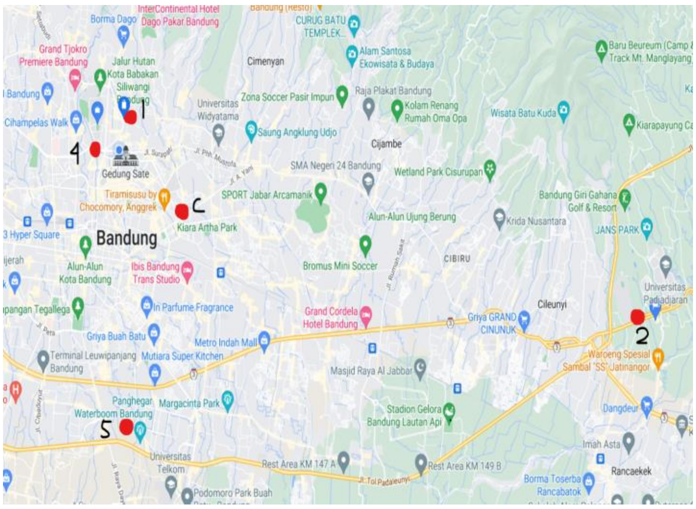
1. Buatlah sebuah graf berbobot yang akan dipecahkan permasalahan *Travelling Salesman Problem*.
2. Bobot tersebut dapat berupa jarak, biaya, maupun waktu yang diperlukan antar simpulnya.
3. Representasikan graf tersebut dalam bentuk Matriks Ketetanggaan (*Adjacency Matrix*).
4. Hitung jumlah total elemen penguran dari semua elemen pengurang dari semua baris dan kolom menjadi batas bawah (*lower bound*) dari tur dengan total bobot minimum. Nilai tersebut akan menjadi nilai untuk simpul akar pada pohon ruang status. ($\hat{c}(root)$)
5. Misalkan A adalah matriks tereduksi untuk simpul R. Lalu, S adalah anak dari simpul R sehingga sisi (R,S) pada pohon ruang status berkoresponden dengan sisi (i,j) pada perjalanan. Hitunglah *cost* pada matriks bobot tereduksi untuk simpul S dengan cara sebagai berikut :
    a. Ubah semua nilai pada baris *i* dan kolom *j* menjadi ∞. Hal ini bertujuan untuk mencegah terjadinya lintasan yang keluar dari simpul *i* atau masuk pada simpul *j*.
    b. Ubah A(*j*,1) menjadi ∞. Hal ini bertujuan agar sisi (*j*,1) tidak digunakan
    c. Reduksi kembali semua baris dan kolom pada matriks A kecuali elemen ∞.
    d. Hitung *cost* pada simpul S dengan cara :
        i. $c(S) = c(R) + A(i,j) + r$
        ii. $c(S)$: bobot minimum simpul S
        iii. $c(R)$: bobot minimum simpul R
        iv. $A(i,j)$: bobot sisi (*i*,*j*) pada graf yang berkoresponden dengan sisi (*R*,*S*) pada pohon ruang status
        v. r : jumlah semua pengurang pada proses memperoleh matriks tereduksi untuk simpul S.
    e. Hasil proses reduksi matriks tersebut akan menghasilkan matriks B
6. Pilih *cost* terkecil untuk setiap S di level yang sama.
7. Ulangi Langkah ke-5 dan 6 untuk level berikutnya hingga diperoleh sirkuit Hamilton dengan bobot paling kecil.

III. HASIL PENELITIAN

A. *Lingkup Area Penelitian*

Pada makalah ini akan dibahas rute paling efisien yang dapat dilalui DAMRI Dipatiukur – Jatinangor, berikut adalah beberapa area pemberhentian darii DAMRI tersebut :

1. **Halte DAMRI Dipatiukur**, Jl. Dipati Ukur, Lebakgede, Kecamatan Coblong, Kota Bandung, Jawa Barat.
2. **Halte Bus DAMRI Jatinangor**, Jl. Raya Cirebon – Bandung No.176, Cikeruh, Kec. Jatinangor, Kab. Sumedang, Jawa Barat.
3. **Jl. Ahmad Yani**, Kacapiring, Kec. Batununggal, Kota Bandung, Jawa Barat.
4. **Baltos Tamansari**, Jl. Tamansari No.27a, Tamansari, Kec. Bandung Wetan, Kota Bandung, Jawa Barat.
5. **Jl. Moh. Toha**, Karasak, Kec. Astanaanyar, Kota Bandung, Jawa Barat.



