Aplikasi Pohon Merentang Minimum dalam Rute Jalur Kereta Api di Pulau Jawa

Darwin Prasetio (13512001)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13512001@std.stei.itb.ac.id

***Abstract*—Graf merupakan salah satu topik yang dibahas dalam Matematika Diskrit. Graf memodelkan sesuatu menjadi simpul dan menggambarkan hubungan-hubungan antar simpul dengan menggunakan sisi-sisi.Pohon merupakan salah satu terapan khusus dari teori graf.**

Salah satu terapan khusus dari pohon yang menarik adalah pohon merentang minimum (minimum spanning tree)..

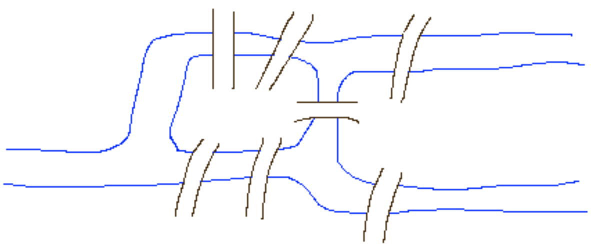
Makalah ini akan membahas tentang aplikasi pohon merentang minimum dalam pengiriman paket-paket JNE agar lebih efisien.

Kata Kunci – graf, pohon merentang minimum, algoritma prim, algoritma kruskal.

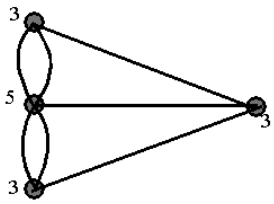
I. PENDAHULUAN

Graf banyak digunakan untuk memodelkan permasalahan-permasalahan yang kita hadapi sehari-hari. Banyak persoalan yang dapat kita modelkan menjadi simpul-simpul dan sisi-sisi yang menghubungkan simpul-simpul tersebut sebagai informasi hubungan antar simpul.

Salah satu contoh penerapan graf adalah penyelesaian permasalahan Jembatan Königsberg pada tahun 1736. Königsberg merupakan suatu kota di Prussia yang terletak di Sungai Pregel, yang merupakan tempat tinggal bagi bangsawan Prussia pada abad ke 16. Sungai Pregel yang mengalir melalui kota tersebut membuat sebuah pulau di tengah kota dan orang-orang membangun jembatan untuk menghubungkan pulau tersebut dengan bagian lain dari kota seperti pada gambar berikut.



Permasalahan yang muncul adalah apakah kita dapat mengelilingi kota Königsberg dan kembali ke posis awal dengan hanya melewati masing-masing jembatan sebanyak satu kali.
Seorang matematikawan Swiss, Leonhard Euler berhasil menyelesaikan masalah tersebut dengan memodelkannya menjadi graf berikut.



Dia menemukan bahwa mengelilingi kota Königsberg dan kembali ke posis awal dengan hanya melewati masing-masing jembatan sebanyak satu kali adalah mustahil karena sisi graf yang berjumlah ganjil lebih dari 2.

Aplikasi khusus dari graf adalah pohon. Salah satu contoh aplikasi pohon yang menarik adalah pohon merentang minimum. Salah satu masalah yang dapat diselesaikan dengan memanfaatkan pohon merentang minimum ini adalah konstruksi jalur kereta api untuk meminimumkan biaya perjalanan dan waktu perjalanan. Ini sangat penting bagi perkertaapian Indonesia karena belakangan ini pesawat terbang mulai mendominasi rute perjalanan, maka alternatif yang dapat diberikan PT KAI adalah dengan meminimumkan waktu perjalanan.

Makalah ini akan membahas tentang perutean jalur kereta api di pulau Jawa agar waktu perjalanan menjadi minimum dan biaya yang dikeluarkan menjadi sedikit dengan menggunakan algoritma Kruskal. Dengan menerapkan pohon merentang minimum maka rute jalur kereta api akan menjadi efektif dan efisien.

II. DASAR TEORI

2.1 Graf

Graf digunakan untuk memodelkan objek-objek diskrit dan hubungan-hubungan antar objek tersebut. Graf G merupakan himpunan (V,E) dengan :

V= himpunan tidak kosong dari simpul-simpul (vertices atau node)
= $\{v_1, v_2, \ldots, v_n\}$, dan
E = himpunan sisi (edges) yang menghubungkan Sepasang simpul
= $\{ e_1, e_2, .., e_n \}$

Notasi penulisan graf adalah G = (V,E).

