Aplikasi Travelling Salesman Problem untuk Menentukan Rute Roadshow ITB JTR di Kota Pekanbaru

Christopher Justine William 13519006[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13519006@std.stei.itb.ac.id

***Abstrak*—Perkembangan teknologi dan ilmu pengetahuan telah membantu manusia untuk dapat melakukan segala sesuatu dengan lebih efisien. Salah satunya contohnya adalah untuk menentukan rute perjalanan. Dengan mengoptimalkan rute perjalanan, hal tersebut dapat meningkatkan efisiensi baik dari segi waktu perjalanan maupun biaya yang harus dikeluarkan. Dalam makalah ini akan dibahas salah satu permasalahan graf, yaitu *Travelling Salesman Problem* (TSP) dengan algoritma *branch and bound* untuk menentukan rute perjalanan yang paling efiseien dari suatu Roadshow acara yang bernama ITB on Journey to Riau-Kepri.**

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

I. Pendahuluan

ITB on Journey to Riau-Kepri (ITB JTR) merupakan acara tahunan yang diselenggarakan oleh Unit Kebudayaan Melayu Riau ITB (UKMR) yang bekerja sama dengan Aku Masuk ITB (AMI) dalam rangka membawa semangat untuk melanjutkan pendidikan tinggi dan membantu memfasilitasi penyebaran informasi mengenai dunia perkuliahan kepada siswa-siswi SMA khususnya di daerah Riau dan Kepulauan Riau. Setiap tahunnya, dalam rangkaian acara tersebut, terdapat roadshow atau kunjungan langsung ke sekolah-sekolah di daerah Riau dan Kepulauan Riau untuk memberikan informasi mengenai perkuliahan di ITB, mulai dari pengenalan tentang ITB itu sendiri, jalur masuknya, hingga uang kuliah dan beasiswa (dikarenakan pandemi, ITB JTR 2021 akan melakukan kunjungan ke sekolah-sekolah secara online). Roadshow tersebut biasanya dilaksanakan selama beberapa hari mulai dari awal bulan Januari. Saat melakukan kunjungan langsung, mahasiswa-mahasiswi ITB yang berasal dari Riau dan Kepulauan Riau yang tergabung dalam UKMR ITB dibagi menjadi beberapa kelompok untuk berkunjung ke sekolah-sekolah yang sudah ditetapkan sesuai dengan kelompoknya. Setiap kelompok kemudian akan melakukan perjalanan dari satu sekolah ke sekolah lainnya. Karena keterbatasan waktu dan juga jumlah kelompok, sementara ada banyak sekolah yang bisa dikunjungi, maka setiap kelompoknya memerlukan rute perjalanan yang paling efisien. Ketika sudah mendapatkan rute perjalanan yang paling efisien, setiap kelompoknya dapat mengkoordinasikan waktu kunjungan ke sekolah yang akan dikunjungi berdasarkan rute perjalanan tersebut agar dapat menghemat waktu dan juga dapat mengunjungi sejumlah tertentu sekolah pada hari yang sama.

Dalam makalah ini, penulis menggunakan salah satu penerapan dari graf, yaitu *Travelling Salesman Problem* (TSP) dengan algoritma *Branch and Bound*, untuk dapat menemukan rute perjalanan yang paling efisien agar dapat mengunjungi setiap sekolah. Pada kenyataannya, ada faktor-faktor lain yang memengaruhi efisien atau tidaknya suatu rute perjalanan, namun dalam makalah ini, penulis berfokus hanya pada jarak tiap-tiap sekolah.

II. Landasan Teori

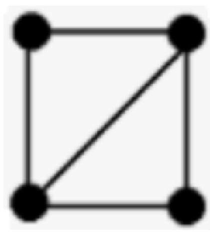
A. Definisi Graf

Graf merupakan struktur yang merepresentasikan hubungan objek-objek diskrit yang ditandai dengan simpul (*vertex*) dan sisi (*edge*). Simpul merepresentasikan objek di dalam struktur tersebut, sedangakn sisi merepresentasikan hubungan antara objek di dalam struktur tersebut. Graf dapat ditulis sebagai $G = (V,E)$, dengan $V = \{v_1, v_2, v_3, \ldots, v_n\}$ merupakan himpunan tidak kosong dari simpul-simpulnya dan $E = \{e_1, e_2, \ldots, e_n\}$ merupakan himpunan sisi yang menghubungkan sepasang simpul di dalam graf.

