Penerapan TSP pada Pencarian Rute Pengunjungan Minimum Stasiun Akhir KRL Commuterline

Ferdian Ifkarsyah - 13517024
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13517024@std.stei.itb.ac.id

***Abstrak*—Salah satu moda transportasi favorit masyarakat Jabodetabek untuk berkegiatan adalah KRL Commuterline. Mayoritas masyarakat menggunakan Commuterline untuk pergi dan pulang kerja dengan satu buah rute yang selalu sama. Namun, bagi sebagian orang, mengelilingi Jakarta atau bahkan Jabodetabek dengan KRL Commuterline adalah hiburan yang cukup menyenangkan di akhir pekan. Makalah ini akan membahas representasi jalur KRL Commuterline dengan graf dan pencarian rute terpendek untuk mengunjungi semua stasiun akhir KRL Commuterline Jabodetabek.**

***Keywords*—KRL, Commuterline, problem, graf, pohon**

I. Pendahuluan

KRL Commuterline adalah sistem pelayanan kereta rel listrik yang menghubungi lima kota besar di Indonesia, yaitu Jakarta, Bogor, Kota Depok, Tangerang dan Bekasi atau lebih dikenal dengan singkatan Jabodetabek. KRL Commuterline dioperasikan oleh PT Kereta Commuter Indonesia(KCI) yang merupakan anak perusahaan dari PT Kereta Api Indonesia(KAI).

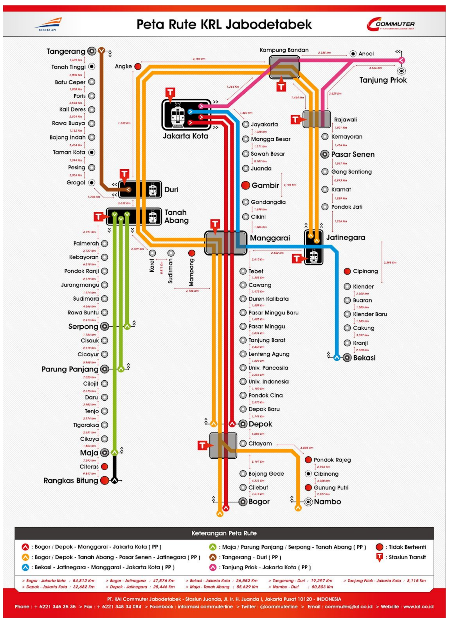
Rangkaian rel Commuterline adalah salah satu yang terpanjang di dunia dengan panjang mencapai 418 km. Selain itu, KRL Commuterline adalah salah satu moda transportasi kota yang terpadat di dunia dengan jumlah rata-rata penumpang per hari pada hari kerja mencapai 1 juta lebih penumpang.

Sebelum masa KRL Commuterline, PT KAI melayani masyarakat Jabodetabek dengan KRL Jabodetabek. Pada masa tersebut, pelayanan KRL dibagi menjadi dua kelas : KRL Ekonomi(lebih murah, tanpa pendingin ruangan) dan KRL Ekspres(lebih mahal, dengan pendingin ruangan).

Namun, karena alasan kepadatan yang tidak wajar dan banyaknya penumpang ilegal yang naik ke atas kereta(yang tentunya sangat membahayakan bagi diri mereka sendiri), KRL Jabodetabek berhenti pada tahun 2013 dan digantikan KRL Commuterline. Penggantian ini ditandai dari hilangnya kasta kelas gerbong ekonomi--ekspres dan modernisasi sistem pembelian tiket dengan kartu tap menggantikan sistem pembelian tiket dengan kertas.