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

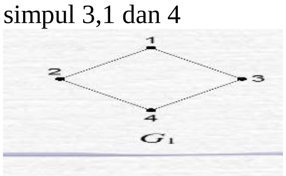
1. **Graf tak-berarah** (*undirected graph*) Graf yang sisinya tidak mempunyai orientasi arah disebut graf tak-berarah.

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

2.2 Terminologi Dasar

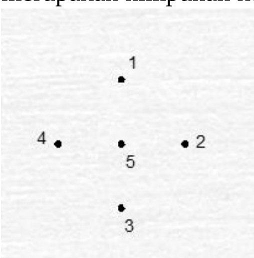
Beberapa terminologi dasar yang berhubungan dengan graf :

a) Ketetanggaan
Dua simpul dikatakan bertetangga jika dua simpul tersebut terhubung secara langsung. Pada graf G1, simpul 1 bertetangga dengan simpul 2,3, simpul 2 bertetangga dengan simpul 3,1 dan 4



b) Bersisian (Incidency)
Sebuah sisi e=(va dan vb) berarti e bersisian dengan simpul va atau e bersisian dengan simpul vb. Pada graf G1 di atas sisi (2,3) bersisian dengan simpul 2 dan simpul 3.

c) Graf kosong (null graph)
Graf kosong merupakan graf yang sisinya merupakan himpunan kosong (Nn).

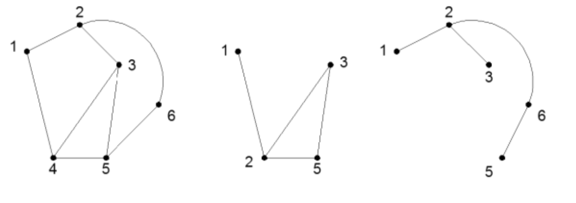


d) Derajat (degree)
Derajat suatu simpul adalah jumlah sisi yang bersisian dengan simpul tersebut. Contoh : pada graf G1 di atas simpul 2 berderajat 3.

e) Siklus (cycle) atau sirkuit (sirkuit)
Jika kita mengunjungi sejumlah simpul dalam sebuah graf G dan bisa kembali pada simpul yang sama disebut graf G memiliki sirkuit. Contoh : pada graf G1 di atas jika kita mengunjuni 1,2,3,1 maka itu merupakan sirkuit.

f) Terhubung (connected)
2 buah simpul v1 dan v2 dikatakan terhubung jika terdapat lintasan yang menghubungkan v1 dan v2. Contoh : pada graf G1 di atas simpul 1 dan simpul 4 terhubung.

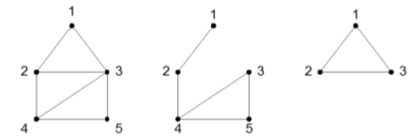
g) Upagraf (Subgraph) dan Komplemen Upagraf
Graf G1=(V1,E1) dikatakan merupakan upagraf dari graf G=(V,E) jika V1 $\subseteq$ V dan E1 $\subseteq$ E.
Komplemen dari upagraf G1 terhadap G adalah graf G2=(V2,E2) sedemikian sehingga E2=E-E1 dan V2 adalah himpunan simpul-simpul yang anggota-anggota E2 bersisian dengannya.



(a) Graf G_1 (b) Sebuah upagraf (c) komplemen dari upagraf (b)

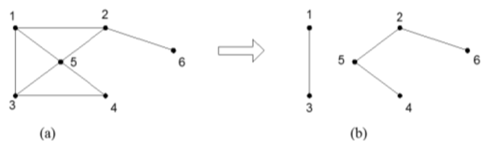
Komponen sebuah graf adalah jumlah maksimum upagraf terhubung dalam graf G.

h) Upagraf Rentang (Spanning Subgraph)
Upagraf G1=(V1,E1) dari G=(V,E) dikatakan upagraf rentang jika G1 mengandung semua simpul dari G.

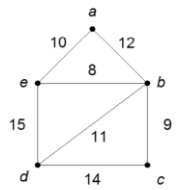


(a) graf G, (b) upagraf rentang dari G, (c) bukan upagraf rentang dari G

i) Cut-set
Cut-set dari graf G adalah himpunan sisi yang menyebabkan graf G menjadi tidak terhubung.

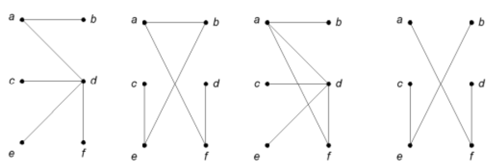


j) Graf berbobot (Weighted Graph)
Graf berbobot adalah graf yang setiap sisinya diberi sebuah harga (bobot).



