

# Aplikasi Pohon Merentang Minimum dalam Pembuatan Jalur Trem di Jakarta Pusat

Muhammad Firas | 13518117<sup>1</sup>

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

<sup>1</sup>13518117@std.stei.itb.ac.id

**Abstrak**—Kemacetan merupakan masalah utama di DKI Jakarta. Jumlah kendaraan pribadi yang semakin meningkat seiring dengan meningkatnya kepadatan penduduk serta kurangnya transportasi umum merupakan penyebab kemacetan di DKI Jakarta. Salah satu solusinya adalah penambahan transportasi umum di dalam kota termasuk trem. Akan tetapi, dengan ditambahnya transportasi umum maka perlu juga dibuatnya jalur khusus untuk transportasi umum tersebut. Oleh karena itu, Makalah ini akan membahas bagaimana suatu penerapan pohon merentang minimum untuk membuat jalur trem di DKI Jakarta terutama Jakarta Pusat. Pohon merentang minimum ini digunakan agar pembuatan jalur trem dapat mengeluarkan biaya seminimum dan seefektif mungkin.

**Kata kunci**—Algoritma Kruskal, Algoritma Prim, Graf, Jakarta Pusat, Pohon Merentang Minimum, Trem.

## I. PENDAHULUAN

Jadwal yang padat di kota besar dan padatnya penduduk merupakan salah satu penyebab kemacetan. Kemacetan dapat menjadi masalah di kota besar seperti Jakarta. Kemacetan di Jakarta sudah terjadi sejak tahun 1965 dan sampai sekarang masih sulit untuk diatasi. Berbagai usaha telah dilakukan pemerintah untuk mengatasi kemacetan ini seperti mengadakan Hari Bebas Kendaraan Bermotor (HBKB) atau yang biasa disebut *Car-Free Days*. Selain Hari Bebas Kendaraan Bermotor usaha lain yang dilakukan pemerintah adalah meningkatkan transportasi umum, contohnya adalah trem.

Trem adalah transportasi umum berbentuk kereta ringan yang memiliki jalur khusus di dalam kota. Letak rel trem ini dapat berbaur di antara lalu-lintas kota atau terpisah seperti bus-way. Trem banyak ditemukan di negara-negara maju terutama negara-negara di Eropa. Trem merupakan salah satu transportasi umum yang efektif dalam mengatasi kemacetan. Kebanyakan trem pada masa sekarang menggunakan tenaga listrik untuk menggerakannya.

Sejak zaman kolonialisme Belanda, Jakarta telah mengalami berbagai macam perubahan transportasi umum yang dipengaruhi oleh pertambahan jumlah penduduk, kemajuan teknologi, dan kebijakan pemerintah. Transportasi umum ini termasuk trem yang dulu pertama kali ada di Jakarta pada tahun 1869. Trem di Batavia (Jakarta) dimulai dengan sebuah trem kuda, yang kemudian berkembang menjadi jaringan trem uap hingga lanjut berkembang menjadi trem listrik. Namun,

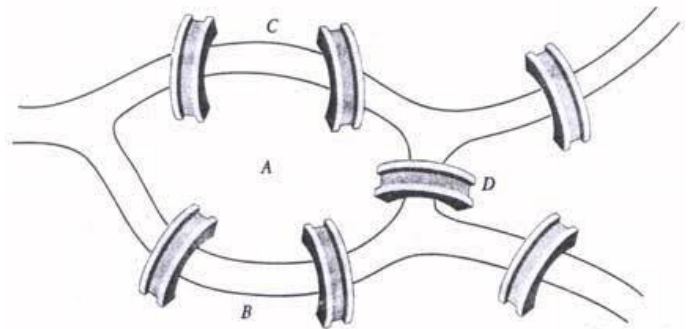
setelah Indonesia merdeka, jaringan trem ini tidak lagi digunakan dan pada akhirnya diputus. Namun, bagaimana jika trem bertenaga listrik dibangun lagi di Jakarta terutama Jakarta Pusat? Untuk pembangunan jalur trem tersebut tentunya menggunakan biaya yang tidak sedikit. Oleh karena itu, digunakan aplikasi teori graf, yaitu pohon merentang minimum untuk meminimalisir biaya yang dikeluarkan untuk pembuatan jalur trem di Jakarta Pusat dalam mengatasi kemacetan.