B. Jenis-Jenis Graf

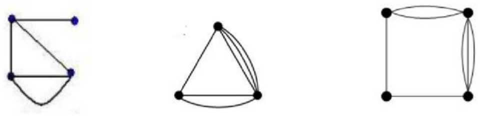
Berdasarkan ada tidaknya gelang atau sisi ganda, graf dapat dikelompokkan menjadi dua jenis:

1. Graf sederhana (*simpe graph*)
   Graf sederhana merupakan graf yang tidak mengandung sisi gelang maupun sisi ganda.



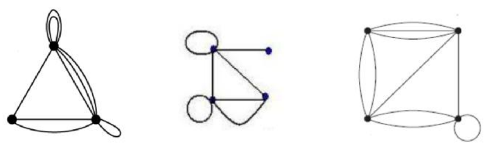
Gambar 1. Graf sederhana
Sumber: Graf Bagian 1 oleh Rinaldi Munir

2. Graf tak-sederhana (*unsimple-graph*)
   Graf tak-sederhana merupakan graf yang mengandung sisi ganda atau sisi gelang. Graf tak-sederhana dapat dibagi lagi menjadi dua jenis:
   a. Graf ganda (*multi graph*)
      Graf ganda merupakan graf yang mengandung sisi ganda.



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

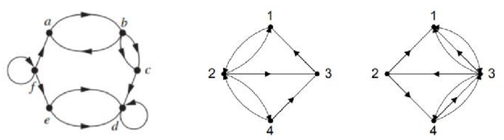
   b. Graf semu (*pseudo graph*)
      Graf semu merupakan graf yang mengandung sisi gelang.



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

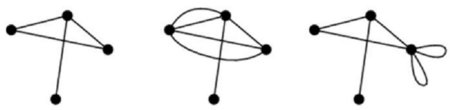
Berdasarkan ada atau tidaknya orientasi arah pada sisi, graf dapat dibagi menjadi dua jenis:

1. Graf berarah *(directed graph*)
   Graf berarah merupakan graf yang memiliki orientasi arah pada setiap sisinya.



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

2. Graf tak-berarah (*undirected graph*)
   Graf tak-berarah merupakan graf yang tidak memiliki orientasi arah pada sisinya.



Gambar 5. Graf tak-berarah
Sumber: Graf Bagian 1 oleh Rinaldi Munir

C. Terminologi Graf

Terdapat beberapa istilah/terminologi yang digunakan dalam graf, pada makalah ini penulis hanya akan memuat beberapa diantaranya:

1. Ketetanggan (*Adjacent*)
   Dua buah simpul pada graf bertetangga jika keduanya terhubung langsung oleh sebuah sisi.

2. Bersisian (*Incidency*)
   Sebuah sisi e dikatakan bersisian dengan simpul v_j dan simpul v_k jika terdapat sembarang $e = (v_j, v_k)$.

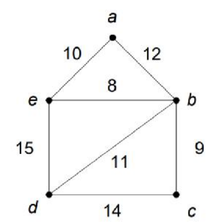
3. Derajat (*Degree*)
   Derajat suatu simpul menyatakan jumlah sisi yang bersisian dengan simpul tersebut. Derajat dapat ditulis dengan notasi $d(v)$.

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

5. Sirkuit (*Circuit*)
   Sirkuit merupakan lintasan yang berawal dam berakhir pada simpul yang sama.

6. Keterhubungan (*Connected*)
   Simpul v_1 dan v_2 disebut terhubung jika terdapat lintasan dari v_1 ke v_2 dan graf G disebut graf terhubung (*connected graph*) jika terdapat lintasan dari v_i ke v_j untuk setiap pasang simpul v_i dan v_j dalam himpunan V.

7. Graf berbobot (*Weighted graph*)
   Graf berbobot merupakan graf yang setiap sisinya memiliki bobot tertentu.



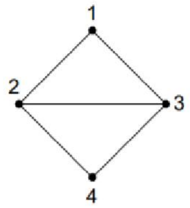
Gambar 6. Graf berbobot
Sumber: Graf Bagian 1 oleh Rinaldi Munir

8. Graf lengkap (Complete graph)
   Graf lengkap merupakan graf sederhana yang setiap simpulnya mempunyai sisi ke semua simpul lainnya. Jumlah sisi pada graf lengkap yang terdiri dari n simpul adalah n(n-1)/2.