Penggantian ini membuat KRL Commuterline jauh lebih nyaman digunakan. Hal ini, berdampak pada bertambahnya manfaat KRL untuk masyarakat Jakarta. Meskipun, kegunaan utamanya tetap untuk pergi dan pulang kerja dari pinggir ke pusat Jakarta. Tapi, untuk beberapa orang, menaiki KRL Commuterline untuk mengelilingi Jabodetabek adalah hiburan yang cukup menyenangkan. Apalagi bagi mereka yang berjiwa petualang, mungkin sangat penasaran dengan keadaan 6 stasiun akhir atau ujung dari KRL Commuterline. Keenam stasiun ini adalah Stasiun Tangerang, Stasiun Tanjung Priok, Stasiun Rangkasbitung, Stasiun Bogor, Stasiun Nambo, dan Stasiun Cikarang.

Namun, dengan panjangnya jarak sistem rangkaian rel yang mencapai 418 km, dibutuhkan suatu cara untuk menentukan urutan stasiun-stasiun ini yang harus dikunjungi agar jarak yang ditempuh menjadi minimum. Di sini lah, penyelesaian *Travelling Salesman Problem* dapat dipakai untuk menentukan urutan stasiun yang harus dikunjungi tersebut.



Gambar 1.1 Peta Rute KRL Commuterline Jabodetabek
(Sumber : *krl.co.id*)

II. Dasar Teori

A. Graf

Secara matematis, graf didefinisikan sebagai berikut :

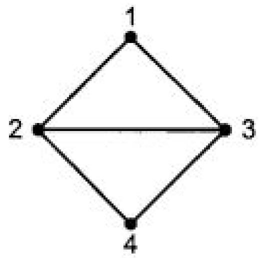
Graf G didefinisikan sebagai pasangan himpunan (V,E), ditulis dengan notasi G = (V,E) dengan V adalah himpunan tidak-kosong dari simpul-simpul (*vertices* atau *node*) dan E adalah himpunan sisi (*edges* atau *arcs)* yang menghubungkan sepasang simpul.

B. Jenis-Jenis Graf

Graf dapat dikategorikan menjadi beberapa jenis bergantung pada kriteria pengkategoriannya. Pengkategorian graf dapat berdasarkan ada tidaknya sisi ganda, berdasarkan jumlah simpul, atau berdasarkan ada tidaknya orientasi sisi.

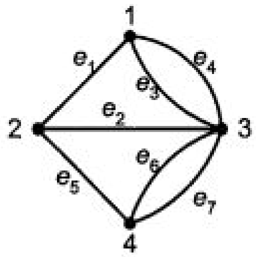
Berdasarkan ada tidaknya sisi ganda, graf dapat digolongkan menjadi dua jenis:

1. Graf Sederhana
   Yaitu, graf yang tidak mengandung gelang maupun sisi-ganda.



Gambar 2.1. Graf sederhana
(Sumber : Matematika Diskrit - Rinaldi Munir)

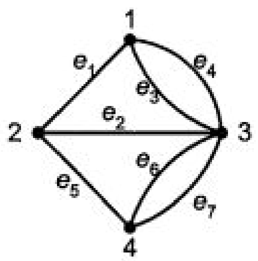
2. Graf Tak Sederhana
   Yaitu, graf yang mengandung gelang atau sisi-ganda.



Gambar 2.2 Graf tak-sederhana
(Sumber : Matematika Diskrit - Rinaldi Munir)

Berdasarkan ada tidaknya orientasi sisi, graf dapat digolongkan menjadi dua jenis:

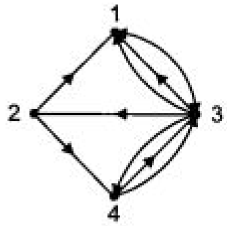
1. Graf Tak Berarah



Gambar 2.2 Graf tak-berarah

(Sumber : Matematika Diskrit - Rinaldi Munir)

2. Graf Berarah



Gambar 2.3 Graf berarah
(Sumber : Matematika Diskrit - Rinaldi Munir)

C. Terminologi Graf

Dalam teori graf, terdapat beberapa terminologi yang perlu diketahui.