## II. LANDASAN TEORI

### A. Teori Graf

#### 1. Definisi Graf

Graf adalah suatu representasi objek-objek diskrit dan hubungan antara objek-objek tersebut. Teori graf ini awalnya dikemukakan oleh seorang ahli matematika yang berasal dari Swiss bernama Leonhard Euler yang memecahkan masalah jembatan Königsberg.



Gambar 1. Masalah jembatan Königsberg  
Sumber : slide Teori graf Matematika Diskrit Rinaldi Munir (2015)

Graf  $G$  dapat didefinisikan sebagai pasangan himpunan  $G = (V, E)$  dengan

$$V = \{v_1, v_2, v_3, \dots, v_n\}$$

$V$  adalah himpunan tidak-kosong dari simpul-simpul (*vertices*)

dan

$$E = \{e_1, e_2, e_3, \dots, e_n\}$$

$E$  adalah himpunan sisi (*edges*) yang menghubungkan

sepasang simpul

2. Jenis-Jenis Graf

Graf dapat dibedakan berdasarkan ada tidaknya sisi ganda pada suatu graf dan berdasarkan orientasi arah pada sisi. Berdasarkan ada tidaknya sisi ganda, graf dapat dibedakan menjadi dua jenis:

1. **Graf sederhana (simple graph)**

Graf sederhana adalah graf yang tidak mengandung gelang maupun sisi-ganda.

2. **Graf tak-sederhana (unsimple graph)**

Graf tak-sederhana adalah graf yang mengandung sisi ganda atau gelang. Graf ganda dan graf semu merupakan contoh graf tak-sederhana.

Sedangkan graf yang berdasarkan orientasi arah pada sisi, dapat dibedakan atas dua jenis:

1. **Graf tak-berarah (undirected graph)**

Graf tak berarah adalah graf yang sisinya tidak mempunyai orientasi arah.

2. **Graf berarah (directed graph atau digraph)**

Graf berarah adalah graf yang setiap sisinya diberikan orientasi arah.

Untuk memperjelas perbedaan graf-graf tersebut, perhatikan tabel berikut:

Jenis	Sisi	Sisi ganda dibolehkan?	Sisi gelang dibolehkan?
Graf Sederhana	Tak-berarah	Tidak	Tidak
Graf ganda	Tak-berarah	Ya	Tidak
Graf semu	Tak-berarah	Ya	Ya
Graf berarah	Berarah	Tidak	Ya
Graf ganda berarah	Berarah	Ya	Ya

Tabel 1. Jenis-jenis graf

Sumber : slide Teori Graf Matematika Diskrit Rinaldi Munir (2015)

3. Terminologi Graf (Munir, 2015)

a. **Ketetanggaan (Adjacent)**

Dua buah simpul dikatakan bertetangga bila keduanya terhubung langsung.

b. **Bersisian (Incidency)**

Untuk sembarang sisi  $e = (v_j, v_k)$  dikatakan  $e$  bersisian dengan simpul  $v_j$ , atau  $e$  bersisian dengan simpul  $v_k$

c. **Simpul Terpencil (Isolated Vertex)**

Simpul terpencil ialah simpul yang tidak mempunyai sisi yang bersisian dengannya.

d. **Graf Kosong (null graph atau empty graph)**

Graf kosong adalah graf yang himpunan sisinya merupakan himpunan kosong ( $N_n$ ).

e. **Derajat (Degree)**

Derajat suatu simpul adalah jumlah sisi yang bersisian dengan simpul tersebut.

f. **Lintasan (Path)**

Lintasan yang panjangnya  $n$  dari simpul awal  $v_0$  ke simpul tujuan  $v_n$  dalam graf  $G$  ialah 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  $e_1 = (v_0, v_1), e_2 = (v_1, v_2), \dots, e_n = (v_{n-1}, v_n)$  adalah sisi-sisi dari graf  $G$ .

g. **Siklus (Cycle) atau Sirkuit (Circuit)**

Lintasan yang berawal dan berakhir pada simpul yang sama disebut sirkuit atau siklus.