Gambar 10. Denah Lingkup Area Penelitian
Sumber: *Google Maps*

Berikut adalah *Adjaceny Matrix* graf berbobot, dengan bobot adalah jarak antar simpul dalam satuan kilometer (km) :

$$\begin{array}{cc} & \begin{array}{ccccc} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} \end{array} \\ \begin{array}{c} \mathbf{1}\\ \mathbf{2}\\ \mathbf{3}\\ \mathbf{4}\\ \mathbf{5} \end{array} & \left\| \begin{array}{ccccc} \infty & 21 & 4.2 & 1.9 & 9.7 \\ 21 & \infty & 25 & 29 & 21 \\ 4.2 & 25 & \infty & 4.2 & 9.1 \\ 1.9 & 29 & 4.2 & \infty & 8.2 \\ 9.7 & 21 & 9.1 & 8.2 & \infty \end{array} \right\| \end{array}$$

B. Batasan Area Penelitian

Terdapat beberapa Batasan dari area penelitian yang dilakukan pada makalah ini :

1. Bobot atau efisiensi rute yang dihitung hanya berdasarkan parameter jarak. Sehingga biaya serta waktu yang dibutuhkan belum dijadikan parameter.
2. Rute yang diambil dari suatu titik ke titik yang lain adalah rute terpendek
3. Titik awal dan akhir dari graf ini adalah titik 1, yaitu Halte DAMRI Dipatiukur

C. Perhitungan Algoritma Branch and Bound dalam Penentuan Rute DAMRI Paling Efisien

1. Lakukan reduksi baris dan kolom pada matriks awal hingga mendapatkan setidaknya satu buah nol pada setiap baris dan kolom.

   Reduced Cost Matrix (1) : Rute 1

$$\begin{array}{cc} & \begin{array}{ccccc} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} \end{array} \\ \begin{array}{c} \mathbf{1}\\ \mathbf{2}\\ \mathbf{3}\\ \mathbf{4}\\ \mathbf{5} \end{array} & \left\| \begin{array}{ccccc} \infty & 21 & 4.2 & 1.9 & 9.7 \\ 21 & \infty & 25 & 29 & 21 \\ 4.2 & 25 & \infty & 4.2 & 9.1 \\ 1.9 & 29 & 4.2 & \infty & 8.2 \\ 9.7 & 21 & 9.1 & 8.2 & \infty \end{array} \right\| \end{array}$$

$R_1 - 1.9$; $R_2 - 21$; $R_3 - 4.2$; $R_4 - 1.9$; $R_5 - 8.2$

Maka diperoleh :

$$\begin{array}{cc} & \begin{array}{ccccc} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} \end{array} \\ \begin{array}{c} \mathbf{1}\\ \mathbf{2}\\ \mathbf{3}\\ \mathbf{4}\\ \mathbf{5} \end{array} & \left\| \begin{array}{ccccc} \infty & 19.1 & 2.3 & 0 & 7.8 \\ 0 & \infty & 4 & 8 & 0 \\ 0 & 20.8 & \infty & 0 & 4.9 \\ 0 & 27.1 & 2.3 & \infty & 6.3 \\ 1.5 & 12.8 & 0.9 & 0 & \infty \end{array} \right\| \end{array}$$

Lakukan reduksi tiap kolom karena tiap baris dan kolom belum memiliki nilai 0.

$C_1 - 0$; $C_2 - 12.8$; $C_3 - 0.9$; $C_4 - 0$; $C_5 - 0$

Maka diperoleh :

$$\begin{array}{cc} & \begin{array}{ccccc} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} \end{array} \\ \begin{array}{c} \mathbf{1}\\ \mathbf{2}\\ \mathbf{3}\\ \mathbf{4}\\ \mathbf{5} \end{array} & \left\| \begin{array}{ccccc} \infty & 6.3 & 1.4 & 0 & 7.8 \\ 0 & \infty & 3.1 & 8 & 0 \\ 0 & 8 & \infty & 0 & 4.9 \\ 0 & 14.3 & 1.4 & \infty & 6.3 \\ 1.5 & 0 & 0 & 0 & \infty \end{array} \right\| \end{array}$$

C(1) = 1.9 + 21 + 4.2 + 1.9 + 8.2 + 12.8 + 0.9 = 50.9

2. Ubah semua nilai baris *i* dan kolom *j* untuk setiap rute yang ditempuh pada matriks A menjadi ∞. Lalu lakukan reduksi baris dan kolom dan hitung costnya.