1. Bertetangga
   Dua buah simpul dikatakan bertetangga jika keduanya dihubungkan secara langsung oleh sebuah sisi. Secara formal dikatakan, simpul u bertetangga dengan simpul v jika (u,v) adalah sebuah sisi pada graf G.

2. Bersisian
   Untuk sembarang sisi $e=(u,v)$, sisi e dikatakan bersisian dengan simpul u dan simpul v.

3. Derajat
   Untuk sembarang simpul v, derajat dari v atau $d(v)$ didefinisikan sebagai banyak sisi e yang bersisian dengan simpul v.
   Dalam sembarang graf G, jumlah derajat dari semua simpul G adalah dua kali banyak sisi graf G. Secara matematis, jika $G=(V,E)$ maka

$$\sum_{v \varepsilon V} d(v) = 2|E|$$

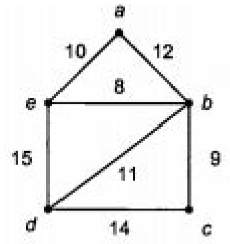
4. Lintasan
   Lintasan dengan panjang n dari simpul awal v_0 ke simpul tujuan v_n pada graf G adalah barisan sisi-sisi $e_1,e_2,e_3,...,e_n$, dengan $e_i=(v_{i-1},v_n)$.

5. Sirkuit
   Sirkuit adalah lintasan dengan $v_0 = v_n$.

6. Terhubung
   Dua buah simpul u dan v dikatakan terhubung apabila terdapat lintasan dengan simpul awal u dan simpul tujuan v.
   Graf yang setiap pasangan v_x dan v_y terhubung disebut graf terhubung.

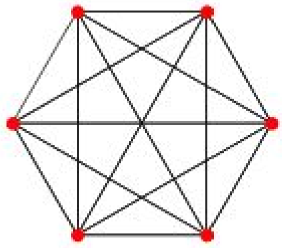
7. Upagraf
   Graf $H=(V_h,E_h)$ disebut upagraf dari graf $G=(V_g,E_g)$ jika $V_h \subseteq V_g$ dan $E_h \subseteq E_g$.

8. Graf Berbobot
   Graf berbobot adalah graf yang sisi-sisinya diberi sebuah nilai atau bobot.



Gambar 2.4 Graf berbobot

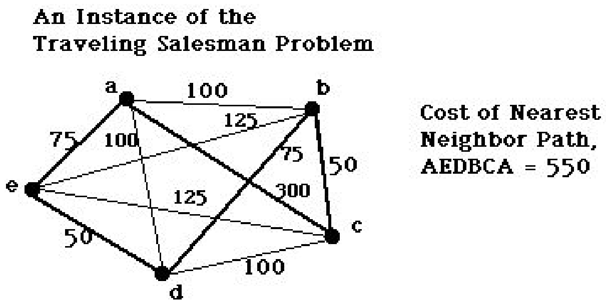
9. Graf Lengkap
   Graf yang setiap pasangan v_x dan v_y terhubung langsung(ada sisi (v_x,v_y)) disebut graf lengkap. Graf lengkap dengan *n* buah simpul disebut K_n.



Gambar 2.5 Graf K_6
(Sumber:*http://mathworld.wolfram.com/CompleteGraph.html*)

D. Travelling Salesman Problem

Travelling Salesman Problem atau disingkat *TSP* memiliki definisi masalah sebagai berikut : Diberikan sejumlah kota dan diketahui jarak setiap dua buah kota. Tentukan tur terpendek yang harus dilalui seorang pedagang(*salesman*) bila sebuah pedagang berangkat dari kota asal tepat satu kali dan kembali lagi ke kota asal.