Panjang sirkuit adalah jumlah sisi dalam suatu sirkuit.

h. **Terhubung**

Dua buah simpul  $v_1$  dan simpul  $v_2$  disebut terhubung jika terdapat lintasan dari  $v_1$  ke  $v_2$ .

$G$  disebut graf terhubung (connected graph) jika untuk setiap pasang simpul  $v_i$  dan  $v_j$  dalam himpunan  $V$  terdapat lintasan dari  $v_i$  ke  $v_j$ . Jika tidak, maka  $G$  disebut graf tak-terhubung (disconnected graph).

i. **Upagraf (Subgraph) dan Komplemen Upagraf**

Misalkan  $G = (V, E)$  adalah sebuah graf.  $G_1 = (V_1, E_1)$  adalah upagraf (subgraph) dari  $G$  jika  $V_1 \subseteq V$  dan  $E_1 \subseteq E$ .

Komplemen dari upagraf  $G_1$  terhadap  $G$  adalah graf  $G_2 = (V_2, E_2)$  sedemikian sehingga  $E_2 = E - E_1$  dan  $V_2$  adalah himpunan simpul yang anggota-anggota  $E_2$  bersisian dengannya.

Komponen graf (connected component) adalah jumlah maksimum upagraf terhubung dalam graf  $G$ .

j. **Upagraf Rentang (Spanning Subgraph)**

Upagraf  $G_1 = (V_1, E_1)$  dan  $G = (V, E)$  dikatakan upagraf rentang jika  $V_1 = V$  (yaitu  $G_1$  mengandung semua simpul dari  $G$ ).

k. **Cut-Set**

Cut-set dari graf terhubung  $G$  adalah himpunan sisi yang bila dibuang dari  $G$  menyebabkan  $G$  tidak terhubung. Jadi, cut-set selalu menghasilkan dua buah komponen.

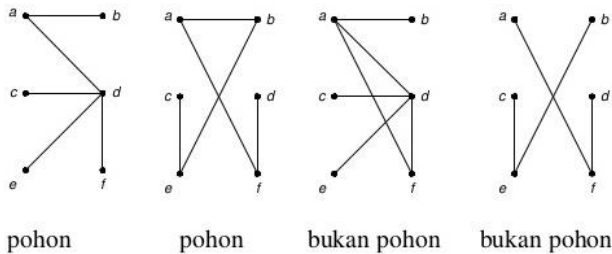
l. **Graf Berbobot (Weighted Graph)**

Graf berbobot adalah graf yang setiap sisinya diberi sebuah harga (bobot).

B. Teori Pohon

1. Definisi Pohon

Pohon (tree) merupakan graf tak-berarah terhubung yang tidak memiliki sirkuit. Ini berarti setiap pasang simpul dalam suatu pohon terhubung dengan lintasan tunggal dengan jumlah simpulnya  $n$  dan memiliki  $n - 1$  buah sisi. Penambahan satu sisi saja pada graf yang merupakan pohon akan membuat suatu sirkuit dan tidak dapat disebut sebagai pohon. Pohon dapat digunakan sebagai alat untuk memecahkan masalah dengan menggambarkan semua alternatif pemecahan.