   Reduced Cost Matrix (2) : Rute 1 – 2

$$\begin{array}{cc} & \begin{array}{ccccc} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} \end{array} \\ \begin{array}{c} \mathbf{1}\\ \mathbf{2}\\ \mathbf{3}\\ \mathbf{4}\\ \mathbf{5} \end{array} & \left\| \begin{array}{ccccc} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 3.1 & 8 & 0 \\ 0 & \infty & \infty & 0 & 4.9 \\ 0 & \infty & 1.4 & \infty & 6.3 \\ 1.5 & \infty & 0 & 0 & \infty \end{array} \right\| \end{array}$$

   r = 0

   Reduced Cost Matrix (2) : Rute 1 – 3

$$\begin{array}{cc} & \begin{array}{ccccc} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} \end{array} \\ \begin{array}{c} \mathbf{1}\\ \mathbf{2}\\ \mathbf{3}\\ \mathbf{4}\\ \mathbf{5} \end{array} & \left\| \begin{array}{ccccc} \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & 8 & 0 \\ \infty & 8 & \infty & 0 & 4.9 \\ 0 & 14.3 & \infty & \infty & 6.3 \\ 1.5 & 0 & \infty & 0 & \infty \end{array} \right\| \end{array}$$

   r = 0

   Reduced Cost Matrix (2) : Rute 1 – 4

$$\begin{array}{cc} & \begin{array}{ccccc} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} \end{array} \\ \begin{array}{c} \mathbf{1}\\ \mathbf{2}\\ \mathbf{3}\\ \mathbf{4}\\ \mathbf{5} \end{array} & \left\| \begin{array}{ccccc} \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & 3.1 & \infty & 0 \\ 0 & 8 & \infty & \infty & 4.9 \\ \infty & 14.3 & 1.4 & \infty & 6.3 \\ 1.5 & 0 & 0 & \infty & \infty \end{array} \right\| \end{array}$$

   r = 1.4

   Reduced Cost Matrix (2) : Rute 1-5

$$\begin{array}{cc} & \begin{array}{ccccc} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} \end{array} \\ \begin{array}{c} \mathbf{1}\\ \mathbf{2}\\ \mathbf{3}\\ \mathbf{4}\\ \mathbf{5} \end{array} & \left\| \begin{array}{ccccc} \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & 3.1 & 8 & \infty \\ 0 & 8 & \infty & 0 & \infty \\ 0 & 14.3 & 1.4 & \infty & \infty \\ \infty & 0 & 0 & 0 & \infty \end{array} \right\| \end{array}$$

   r = 0

C(2) = C(1) + r

Dari 5 matrix tersebut, nilai C(2) terkecil diperoleh pada matriks Rute 1-2, 1-3, 1-5 dengan nilai C(2) = 50.9, maka jadikan matriks tersebut sebagai matriks B.

3. Lakukan Langkah yang sama dengan Langkah kedua, namun gunakan matriks B, yaitu matriks yang diperoleh pada Langkah kedua. Lakukan reduksi matriks B dan hitung cost-nya.

   Reduced Cost Matrix (3) : Rute 1-2-3

$$\begin{array}{cc} & \begin{array}{ccccc} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} \end{array} \\ \begin{array}{c} \mathbf{1}\\ \mathbf{2}\\ \mathbf{3}\\ \mathbf{4}\\ \mathbf{5} \end{array} & \left\| \begin{array}{ccccc} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 8 & 0 \\ \infty & \infty & \infty & 0 & 4.9 \\ 0 & \infty & \infty & \infty & 6.3 \\ 1.5 & \infty & \infty & 0 & \infty \end{array} \right\| \end{array}$$

   r = 0

   Reduced Cost Matrix (3) : Rute 1-2-4

$$\begin{array}{c||ccccc||} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & \infty & \infty & \infty & \infty \\ 2 & \infty & \infty & 3.1 & \infty & 0 \\ 3 & 0 & \infty & \infty & \infty & 4.9 \\ 4 & \infty & \infty & 1.4 & \infty & 6.3 \\ 5 & 1.5 & \infty & 0 & \infty & \infty \end{array}$$
r = 1.4