D. Representasi Graf

Terdapat beberapa representasi graf yang dapat digunakan, namun pada makalah ini penulis hanya memuat Matriks Ketetanggaan.

Matriks Ketetanggaan (*Adjacency Matrix*)

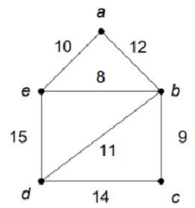


$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}\begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Gambar 7. Matriks ketetanggaan
Sumber: Graf Bagian 2 oleh Rinaldi Munir

Setiap elemen matriks disimbolkan dengan a_{ij}, bernilai 1 jika simpul *i* dan *j* bertetangga dan bernilai 0 jika simpul *i* dan *j* tidak bertetangga.

Untuk graf berbobot, a_{ij} diisi dengan bobot yang dimiliki sisi yang menghubungkan simpul *i* dan simpul *j*, atau ∞ jika tidak bertetangga.

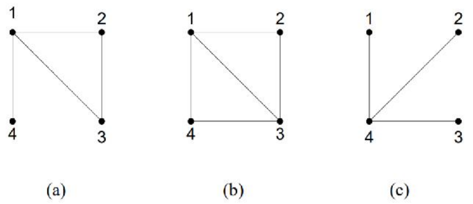


$$\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array}\begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{bmatrix} \infty & 12 & \infty & \infty & 10 \\ 12 & \infty & 9 & 11 & 8 \\ \infty & 9 & \infty & 14 & \infty \\ \infty & 11 & 14 & \infty & 15 \\ 10 & 8 & \infty & 15 &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

Gambar 8. Matriks ketetanggaan graf berbobot
Sumber: Graf Bagian 2 oleh Rinaldi Munir

E. Lintasan dan Sirkuit Hamilton

Lintasan Hamilton merupakan lintasan yang melalui setiap simpul di dalam graf tepat satu kali. Sirkuit Hamilton merupakan sirkut yang melalui setiap simpul di dalam graf tepat satu kali, kecuali simpul asal yang sekaligus sebagai simpul akhir yang dilalui dua kali. Graf dapat dikatakan sebagai graf Hamilton apabila memiliki sirkuit Hamilton. Jika graf hanya memiliki lintasan Hamilton, maka graf tersebut dapat dikatakan sebagai graf semi-Hamilton.



Gambar 9. (a) graf dengan lintasan Hamilton. (b) graf dengan sirkuit Hamilton. (c) graf yang tidak memiliki sirkuit dan lintasan hamilton
Sumber: Graf Bagian 3 oleh Rinaldi Munir

F. Travelling Salesman Problem

Travelling Salesman Problem merupakan salah satu permasalahan sirkuit Hamilton, yaitu untuk mencari biaya minimum tur dari beberapa kota yang terhubung jika mengunjungi setiap kota tepat satu kali. Jika terdapat himpunan tempat yang membentuk graf terhubung, maka jalur yang mengubungkan tempat-tempat tersebut mempunyai biaya atau harga tertentu yang dapat direpresentasikan dengan graf berbobot. Untuk suatu graf lengkap yang terdiri dari *n* kota, terdapat $\frac{(n-1)!}{2}$ sirkuit Hamilton, yang berarti terdapat $\frac{(n-1)!}{2}$ rute yang berbeda untuk mengunjungi setiap kota dan kembali ke tempat asal. Permasalahan TSP tidak terbatas hanya pada graf lengkap asalkan graf tersebut memiliki sirkuit Hamilton.

G. Algoritma Branch and Bound

Algoritma *Branch and Bound*, diperkenalkan oleh A.H. Land dan A.G. Doig pada tahun 1960, merupakan algoritma yang digunakan untuk menyelesaikan permasalahan optimasi dengan meminimalkan atau memaksimalkan suatu fungsi objektif yang tidak melanggar batasan persoalan. Algoritma ini terdiri dari dua bagian, branching dan bounding. *Branching* dilakukan dengan menelusuri daerah penyelesaian ke beberapa sub daerah yang lebih kecil. *Bounding* dilakukan dengan menentukan nilai batasan untuk suatu penyelesaian optimal di dalam sub daerah.