Gambar 2.6 Contoh Solusi *TSP*(Sumber : *https://www.computing.dcu.ie*)

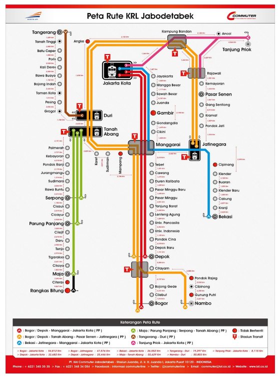
D. Shortest-Path Problem

Shortest-Path Problem adalah masalah pencarian lintasan terpendek pada dua buah simpul *u* dan *v* pada graf G tak-berarah terhubung.

III. Pencarian Rute Terpendek KRL Commuterline

A. Pemodelan Masalah

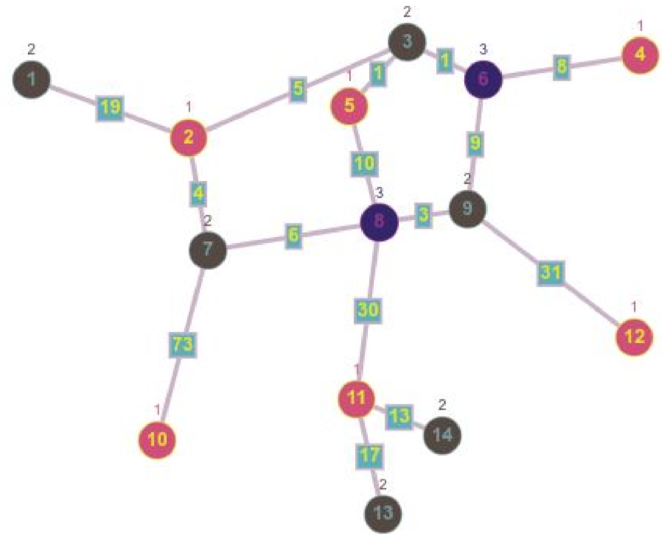
Setiap stasiun di rangkaian rel KRL Commuterline dapat diibaratkan dengan sebuah simpul graf G. Kemudian, setiap rel yang menghubungkan dua buah stasiun dapat diibaratkan sebagai sisi graf G. Jadi, sekarang kita mempunyai sebuah graf tak-berarah *G=(V,E)* dengan *V* adalah himpunan stasiun-stasiun dan *E* adalah himpunan rangkaian yang menghubungkan dua buah stasiun.



Gambar 3.1 Peta Rute KRL Jabodetabek

B. Penyederhanaan Masalah

Dikarenakan jumlah stasiun yang cukup banyak(mencapai 72), kita akan membatasi permasalahan hanya pada stasiun transit yang berjumlah 14 buah. Stasiun transit adalah stasiun dimana terdapat pergantian jalur KRL Commuterline, misalnya Stasiun Manggarai, Stasiun Tanah Abang, dan lainnya. Selain karena alasan komputasi, pembatasan ini cukup masuk akal karena jika kita bepergian dari sebuah stasiun bukan-transit ke stasiun transit, dengan tujuan untuk mengunjungi semua stasiun, itu sama saja dengan kita bepergian dari stasiun transit sebelumnya ke stasiun transit tujuan. Jadi, graf G kita akan seperti yang digambarkan pada gambar 3.2.



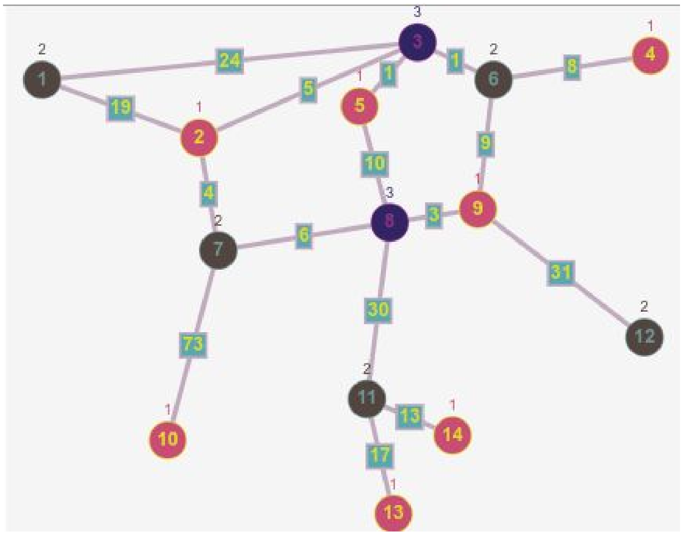
Gambar 3.2 Graf tak-berarah terhubung rangkaian KRL Commuterline, dengan simpul adalah stasiun dan bobot sisi adalah panjang rel dalam kilometer.
(Dibuat dengan graphonline.ru/en/)