Reduced Cost Matrix (3) : Rute 1-2-5

$$\begin{array}{c||ccccc||} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & \infty & \infty & \infty & \infty \\ 2 & \infty & \infty & 3.1 & 8 & \infty \\ 3 & 0 & \infty & \infty & 0 & \infty \\ 4 & 0 & \infty & 1.4 & \infty & \infty \\ 5 & \infty & \infty & 0 & 0 & \infty \end{array}$$
r = 3.1

Reduced Cost Matrix (3) : Rute 1-3-2

$$\begin{array}{c||ccccc||} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & \infty & \infty & \infty & \infty \\ 2 & \infty & \infty & \infty & 8 & 0 \\ 3 & \infty & \infty & \infty & 0 & 4.9 \\ 4 & 0 & \infty & \infty & \infty & 6.3 \\ 5 & 1.5 & \infty & \infty & 0 & \infty \end{array}$$
r = 0

Reduced Cost Matrix (3) : Rute 1-3-4

$$\begin{array}{c||ccccc||} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & \infty & \infty & \infty & \infty \\ 2 & 0 & \infty & \infty & \infty & 0 \\ 3 & \infty & 8 & \infty & \infty & 4.9 \\ 4 & \infty & 14.3 & \infty & \infty & 6.3 \\ 5 & 1.5 & 0 & \infty & \infty & \infty \end{array}$$
r = 4.9 + 6.3 = 11.2

Reduced Cost Matrix (3) : Rute 1 – 3 – 5

$$\begin{array}{c||ccccc||} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & \infty & \infty & \infty & \infty \\ 2 & 0 & \infty & \infty & 8 & \infty \\ 3 & \infty & 8 & \infty & 0 & \infty \\ 4 & 0 & 14.3 & \infty & \infty & \infty \\ 5 & \infty & 0 & \infty & 0 & \infty \end{array}$$

r = 0

Reduced Cost Matrix (3) : Rute 1 – 5 – 2

$$\begin{array}{c||ccccc||} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & \infty & \infty & \infty & \infty \\ 2 & \infty & \infty & 3.1 & 8 & \infty \\ 3 & 0 & \infty & \infty & 0 & \infty \\ 4 & 0 & \infty & 1.4 & \infty & \infty \\ 5 & \infty & \infty & 0 & 0 & \infty \end{array}$$
r = 3.1

Reduced Cost Matrix (3) : Rute 1-5-3

$$\begin{array}{c||ccccc||} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & \infty & \infty & \infty & \infty \\ 2 & 0 & \infty & \infty & 8 & \infty \\ 3 & \infty & 8 & \infty & 0 & \infty \\ 4 & 0 & 14.3 & \infty & \infty & \infty \\ 5 & \infty & 0 & \infty & 0 & \infty \end{array}$$

r = 0

Reduced Cost Matrix (3) : Rute 1-5-4

$$\begin{array}{c||ccccc||} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & \infty & \infty & \infty & \infty \\ 2 & 0 & \infty & 3.1 & \infty & \infty \\ 3 & 0 & 8 & \infty & \infty & \infty \\ 4 & \infty & 14.3 & 1.4 & \infty & \infty \\ 5 & \infty & 0 & 0 & \infty & \infty \end{array}$$
r = 1.4

C(3) = C(2) + r
Dari matriks-matriks tersebut, diperoleh nilai C(3) terkecil adalah 50.9, yaitu Rute 1-2-3, 1-3-2, 1-3-5, 1-5-3. Jadikan matriks-matriks tersebut sebagai matriks B.