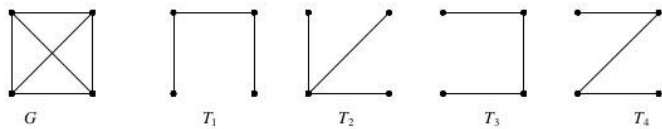


Gambar 2. Contoh pohon

Sumber : slide Pohon Matematika Diskrit Rinaldi Munir (2015)

### 2. Pohon Merentang (spanning tree)

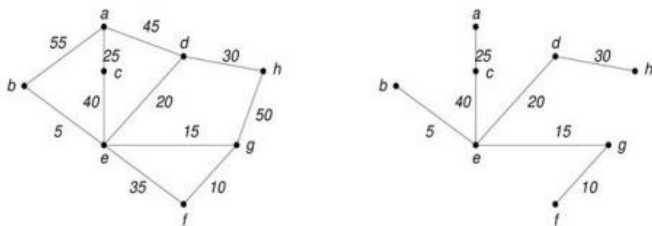
Pohon merentang adalah upagraf (subgraph) dari graf  $G$  yang merupakan pohon dan mencakup semua titik dari graf  $G$ . Pohon merentang diperoleh dengan cara memutuskan sirkuit di dalam graf  $G$  tersebut.



Gambar 3. Graf  $G$  dan empat buah pohon merentangnyanya ( $T_1, T_2, T_3, T_4$ )

Sumber : slide Pohon Matematika Diskrit Rinaldi Munir (2015)

Pohon merentang minimum adalah graf terhubung-berbobot yang merupakan pohon merentang dan memiliki bobot paling minimum.



Gambar 4. Graf  $G$  dan pohon merentang minimumnya

Sumber : slide Pohon Matematika Diskrit Rinaldi Munir (2015)

Dalam menentukan pohon merentang minimum, terdapat dua algoritma yang dapat digunakan, yaitu algoritma Prim dan algoritma Kruskal.

### 3. Algoritma Prim

Menentukan pohon merentang minimum dengan menggunakan algoritma Prim dapat dilakukan dengan langkah berikut:

- 1) Ambil sisi dari graf  $G$  yang berbobot minimum, masukkan ke dalam  $T$ .
- 2) Pilih sisi  $(u, v)$  yang mempunyai bobot minimum dan bersisian dengan simpul di  $T$ , tetapi  $(u, v)$  tidak membentuk sirkuit di  $T$ . Masukkan  $(u, v)$  ke dalam  $T$ .
- 3) Ulangi langkah 2 sebanyak  $n - 2$  kali.

```

Procedure Prim (input G : graf, output T : pohon)
{ Membentuk pohon merentang minimum T dari graf terhubung-berbobot G.
Masukan: graf-berbobot terhubung  $G = (V, E)$ , dengan  $|V| = n$ 
Keluaran: pohon rentang minimum  $T = (V, E')$ 
}

```

#### Deklarasi

$i, p, q, u, v$  : integer

#### Algoritma

Cari sisi  $(p, q)$  dari  $E$  yang berbobot terkecil

$T \leftarrow \{(p, q)\}$

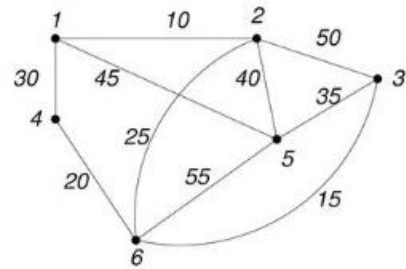
for  $i \leftarrow 1$  to  $n-2$  do

    Pilih sisi  $(u, v)$  dari  $E$  yang bobotnya terkecil namun bersisian dengan simpul di  $T$

$T \leftarrow T \cup \{(u, v)\}$

endfor

Contoh:



Langkah	Sisi	Bobot	Pohon rentang
1	(1, 2)	10	
2	(2, 6)	25	
3	(3, 6)	15	
4	(4, 6)	20	
5	(3, 5)	35	