Keterangan nomor simpul:

1. Stasiun Tangerang
2. Stasiun Duri
3. Stasiun Kampung Bandan
4. Stasiun Tanjung Priok
5. Stasiun Jakarta Kota
6. Stasiun Rajawali
7. Stasiun Tanah Abang
8. Stasiun Manggarai
9. Stasiun Jatinegara
10. Stasiun Rangkasbitung
11. Stasiun Citayam
12. Stasiun Bogor
13. Stasiun Nambo

Untuk membuat masalah pencarian rute minimum ini dapat diselesaikan dengan penyelesaian *Travelling Salesman Problem,* graf G pada gambar 3.2 harus dibuat menjadi graf lengkap.

Cara melengkapkan graf G dilakukan dengan memasangkan dua buah simpul *u* dan *v* dalam G yang tidak bertetangga dengan sebuah sisi *(u,v)*. Bobot *(u,v)* adalah *shortest path* atau jarak terpendek antara simpul *u* dan *v*. Diasumsikan, kita telah mengimplementasikan fungsi untuk mencari *shortest path*. Contohnya, untuk menghubungkan simpul 1 dan 3 pada graf G, kita dapat menambahkan sisi *(1,3)* dengan bobot 24 yang merupakan *shortest path* antara 1 dan 3.



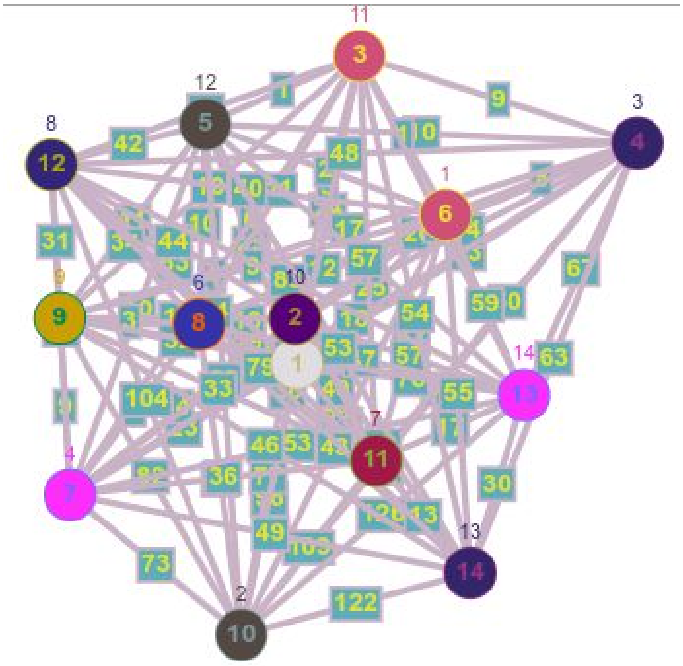
Gambar 3.3 Graf G yang sudah ditambahkan sisi *(1,3)*
(Dibuat dengan graphonline.ru/en/)

Pseudocode dari algoritma melengkapkan graf tersebut dijelaskan sebagai berikut(diasumsikan kita telah memiliki fungsi *shortestPath(u,v)*) :

```
function LengkapkanGraf(G=(V,E));
  for u in V
    for v in V
      if (u,v) not in E then
        weight=shortestPath(u,v)
        InsertEdge(E,(u,v,weigth))
```