4. Lakukan langkah yang sama dengan menggunakan matriks B yang baru.

   Reduced Cost Matrix (4) : Rute 1-2-3-4

$$\begin{array}{c||ccccc||} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & \infty & \infty & \infty & \infty \\ 2 & \infty & \infty & \infty & \infty & 0 \\ 3 & \infty & \infty & \infty & \infty & 4.9 \\ 4 & \infty & \infty & \infty & \infty & 6.3 \\ 5 & 1.5 & \infty & \infty & \infty & \infty \end{array}$$
r = 4.9 + 6.3 + 1.5 = 12.7

Reduced Cost Matrix (4) : Rute 1-2-3-5

$$\begin{array}{c||ccccc||} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & \infty & \infty & \infty & \infty \\ 2 & \infty & \infty & \infty & 8 & \infty \\ 3 & \infty & \infty & \infty & 0 & \infty \\ 4 & 0 & \infty & \infty & \infty & \infty \\ 5 & \infty & \infty & \infty & 0 & \infty \end{array}$$
r = 8

Reduced Cost Matrix (4) : Rute 1-3-2-4

$$\begin{array}{c||ccccc||} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & \infty & \infty & \infty & \infty \\ 2 & \infty & \infty & \infty & \infty & 0 \\ 3 & \infty & \infty & \infty & \infty & 4.9 \\ 4 & \infty & \infty & \infty & \infty & 6.3 \\ 5 & 1.5 & \infty & \infty & \infty & \infty \end{array}$$
r = 4.9 + 6.3 + 1.5 = 12.7

Reduced Cost Matrix (4) : Rute 1-3-2-5

$$\begin{array}{c||ccccc||} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & \infty & \infty & \infty & \infty \\ 2 & \infty & \infty & \infty & 8 & \infty \\ 3 & \infty & \infty & \infty & 0 & \infty \\ 4 & 0 & \infty & \infty & \infty & \infty \\ 5 & \infty & \infty & \infty & 0 & \infty \end{array}$$
r = 8

Reduced Cost Matrix (4) : Rute 1 – 3 – 5 - 2

$$\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array}$$

$$\begin{array}{c}\mathbf{1}\\\mathbf{2}\\\mathbf{3}\\\mathbf{4}\\\mathbf{5}\end{array}\left\|\begin{array}{ccccc}\infty&\infty&\infty&\infty&\infty\\\infty&\infty&\infty&8&\infty\\\infty&\infty&\infty&0&\infty\\0&\infty&\infty&\infty&\infty\\\infty&\infty&\infty&0&\infty\end{array}\right\|$$

r = 8

Reduced Cost Matrix (4) : Rute 1 – 3 – 5 - 4

$$\begin{array}{cc} & \begin{array}{ccccc}\mathbf{1}&\mathbf{2}&\mathbf{3}&\mathbf{4}&\mathbf{5}\end{array}\\ \begin{array}{c}\mathbf{1}\\\mathbf{2}\\\mathbf{3}\\\mathbf{4}\\\mathbf{5}\end{array} & \left\|\begin{array}{ccccc}\infty&\infty&\infty&\infty&\infty\\0&\infty&\infty&\infty&\infty\\\infty&8&\infty&\infty&\infty\\\infty&14.3&\infty&\infty&\infty\\\infty&0&\infty&\infty&\infty\end{array}\right\|\end{array}$$

r = 14.3

Reduced Cost Matrix (4) : Rute 1-5-3-2

$$\begin{array}{cc} & \begin{array}{ccccc}\mathbf{1}&\mathbf{2}&\mathbf{3}&\mathbf{4}&\mathbf{5}\end{array}\\ \begin{array}{c}\mathbf{1}\\\mathbf{2}\\\mathbf{3}\\\mathbf{4}\\\mathbf{5}\end{array} & \left\|\begin{array}{ccccc}\infty&\infty&\infty&\infty&\infty\\\infty&\infty&\infty&8&\infty\\\infty&\infty&\infty&0&\infty\\0&\infty&\infty&\infty&\infty\\\infty&\infty&\infty&0&\infty\end{array}\right\|\end{array}$$

r = 8

Reduced Cost Matrix (4) : Rute 1-5-3-4

$$\begin{array}{cc} & \begin{array}{ccccc}\mathbf{1}&\mathbf{2}&\mathbf{3}&\mathbf{4}&\mathbf{5}\end{array}\\ \begin{array}{c}\mathbf{1}\\\mathbf{2}\\\mathbf{3}\\\mathbf{4}\\\mathbf{5}\end{array} & \left\|\begin{array}{ccccc}\infty&\infty&\infty&\infty&\infty\\0&\infty&\infty&\infty&\infty\\\infty&8&\infty&\infty&\infty\\\infty&14.3&\infty&\infty&\infty\\\infty&0&\infty&\infty&\infty\end{array}\right\|\end{array}$$

r = 14.3

C(4) = C(3) + r

Dari matriks-matriks tersebut, diperoleh nilai C(4) terkecil adalah 58.9, yaitu Rute 1-2-3-5, 1-3-2-5, 1-3-5-2, 1-5-3-2. Jadikan matriks-matriks tersebut sebagai matriks B.