Gambar 5. Contoh penerapan algoritma Prim

Sumber : slide Pohon Matematika Diskrit Rinaldi Munir (2015)

### 4. Algoritma Kruskal

Menentukan pohon merentang minimum dengan menggunakan algoritma Kruskal dapat dilakukan dengan

langkah berikut:

- 1) Urutkan sisi-sisi dari graf secara menaik berdasarkan bobotnya, yaitu dari bobot terkecil sampai bobot terbesar.
- 2)  $T$  masih kosong
- 3) Pilih sisi  $(u, v)$  dengan bobot minimum yang tidak membentuk sirkuit di  $T$ . Tambahkan  $(u, v)$  ke dalam  $T$ .
- 4) Ulangi langkah 3 sebanyak  $n-1$  kali.

```

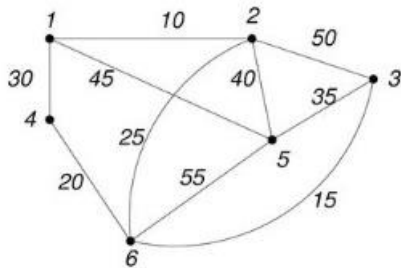
Procedure Kruskal (input G : graf, output T : pohon)
{ Membentuk pohon merentang minimum T dari graf terhubung-berbobot G.
Masukan: graf-berbobot terhubung G = (V, E), dengan |V| = n
Keluaran: pohon rentang minimum T = (V, E')
}

```

**Deklarasi**  
 $u, v$  : integer

**Algoritma**  
( Asumsi : sisi-sisi dari graf sudah diurut menaik berdasarkan bobotnya yaitu dari bobot terkecil ke bobot terbesar )  
 $T \leftarrow \{ \}$   
while jumlah sisi  $T < n-1$  do  
Pilih sisi  $(u,v)$  dari  $E$  yang bobotnya terkecil  
if  $(u,v)$  tidak membentuk siklus di  $T$  then  
 $T \leftarrow T \cup \{(u,v)\}$   
endif  
endfor

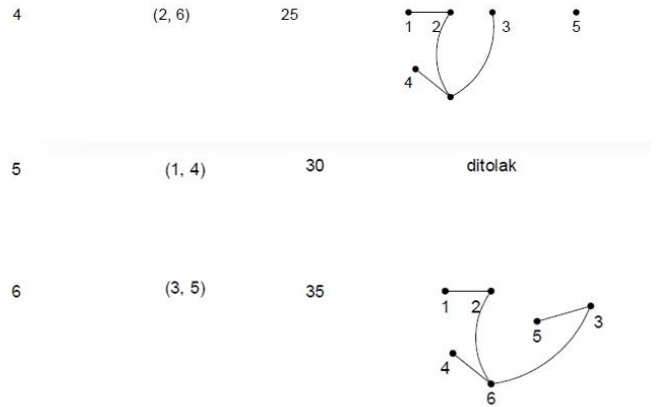
Contoh:



Sisi-sisi diurut menaik:

Sisi	(1,2)	(3,6)	(4,6)	(2,6)	(1,4)	(3,5)	(2,5)	(1,5)	(2,3)	(5,6)
Bobot	10	15	20	25	30	35	40	45	50	55

Langkah	Sisi	Bobot	Hutan merentang
0			
1	(1, 2)	10	
2	(3, 6)	15	
3	(4, 6)	20	



Gambar 6. Contoh penerapan algoritma Kruskal  
Sumber: slide Pohon Matematika Diskrit Rinaldi Munir (2015)

### C. Trem

Trem adalah kereta ringan yang memiliki rel khusus dan berbaur di antara jalan umum di dalam kota. Untuk menggerakkan trem, telah banyak sumber energi yang digunakan seperti tenaga kuda, tenaga uap, tenaga listrik dan tenaga matahari.