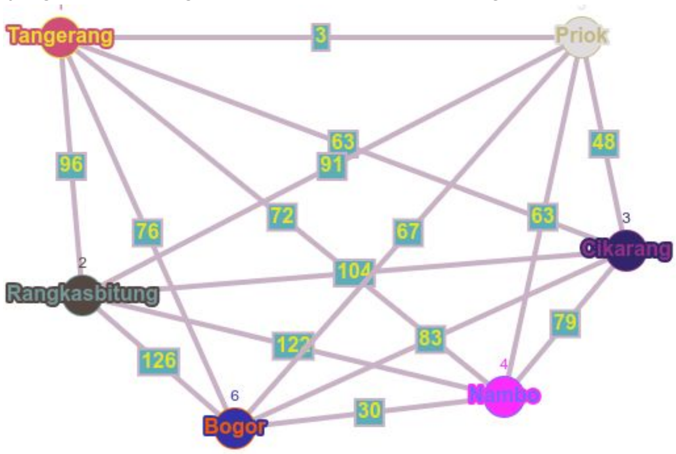
Gambar 3.4 Algoritma melengkapkan graf G

Setelah semua simpul pada graf G saling bertetangga, graf G yang sudah dilengkapkan akan menjadi K_{14}.



Gambar 3.5 Graf G yang sudah dilengkapkan

Kemudian, karena dalam persoalan ini, kita hanya mementingkan stasiun yang merupakan stasiun akhir atau final, maka kita perlu menghilangkan beberapa simpul graf lengkap G di atas dan menyisakan simpul yang merepresentasikan stasiun ujung. Dari graf G di gambar 3.5, kita hanya akan memakai simpul 1,4,10,13,14, dan 12 yang merepresentasikan secara berturut-turut St. Tangerang, St. Tanjung Priok, St. Bogor, St. Nambo, dan St. Cikarang.



Gambar 3.6 Graf G yang hanya merepresentasikan stasiun akhir

Untuk mempermudah pengerjaan program, kita menomori ulang graf G menjadi seperti berikut :

1. Stasiun Tangerang
2. Stasiun Tanjung Priok
3. Stasiun Rangkasbitung
4. Stasiun Bogor
5. Stasiun Nambo
6. Stasiun Cikarang

Representasi matriks ketetanggan graf G gambar 3.6 diberikan sebagai berikut :

[[0, 3, 96, 76, 72, 63] ,
[3, 0, 91, 67, 63, 48],
[96, 91, 0, 126, 122, 104],
[76, 67, 126, 0, 30, 83] ,
[72, 63, 122, 30, 0, 79] ,
[63, 48, 104, 83, 79, 0]]

Inilah graf yang akan digunakan untuk merepresentasikan rencana perjalanan seseorang yang akan mengunjungi semua stasiun ujung KRL Commuterline.

D. Menjalankan TSP pada graf G

Algoritma TSP yang digunakan pada graf G menggunakan teknik *Dynamic Programming*. Algoritma ini memiliki kompleksitas **O**(2^n*n^2). Algoritma TSP yang dipakai diberikan sebagai berikut :

```
#include<stdio.h>

int rail[10][10];
int visited[10]={0};
int n,distance=0;
const int MAX = 99999;

void inputStasiun();
// menginput graf adjacency matrix G
int least(int c);
// mengambil data dari memo
void mincost(int station);
// proses DP dan memoisasi

int main(){
    inputStasiun();
    printf("\n\nRute:\n");
    int startStation = 0;
    mincost(startStation);
    printf("\n\nJarak min %d\n ",distance);

}
```

```
void inputStasiun(){
    int i,j;
    printf("Masukkan jumlah stasiun: ");
    scanf("%d",&n);
    printf("\nMasukkan Graf : \n");
    for(i=0;i < n;i++)
        for( j=0;j < n;j++)
         scanf("%d",&rail[i][j]);
}

void mincost(int station){
    int i,nstation;
    visited[station]=1;
    printf("%d--->",station+1);
    nstation=least(station);
    if(nstation==MAX){
        nstation=0;
        printf("%d",nstation+1);
        distance+=rail[station][nstation];
        return;
    }
    mincost(nstation);
}

int least(int c){
    int i,nc=MAX;
    int min=MAX,kmin;
    for(i=0;i < n;i++){
        if((rail[c][i]!=0)&&(visited[i]==0))
            if(rail[c][i]+rail[i][c] < min){
                min=rail[i][0]+rail[c][i];
                kmin=rail[c][i];
                nc=i;
            }
    }
    if(min!=MAX) distance+=kmin;
    return nc;
}
```