Algoritma ini dapat digunakan untuk menyelesaikan permasalahan *Travelling Salesman Problem* dengan menentukan rute yang paling optimal yaitu rute dengan bobot minimum untuk tiap-tiap jalur yang dilaluinya di dalam graf.

Langkah-langkah yang diperlukan untuk menyelesaikan TSP dengan algoritma *branch and bound* adalah sebagai berikut :

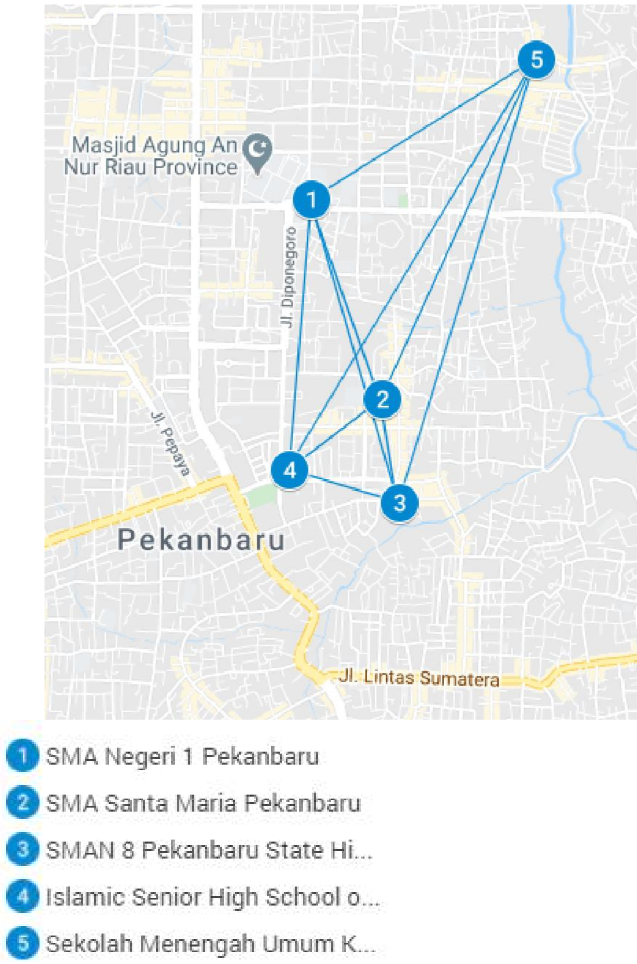
1. Representasikan graf berbobot ke bentuk *Adjacency Matrix.*
2. Lakukan reduksi untuk setiap baris dan kolom pada matriks tersebut hingga setiap baris dan kolomnya mengandung paling sedikit satu buah nol.
3. Hitung batas bawah dari total bobot minimum, yaitu jumlah total elemen pengurang dari semua baris dan kolom yang telah tereduksi. Batas bawah tersebut akan menjadi cost simpul akar (simpul R).
4. Misalkan A adalah matriks tereduksi simpul R, S adalah anak dari simpul R sehingga sisi (*R, S*) pada pohon ruang status berkoresponden dengan sisi (*i, j*) pada perjalanan. Hitung cost pada matriks tereduksi untuk simpul S :
   - Ubah semua nilai baris i dan kolom j pada matriks A menjadi ∞.
   - Ubah A(j, 1) menjadi ∞.
   - Lakukan reduksi kembali semua baris dan kolom pada matriks A kecuali elemen ∞.
   - Hitung cost simpul S dengan :
     $c(S) = c(R) + A(i,j) + r$
     $c(S)$: bobot minimum simpul S
     $c(R)$: bobot minimum simpul R
     $A(i,j)$: bobot sisi (*i, j*) pada graf yang berkoresponden dengan sisi (*R, S*)
     r : total semua pengurang matriks tereduksi simpul S
   - Hasil reduksi tersebut menghasilkan matriks B.
5. Pilihlah cost terkecil untuk setiap S di level yang sama, lalu ulangi langkah 4 untuk level berikutnya hingga diperoleh sirkuit Hamilton dengan bobot minimum.