Di kota-kota besar, trem sering kali merupakan solusi yang baik untuk mengatasi kemacetan. Trem dapat mengangkut lebih banyak orang daripada transportasi publik yang lain di dalam kota seperti bus. Trem merupakan transportasi umum yang ramah lingkungan karena energi yang digunakannya. Menurut DEFRA (The Department for Environment, Food and Rural Affairs) London, UK, penggunaan mobil menghasilkan gas karbon dioksida sebanyak tiga kali lebih banyak daripada penggunaan trem.

Trem dulu pernah beroperasi di Batavia (Jakarta) pada tahun 1869 pada masa kolonialisme Belanda. Trem pada saat itu masih menggunakan tenaga kuda. Dalam pengoperasiannya, trem kuda masih menggunakan alat-alat sederhana. Kusir menggunakan terompet sebagai klakson. Tiap kali penumpang akan turun, penjual karcis membunyikan lonceng. Lalu kusir yang mendengar bunyi lonceng akan memutar alat menyerupai kompas yang berfungsi sebagai rem.



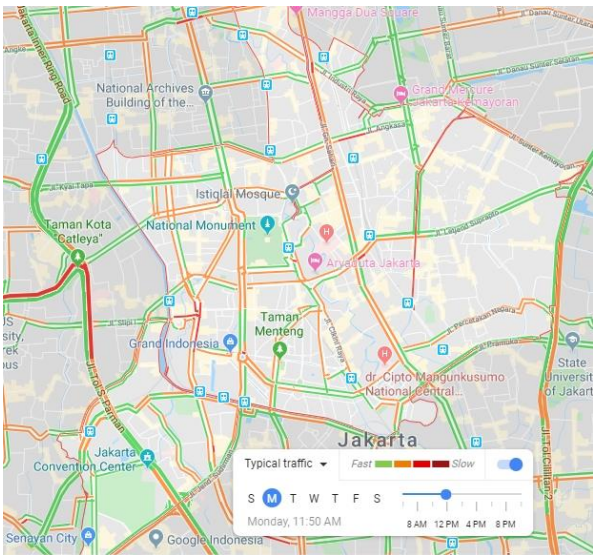
Gambar 7. Trem pada zaman dahulu  
Sumber:

<https://www.liputan6.com/citizen6/read/3187646/ini-jadinya-kalau-trem-masih-ada-di-jakarta>

Teknologi trem ini terus berkembang ke tenaga uap hingga akhirnya menggunakan tenaga listrik. Trem di Jakarta sudah tidak digunakan lagi setelah berakhirnya masa kolonialisme Belanda, padahal menurut penulis, trem ini merupakan salah satu solusi untuk mengatasi kemacetan yang sangat padat di Jakarta terutama Jakarta Pusat.

### III. PENERAPAN POHON MERENTANG MINIMUM DALAM PEMBUATAN JALUR TREM

Dalam menentukan stasiun-stasiun trem di Jakarta Pusat, penulis menggunakan *google maps* untuk mencari titik-titik rawan kemacetan dan menjadikan titik-titik tersebut sebagai stasiun-stasiun untuk pemberhentian trem. Diasumsikan semakin pendek jarak antar titik, maka akan semakin murah pula biaya pembuatan jalur trem.



Gambar 8. Peta Jakarta Pusat berdasarkan Google maps lewat satelit

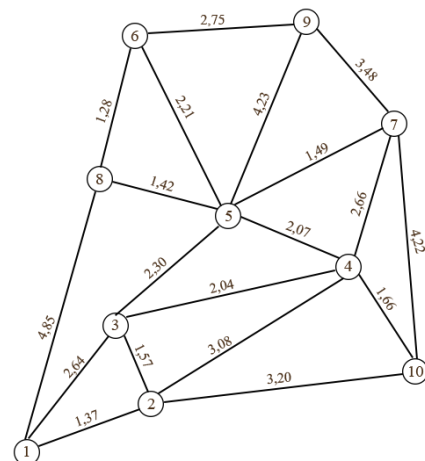
Berdasarkan status kemacetan yang ada pada *google maps* di atas, penulis mengasumsikan titik-titik berikut sebagai tempat yang rawan macet dan dapat dijadikan sebagai stasiun trem.