5. Dari matriks-matriks tersebut, sudah dapat disimpulkan terdapat 4 solusi yang dinilai efisien, lalu kita perlu untuk menghitung bobot yang ada dari solusi-solusi tersebut untuk menentukan rute yang paling efisien :
   a. 1-2-3-5-4-1 : 65.2
   b. 1-3-2-5-4-1 : 60.3
   c. 1-3-5-2-4-1 : 65.2
   d. 1-5-3-2-4-1 : 74.7

D. Hasil

Dari perhitungan menggunakan Algoritma *Branch and Bound* tersebut, dapat ditentukan rute paling efisien adalah **1-3-2-5-4-1**, dengan jarak tempuh **60.3 Km** yaitu :

Halte DAMRI Dipatiukur - Jl. Ahmad Yani - Halte Bus DAMRI Jatinangor - Jl. Moh. Toha - Baltos Tamansari - Halte DAMRI Dipatiukur

IV. KESIMPULAN

Sebuah graf dapat diimplementasikan untuk menyelesaikan banyak permasalahan, salah satunya adalah untuk menyelesaikan *Travelling Salesman Problem* (TSP), yaitu sebuah permasalahan dimana kita mencari rute terpendek saat mengunjungi beberapa titik lokasi tepat satu kali dan kembali ke titik awal. Salah satu cara dalam menyeesaikan TSP ini adalah dengan menggunakan Algoritma *Branch and Bound*.

Penerapan TSP dengan Algoritma *Branch and Bound* pada makalah ini dicontohkan dengan menentukan rute terpendek dari rute perjalanan DAMRI Dipatiukur-Jatinangor. Dengan diselesaikannya permasalahan ini, jarak perjalanan DAMRI akan lebih pendek, sehingga juga akan mempersingkat waktu perjalanan.

V. UCAPAN TERIMA KASIH

Puji dan syukur diucapkan kepada Tuhan yang Maha Esa sehingga makalah ini dapat diselesaikan. Tidak lupa juga kepada Orang Tua penulis yang kerap memberi dukungan jasmani dan rohani sehingga makalah ini dapat disusun dengan baik. Lalu, kepada dosen pengampu mata kuliah Matematika Diskrit K3, Dr.Ir. Rinaldi Munir, M.T. yang telah menurunkan ilmunya kepada penulis. Terakhir, rekan-rekan penulis yang telah kerap menyemangati dan menginspirasi penulis dalam menyelesaikan makalah ini.

REFERENCES

[1] Munir, Rinaldi (2022), Graf Bagian 1, *Slide Kuliah IF2120 Matematika Diskrit*, Bandung, Indonesia. Diaksses 4 Desember 2022.
[2] Munir, Rinaldi (2022), Graf Bagian 2, *Slide Kuliah IF2120 Matematika Diskrit*, Bandung, Indonesia. Diakses 4 Desember 2022.
[3] Munir, Rinaldi (2022), Graf Bagian 3, *Slide Kuliah IF2120 Matematika Diskrit*, Bandung, Indonesia. Diakses 4 Desember 2022.
[4] Wengrow, Jay (2020). Connecting Everything with Graphs, *A Common Sense Guide to Data Structures and Algorithm*, pp 331-386, Raleigh, North Carolina. Diakses 4 Desember 2022.
[5] Munir, Rinaldi (2022), Algoritma Branch and Bound Bagian 1, *Slide Kuliah IF2251 Strategi Algoritma*, Bandung, Indonesia. Diakses 9 Desember 2022.
[6] Munir, Rinaldi (2022), Algoritma Branch and Bound Bagian 2, *Slide Kuliah IF2251 Strategi Algoritma*, Bandung, Indonesia. Diakses 9 Desember 2022.

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, 10 Desember 2022

Ahmad Nadil
13521024