Gambar 3.7 Algoritma TSP untuk masalah pencarian rute pengunjungan stasiun akhir KRL(Sumber algoritma : https://www.thecrazyprogrammer.com/2017/05/travelling-salesman-problem.html)

Karena program ini hanya dapat memberikan TSP dari stasiun simpul 1, kita dapat mengakalinya dengan memutar penomoran simpul sebanyak 6 kali.

Pertama : 1-2-3-4-5-6(Tangerang--1,Bogor--4,Cikarang--6, semua tetap)

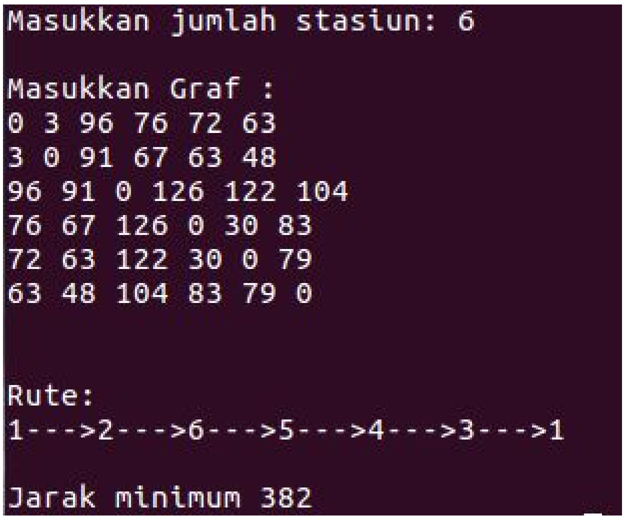
Kedua : 2-3-4-5-6-1(Priok(2)--dalam program dianggap 1, Rangkasbitung(3)--dalam program dianggap 2, dan seterusnya).

Ketiga : 3-4-5-6-1-2(Rangkasbitung(3)--dalam program dianggap 1, Bogor(4)--dalam program dianggap 4, dan seterusnya).

Hal yang sama berlaku sampai penomoran terakhir menjadi 6-1-2-3-4-5

IV. Hasil dan Pembahasan

A. Hasil



```
Masukkan jumlah stasiun: 6

Masukkan Graf :
0 3 96 76 72 63
3 0 91 67 63 48
96 91 0 126 122 104
76 67 126 0 30 83
72 63 122 30 0 79
63 48 104 83 79 0


Rute:
1--->2--->6--->5--->4--->3--->1

Jarak minimum 382
```

Gambar 4.1 Hasil pengeksekusian program, dengan stasiun awal adalah simpul 1 atau St. Tangerang

Hasil lengkap dari program diberikan dengan tabel berikut :

Stasiun Awal	Rute(berdasar penomoran awal kita(bukan program))	Jarak(kilometer)
1 -- Tangerang	1-2-6-5-4-3-1	382
2 -- Tanjung Priok	2-5-4-2-3-1-2	392
3 -- Rangkasbitung	3-2-1-5-6-4-3	480
4 -- Bogor	4-5-2-6-3-1-4	382
5 -- Nambo	5-4-3-1-2-6-5	382
6 -- Cikarang	6-2-1-3-4-5-6	366

Tabel 4.1 Hasil eksekusi TSP dari keenam kota asal

B. Pembahasan

Dari tabel 4.1, dapat dilihat bahwa jarak terpendek didapatkan oleh masyarakat Cikarang yang dapat seluruh stasiun akhir dengan rute St.Cikarang-St.Tanjung Priok-St.Tangerang-St.Rangkasbitung-St.Bogor-St.Nambo dan kembali ke St.Cikarang yang memiliki panjang 366 kilometer.