No.	Nama Tempat	Estimasi Jarak (km)
(1,2)	(Jakarta Convention Center, Citywalk Sudirman Jakarta)	1,37
(1,3)	(Jakarta Convention Center, Grand Indonesia)	2,64
(2,3)	(Citywalk Sudirman Jakarta, Grand Indonesia)	1,57
(2,4)	(Citywalk Sudirman Jakarta, Taman Ismail Marzuki)	3,08
(3,4)	(Grand Indonesia, Taman Ismail Marzuki)	2,04
(3,5)	(Grand Indonesia, Monumen Nasional)	2,30
(4,5)	(Taman Ismail Marzuki, Monumen Nasional)	2,07

	Monumen Nasional)	
(5,6)	(Monumen Nasional, Gajah Mada Plaza)	2,21
(6,7)	(Gajah Mada Plaza, Gedung Kesenian Jakarta)	1,88
(6,8)	(Gajah Mada Plaza, Taman Prasasti Museum)	1,28
(5,8)	(Monumen Nasional, Taman Prasasti Museum)	1,42
(7,9)	(Gedung Kesenian Jakarta, Mangga Dua Mall)	3,48
(6,9)	(Gajah Mada Plaza, Mangga Dua Mall)	2,75
(4,10)	(Taman Ismail Marzuki, Tugu Proklamasi)	1,66
(7,10)	(Gedung Kesenian Jakarta, Tugu Proklamasi)	4,22
(3,8)	(Grand Indonesia, Taman Prasasti Museum)	4,85
(1, 8)	(Jakarta Convention Center, Taman Prasasti Museum)	4,85
(2,10)	(Citywalk Sudirman Jakarta, Tugu Proklamasi)	3,20
(4,7)	(Taman Ismail Marzuki, Gedung Kesenian Jakarta)	2,66
(5,7)	(Monumen Nasional, Gedung Kesenian Jakarta)	1,49
(5,9)	(Monumen Nasional, Mangga Dua Mall)	4,23

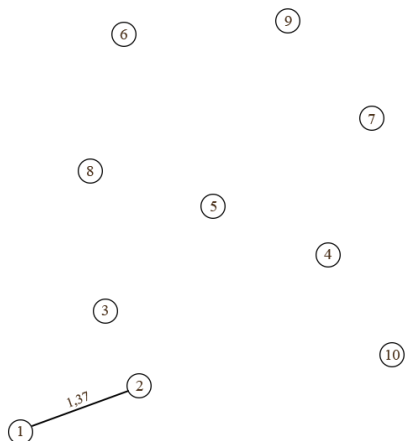
Tabel 2. Jarak antar titik di Jakarta Pusat

Jika titik-titik di atas disederhanakan dalam bentuk graf, maka akan terbentuk seperti berikut:



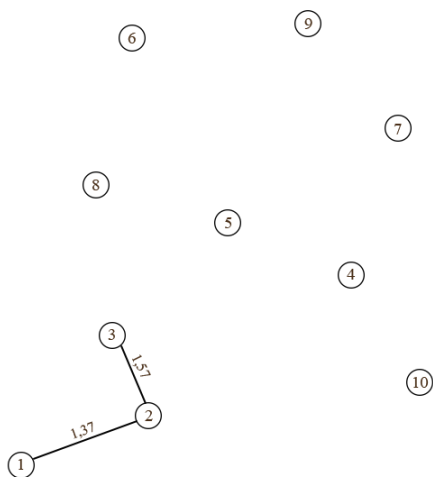
Gambar 9. Graf titik-titik kemacetan di Jakarta Pusat

Dengan menggunakan algoritma Prim, mula-mula pilih satu titik dan sambungkan dengan titik lain yang memiliki bobot terkecil.