III. Pembahasan

A. Area Penelitian

Lokasi yang digunakan dalam makalah ini berasal dari sekolah-sekolah yang terdapat di kota Pekanbaru, berikut beberapa diantaranya:

1. SMA Negeri 1 Pekanbaru
   Jl. Sultan Syarif Qasim No.159, Rintis, Kec. Lima Puluh, Kota Pekanbaru.
2. SMA Santa Maria Pekanbaru
   Jl. Ronggo Warsito, Gobah, Suka Maju, Kec. Sail, Kota Pekanbaru.
3. SMA Negeri 8 Pekanbaru
   Jl. Abdul Muis No.14, Cinta Raja, Kec. Sail, Kota Pekanbaru.
4. Madrasah Aliyah Negeri 2 Kota Pekanbaru
   Jl. Diponegoro No.55, Cinta Raja, Kec. Sail, Kota Pekanbaru.
5. Sekolah Menengah Umum Kristen Kalam Kudus
   Jl. Lokomotif No.118, Rejosari, Kec. Tenayan Raya, Kota Pekanbaru.



Gambar 10. Denah Area Penelitian
Sumber: Google Map

Berikut merupakan area penelitian yang telah direpresentasikan dalam bentuk *adjacency matrix* graf berarah berbobot dengan bobot tiap sisi merepresentasikan jarak tiap-tiap sekolah dalam satuan kilometer (km).

sekolah	1	2	3	4	5
1	∞	2.2	2.6	1.7	2.2
2	2.6	∞	0.8	1	3
3	3	0.75	∞	0.85	3.4
4	3.1	1	0.85	∞	3.9
5	2	3	3.7	3.8	∞

B. Batasan dan Asumsi Penelitian

Dalam penulisan makalah ini, terdapat beberapa batasan dan asumsi yang digunakan, yaitu sebagai berikut:

1. Efisiensi rute hanya ditentukan berdasarkan jarak.
2. Jika ada lebih dari satu rute yang menghubungkan satu sekolah ke sekolah yang lain, maka dalam makalah ini diambil rute terpendeknya.
3. Jumlah sekolah yang dikunjungi dibatasi lima sekolah dalam hari yang sama.
4. Setiap sekolah yang akan dikunjungi sudah pasti menerima kunjungan pada waktu yang telah ditentukan sebelumnya berdasarkan pemilihan rute roadshow.
5. Titik kumpul awal dan titik kumpul akhir diasumsikan sama, yaitu di SMA Negeri 1 Pekanbaru.

C. Penerapan Algoritma Branch and Bound dalam Penentuan Rute Roadshow yang Paling Efisien

Langkah pertama : Lakukan reduksi baris dan kolom pada matriks awal hingga mendapatkan setidaknya satu buah nol pada setiap baris dan kolom. Matriks hasil reduksi tersebut membentuk matriks A. Hitung cost dari jumlah total pengurang yang mereduksi matriks tersebut. Cost tersebut akan menjadi batas bawah total bobot minimum.

- Reduced Cost Matrix (1) : Rute 1

	1	2	3	4	5
1	∞	2.2	2.6	1.7	2.2
2	2.6	∞	0.8	1	3
3	3	0.75	∞	0.85	3.4
4	3.1	1	0.85	∞	3.9
5	2	3	3.7	3.8	∞

R_1 – 1.7; R_2 – 0.8; R_3 – 0.75; R_4 – 0.85; R_5 – 2; C_5 – 0.5

$$\begin{array}{c|ccccc|}
 & 1 & 2 & 3 & 4 & 5 \\
1 & \infty & 0.5 & 0.9 & 0 & 0 \\
2 & 1.8 & \infty & 0 & 0.2 & 1.7 \\
3 & 2.25 & 0 & \infty & 0.1 & 2.15 \\
4 & 2.25 & 0.15 & 0 & \infty & 2.55 \\
5 & 0 & 1 & 1.7 & 1.8 & \infty
\end{array} = A$$

$C(1) = 1.7 + 0.8 + 0.75 + 0.85 + 2 + 0.5 = 6.6$

Langkah kedua : Ubah semua nilai baris i dan kolom j untuk setiap rute yang ditempuh pada matriks A menjadi ∞. Ubah juga A(j,1) menjadi ∞. Lakukan reduksi baris dan hitung cost-nya. Matriks yang memiliki cost terendah akan menjadi matriks B. Ubah batas bawah total bobot minimum menjadi cost terendah hasil reduksi matriks A.