Sementara, nasib malang menghadiri masyarakat yang memilih St.Rangkasbitung sebagai poin awal pemberangkatan karena harus menempuh jarak 480 kilometer dengan rute St.Rangkasbitung-St.Tanjung Priok-St.Tangerang-St.Nambo-St.Cikarang-St.Bogor dan kembali ke St.Rangkasbitung.

VI. Kesimpulan

Semua objek atau lokasi yang ada di muka bumi dapat direpresentasikan menjadi graf dengan suatu objek atau lokasi dapat disimbolkan dengan sebuah simpul graf dan jalan antara dua buah objek disimbolkan dengan sebuah sisi.

Dengan adanya representasi simpul dan sisi ini, kita dapat melakukan berbagai hal. Salah satunya adalah menentukan jarak minimum untuk mengunjungi semua objek atau lokasi dalam suatu daerah.

Dalam makalah ini, telah diperlihatkan bahwa rangkaian sistem KRL dapat direpresentasikan dengan graf. Lalu, graf hasil representasi tersebut dapat dimanfaatkan orang yang ingin mengunjungi semua stasiun akhir di rangkaian dengan stasiun awal adalah stasiun yang paling dekat dengan rumahnya.

Dalam perhitungan, terlihat bahwa orang yang paling beruntung adalah yang memiliki rumah dekat dengan Stasiun Cikarang karena akan menempuh jarak terpendek, 366 kilometer. Sedangkan orang paling tidak beruntung adalah mereka yang berangkat dari Stasiun Rangkasbitung.

VII. Ucapan Terima Kasih

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada Ibu Harlili, Pak Rinaldi Munir, dan Pak Judhi Santoso selaku dosen mata kuliah IF2120 -- Matematika Diskrit yang telah rela memberikan ilmu serta pengalaman selama satu semester ini. Penulis juga berterima kasih kepada teman-teman mahasiswa yang menjadi lawan bicara untuk bertukar pikiran sehingga penulis mendapat banyak perspektif baru selama penulisan makalah ini. Terakhir, penulis juga berterima kasih kepada para penulis buku atau artikel yang tercantum dalam referensi di bawah. Akhir kata, penulis berharap dengan adanya makalah ini dapat memberi manfaat yang sebanyak-banyaknya kepada masyarakat luas.

References

[1] Munir, Rinalid. *Matematika Diskrit.* Bandung:ITB,2006.

[2] Rosen, Kenneth H. 2006. *Discrete Mathematics and Its Applications, 6th Edition. Singapore: McGraw-Hill.*

[3] krl.co.id
Tanggal akses: 8 Desember 2018, pukul 10:30 WIB

[4] https://www.geeksforgeeks.org/traveling-salesman-problem-tsp-implementation/
Tanggal akses: 9 Desember 2018, pukul 8.30 WIB

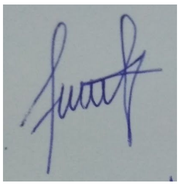
[5] https://codingblocks.com/resources/travelling-salesman/
Tanggal akses: 9 Desember 2018, pukul 14.30 WIB

[6] https://www.thecrazyprogrammer.com/2017/05/travelling-salesman-problem.html
Tanggal akses: 9 Desember 2018, pukul 15.20 WIB

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, 9 Desember 2018



Ferdian Ifkarsyah
13517024