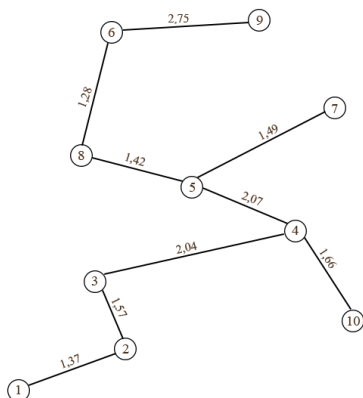
Gambar 10. Tahap pertama algoritma Prim

Selanjutnya lakukan langkah yang sama pada langkah pertama di titik 2.



Gambar 11. Tahap kedua algoritma Prim

Ulangi langkah ini sebanyak  $n-2$  kali maka akan didapatkan hasil jalur trem di Jakarta Pusat dengan pengeluaran biaya seminimum mungkin.



Gambar 12. Pohon merentang minimum dengan algoritma prim

Gambar di atas merupakan hasil perancangan jalur trem di Jakarta Pusat dengan menggunakan algoritma Prim yang dapat meminimalisir biaya pembangunannya. Dengan simpul-simpul yang bernomor dapat dijadikan sebagai stasiun pemberhentian trem.

Dengan konsep pohon merentang minimum dapat dipastikan setiap trem dapat mengakses seluruh stasiun pemberhentian yang merupakan simpul pada graf. Hasil dari pohon merentang minimum ini adalah jalur trem dengan biaya pembangunannya yang paling minimum bila diasumsikan biaya pembangunan berbanding lurus dengan panjangnya jarak antar simpul.

#### IV. KESIMPULAN

Dengan menentukan titik-titik yang merupakan rawan kemacetan karena lalu-lintas pada *google maps*, didapatkan graf dengan sisi-sisi pada simpulnya merepresentasikan jarak antar titik. Penentuan titik-titik ini hanyalah berdasarkan hasil penelitian penulis dengan meneliti kepadatan lalu-lintas di *google maps* jadi titik-titik tersebut dapat saja berubah.

Diterapkannya konsep pohon merentang minimum ini untuk pembuatan jalur trem diharapkan dapat meminimalisir biaya yang dikeluarkan pemerintah agar dapat mengatasi kemacetan dengan dioperasikannya trem.

#### V. PENUTUP

Pertama-tama penulis ingin mengungkapkan rasa syukur kepada Tuhan Yang Maha Esa karena penulis dapat menyelesaikan makalah Aplikasi Pohon Merentang Minimum dalam Pembuatan Jalur Trem di Jakarta Pusat ini. Penulis juga ingin mengucapkan rasa terima kasih yang sebesar-besarnya kepada bapak Dr. Ir. Rinaldi Munir, MT. Dan Ibu Fariska Zakhralativa Ruskanda selaku dosen pengajar mata kuliah Matematika Diskrit yang telah membimbing penulis dalam membuat makalah ini.

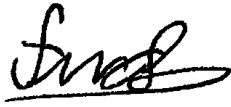
#### REFERENSI

- [1] Munir, Rinaldi, *Matematika Diskrit*, Bandung : Informatika, 2015
- [2] <https://www.google.com/maps>, diakses pada 4 Desember 2019.
- [3] <https://tirto.id/sejarah-trem-di-jakarta-dari-helaan-kuda-sampai-tenaga-listrik-dkha>, diakses pada 4 Desember 2019.
- [4] <https://www.liputan6.com/citizen6/read/3187646/ini-jadinya-kalau-trem-masih-ada-di-jakarta>, diakses pada 4 Desember 2019
- [5] <https://bettertransport.org.uk/bettertrams>, diakses pada tanggal 5 Desember 2019.

## 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, 5 Desember 2019

A handwritten signature in black ink, appearing to read 'Firas', with a horizontal line underneath it.

Muhammad Firas / 13518117