- Reduced Cost Matrix (2): Rute 1 – 2

$$\begin{array}{c|ccccc|}
 & 1 & 2 & 3 & 4 & 5 \\
1 & \infty & \infty & \infty & \infty & \infty \\
2 & \infty & \infty & 0 & 0.2 & 1.7 \\
3 & 2.25 & \infty & \infty & 0.1 & 2.15 \\
4 & 2.25 & \infty & 0 & \infty & 2.55 \\
5 & 0 & \infty & 1.7 & 1.8 & \infty
\end{array}$$

$R_4 - 0.1$; $R_5 - 1.7$;

$$\begin{array}{c|ccccc|}
 & 1 & 2 & 3 & 4 & 5 \\
1 & \infty & \infty & \infty & \infty & \infty \\
2 & \infty & \infty & 0 & 0.1 & 0 \\
3 & 2.25 & \infty & \infty & 0 & 0.45 \\
4 & 2.25 & \infty & 0 & \infty & 0.85 \\
5 & 0 & \infty & 1.7 & 1.7 & \infty
\end{array}$$

$r = 0.1 + 1.7 = 1.8$
$C(2) = C(1) + A(1,2) + r = 6.6 + 0.5 + 1.8 = 8.9$

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

$$\begin{array}{c|ccccc|}
 & 1 & 2 & 3 & 4 & 5 \\
1 & \infty & \infty & \infty & \infty & \infty \\
2 & 1.8 & \infty & \infty & 0.2 & 1.7 \\
3 & \infty & 0 & \infty & 0.1 & 2.15 \\
4 & 2.25 & 0.15 & \infty & \infty & 2.55 \\
5 & 0 & 1 & \infty & 1.8 & \infty
\end{array}$$

$R_2 - 0.2$; $R_4 - 0.15$;
$C_5 - 1.5$

$$\begin{array}{c|ccccc|}
 & 1 & 2 & 3 & 4 & 5 \\
1 & \infty & \infty & \infty & \infty & \infty \\
2 & 1.6 & \infty & \infty & 0 & 0 \\
3 & \infty & 0 & \infty & 0.1 & 0.65 \\
4 & 2.1 & 0 & \infty & \infty & 0.9 \\
5 & 0 & 1 & \infty & 1.8 & \infty
\end{array}$$

$r = 0.2 + 0.15 + 1.5 = 1.85$
$C(3) = C(1) + A(1,3) + r = 6.6 + 0.9 + 1.85 = 9.35$

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

$$\begin{array}{c|ccccc|}
 & 1 & 2 & 3 & 4 & 5 \\
1 & \infty & \infty & \infty & \infty & \infty \\
2 & 1.8 & \infty & 0 & \infty & 1.7 \\
3 & 2.25 & 0 & \infty & \infty & 2.15 \\
4 & \infty & 0.15 & 0 & \infty & 2.55 \\
5 & 0 & 1 & 1.7 & \infty & \infty
\end{array}$$

$C_5 - 1.7$

$$\begin{array}{c|ccccc|}
 & 1 & 2 & 3 & 4 & 5 \\
1 & \infty & \infty & \infty & \infty & \infty \\
2 & 1.8 & \infty & 0 & \infty & 0 \\
3 & 2.25 & 0 & \infty & \infty & 0.45 \\
4 & \infty & 0.15 & 0 & \infty & 0.85 \\
5 & 0 & 1 & 1.7 & \infty & \infty
\end{array} = B$$

$r = 1.9$
$C(4) = C(1) + A(1,4) + r = 6.6 + 0 + 1.7 = 8.3$

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

$$\begin{array}{c|ccccc|}
 & 1 & 2 & 3 & 4 & 5 \\
1 & \infty & \infty & \infty & \infty & \infty \\
2 & 1.8 & \infty & 0 & 0.2 & \infty \\
3 & 2.25 & 0 & \infty & 0.1 & \infty \\
4 & 2.25 & 0.15 & 0 & \infty & \infty \\
5 & \infty & 1 & 1.7 & 1.8 & \infty
\end{array}$$

$R_5 - 1$;
$C_1 - 1.8$; $C_4 - 0.1$

$$\begin{array}{c|ccccc|} & 1 & 2 & 3 & 4 & 5 \\ 1 & \infty & \infty & \infty & \infty & \infty \\ 2 & 0 & \infty & 0 & 0.1 & \infty \\ 3 & 0.45 & 0 & \infty & 0 & \infty \\ 4 & 0.45 & 0.15 & 0 & \infty & \infty \\ 5 & \infty & 0 & 0.7 & 0.7 & \infty \end{array}$$

r = 1 + 1.8 + 0.1 = 2.9
C(5) = C(1) + A(1,5) + r = 6.6 + 0 + 2.9 = 9.5

Langkah ketiga : Langkah yang sama dengan langkah kedua, namun untuk matriks B. Lakukan reduksi matriks B dan hitung cost-nya. Matriks yang memiliki cost terendah akan menjadi matriks C. Ubah batas bawah total bobot minimum menjadi cost terendah hasil reduksi matriks B.

- Reduced Cost Matrix (6) : Rute 1 – 4 – 2

$$\begin{array}{c|ccccc|} & 1 & 2 & 3 & 4 & 5 \\ 1 & \infty & \infty & \infty & \infty & \infty \\ 2 & \infty & \infty & 0 & \infty & 0 \\ 3 & 2.25 & \infty & \infty & \infty & 0.45 \\ 4 & \infty & \infty & \infty & \infty & \infty \\ 5 & 0 & \infty & 1.7 & \infty & \infty \end{array}$$

$R_3 - 0.45$;

$$\begin{array}{c|ccccc|} & 1 & 2 & 3 & 4 & 5 \\ 1 & \infty & \infty & \infty & \infty & \infty \\ 2 & \infty & \infty & 0 & \infty & 0 \\ 3 & 2 & \infty & \infty & \infty & 0 \\ 4 & \infty & \infty & \infty & \infty & \infty \\ 5 & 0 & \infty & 1.7 & \infty & \infty \end{array}$$

r = 0.25
C(6) = C(4) + B(4,2) + r = 8.3 + 0.15 + 0.45 = 8.9

- Reduced Cost Matrix (7) : Rute 1 – 4 – 3

$$\begin{array}{c|ccccc|l} & 1 & 2 & 3 & 4 & 5 & \\ 1 & \infty & \infty & \infty & \infty & \infty & \\ 2 & 1.8 & \infty & \infty & \infty & 0 & \\ 3 & \infty & 0 & \infty & \infty & 0.45 & = C \\ 4 & \infty & \infty & \infty & \infty & \infty & \\ 5 & 0 & 1 & \infty & \infty & \infty & \end{array}$$

r = 0
C(7) = C(4) + B(4,3) + r = 8.3 + 0 + 0 = 8.3

- Reduced Cost Matrix (8) : Rute 1 – 4 – 5

$$\begin{array}{c|ccccc|} & 1 & 2 & 3 & 4 & 5 \\ 1 & \infty & \infty & \infty & \infty & \infty \\ 2 & 1.8 & \infty & 0 & \infty & \infty \\ 3 & 2.25 & 0 & \infty & \infty & \infty \\ 4 & \infty & \infty & \infty & \infty & \infty \\ 5 & \infty & 1 & 1.7 & \infty & \infty \end{array}$$

$R_5 - 1$;
$C_1 - 1.8$

$$\begin{array}{c|ccccc|} & 1 & 2 & 3 & 4 & 5 \\ 1 & \infty & \infty & \infty & \infty & \infty \\ 2 & 0 & \infty & 0 & \infty & \infty \\ 3 & 0.45 & 0 & \infty & \infty & \infty \\ 4 & \infty & \infty & \infty & \infty & \infty \\ 5 & \infty & 0 & 0.7 & \infty & \infty \end{array}$$

r = 1 + 1.8 = 2.8
C(8) = C(4) + B(4,5) + r = 8.3 + 0.65 + 2.8 = 11.75

Langkah keempat : Langkah yang sama dengan langkah kedua, namun untuk matriks C. Lakukan reduksi matriks C dan hitung cost-nya. Matriks yang memiliki cost terendah akan menjadi matriks D. Ubah batas bawah total bobot minimum menjadi cost terendah hasil reduksi matriks C.

- Reduced Cost Matrix (9) : Rute 1 – 4 – 3 – 2

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

r = 0
C(9) = C(7) + C(3,2) + r = 8.3 + 0 + 0 = 8.3

- Reduced Cost Matrix (10) : Rute 1 – 4 – 3 – 5

$$\begin{array}{c|ccccc|}
 & 1 & 2 & 3 & 4 & 5 \\
1 & \infty & \infty & \infty & \infty & \infty \\
2 & 1.8 & \infty & \infty & \infty & \infty \\
3 & \infty & \infty & \infty & \infty & \infty \\
4 & \infty & \infty & \infty & \infty & \infty \\
5 & \infty & 1 & \infty & \infty & \infty
\end{array}$$

R_2 – 1.8; R_5 – 1;

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

r = 1.8 + 1 = 2.8
C(10) = C(7) + C(3,5) + r = 8.3 + 0.25 + 2.8 = 11.35

Langkah kelima : Hasil reduksi dari matriks D sudah tidak bisa direduksi lagi sehingga batas bawah total bobot minimum adalah cost hasil reduksi matriks D.

- Reduced Cost Matrix (11) : Rute 1 – 4 – 3 – 2 – 5

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

r = 0
C(11) = C(9) + D(2,5) + r = 8.5 + 0 + 0 = 8.3

D. Hasil

Berdasarkan hasil pencarian rute terpendek dengan menggunakan algoritma *branch and bound*, didapatkan rute roadshow terpendeknya adalah rute sekolah 1 – 4 – 3 – 2 – 5 – 1 , yaitu SMA Negeri 1 Pekanbaru – Madrasah Aliyah Negeri 2 Kota Pekanbaru – SMA Negeri 8 Pekanbaru – SMA Santa Maria Pekanbaru – Sekolah Menengah Umum Kristen Kalam Kudus – SMA Negeri 1 Pekanbaru dengan total jarak tempuh 8,3 km.

IV. KESIMPULAN

Graf dapat diaplikasikan untuk menyelesaikan banyak permasalahan, salah satunya adalah untuk mencari rute terpendek ketika mengunjungi beberapa tempat tepat satu kali hingga kembali ke tempat awal atau yang biasa diebut dengan *Travelling Salesman Problem* (TSP). Salah satu cara untuk menyelesaikan permasalahan TSP adalah dengan menggunakan algoritma *branch and bound*.

Penerapan *Travelling Salesman Problem* dengan algoritma *branch and bound* dicontohkan dalam makalah ini untuk mencari rute roadshow terpendek dari sebuah acara yang bernama ITB on Journey to Riau-Kepri. Dengan mengoptimasi rute perjalanan, maka waktu yang diperlukan dalam perjalanan juga akan menjadi lebih singkat sehingga dengan keterbatasan jumlah kelompok roadshow, tetap dapat mengunjungi sejumlah tertentu sekolah dalam hari yang sama dengan lebih efisien.

V. UCAPAN TERIMA KASIH

Puji syukur kepada Tuhan Yang Maha Esa atas berkat dan rahmatnya, makalah ini dapat terselesaikan dengan tepat waktu. Penulis menyampaikan terima kasih kepada semua pihak yang telah mendukung studi dan proses pembelajaran dalam mata kuliah IF2120 Matematika Diskrit. Penulis juga mengucapkan terima kasih terutama kepada dosen pengampu mata kuliah IF2120 Matematika Diskrit yaitu, Bapak Dr. Rinaldi Munir, Ibu Harlili, M.Sc, Ibu Fariska Zakhralativa, dan Ibu Nur Ulfa Maulidevi, atas ilmunya yang telah diberikan selama semester 1 tahun ajaran 2020/2021.

REFERENSI

[1] Munir, Rinaldi. Graf Bagian 1. Informatika, Bandung: 2020. Diakses tanggal 7 Desember 2020.

[2] Munir, Rinaldi. Graf Bagian 2. Informatika, Bandung: 2020. Diakses tanggal 7 Desember 2020.

[3] Munir, Rinaldi. Graf Bagian 3. Informatika, Bandung: 2020. Diakses tanggal 7 Desember 2020.

[4] Prasetyo, Yogo. penyelesaian travelling salesman problem dengan algoritma branch and bound. Diakses tanggal 8 Desember 2020.

[5] TechieDelight. Travelling Salesman Problem using Branch and Bound. Diakses tanggal 8 Desember 2020.

[6] Munir, Rinaldi. Algoritma Branch and Bound Bagian 1. Informatika, Bandung: 2020. Diakses tanggal 9 Desember 2020.

[7] Munir, Rinaldi. Algoritma Branch and Bound Bagian 2. Informatika, Bandung: 2020. Diakses tanggal 9 Desember 2020.

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 2020

Christopher Justine William
13519006