2.3 Pohon

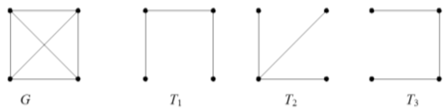
Pohon merupakan graf tak-berarah terhubung yang tidak mengandung sirkuit.



pohon pohon bukan pohon bukan pohon

2.4 Pohon merentang (spanning tree)

Pohon merentang dari graf terhubung merupakan upagraf yang berupa pohon. Dengan memutus sirkuit di dalam graf maka akan diperoleh pohon merentang.



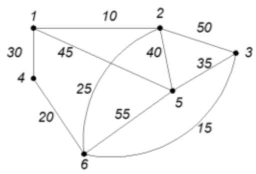
Setiap graf terhubung akan mempunyai paling sedikit 1 pohon merentang. Pohon merentang dari graf terhubung-berbobot yang mempunyai total bobot paling sedikit dinamakan pohon merentang minimum (minimum spanning tree). Terdapat 2 algoritma untuk membangun pohon merentang minimum yaitu :

a) Algoritma Prim

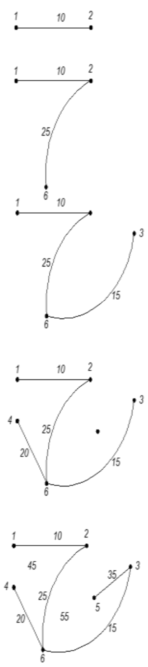
Langkah-langkah untuk membuat pohon merentang minimum dengan algoritma Prim adalah sebagai berikut :

1. Pilih sisi dari graf G yang mempunyai bobot minimum, masukkan ke dalam T.
2. pilih sisi(i,j) yang bersisian dengan simpul di T tetapi tidak membentuk sirkuit di T.
3. Ulangi langkah 2 sebanyak n-2 kali.

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	



b) Algoritma Kruskal

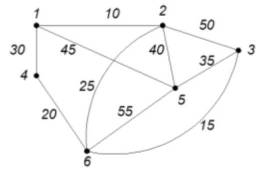
Langkah-langkah untuk membangun pohon merentang minimum dengan algoritma Kruskal adalah sebagai berikut:

0.Sisi-sisi di graf G sudah terurut menaik berdasarkan bobotnya.

1. T masih kosong
2. Pilih sisi (i,j) dengan bobot minimum yang tidak membentuk sirkuit di T. Tambahkan (i,j) ke T

3.Ulangi langkah 2 sebanyak n-1 kali.

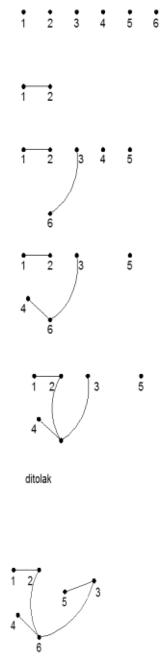
Contoh :



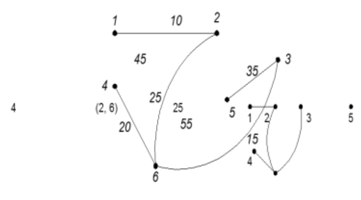
Sisi-sisi diurut menaik:

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	
4	(2, 6)	25	
5	(1, 4)	30	ditolak
6	(3, 5)	35	



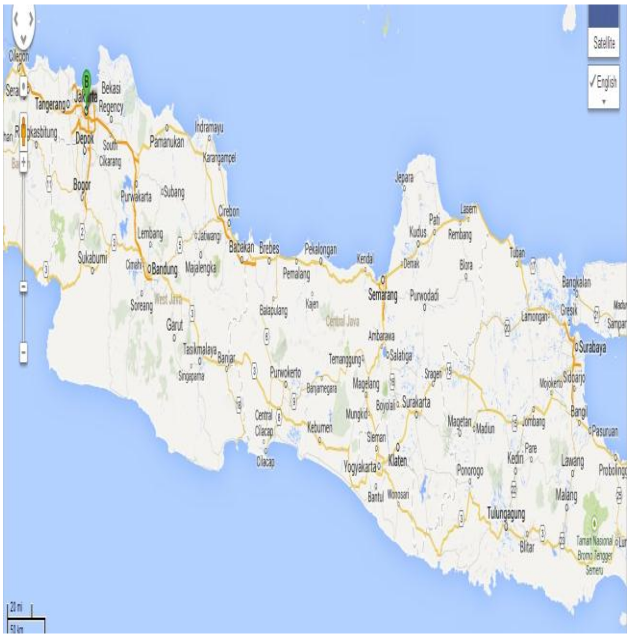
Pohon merentang minimum yang dihasilkan:



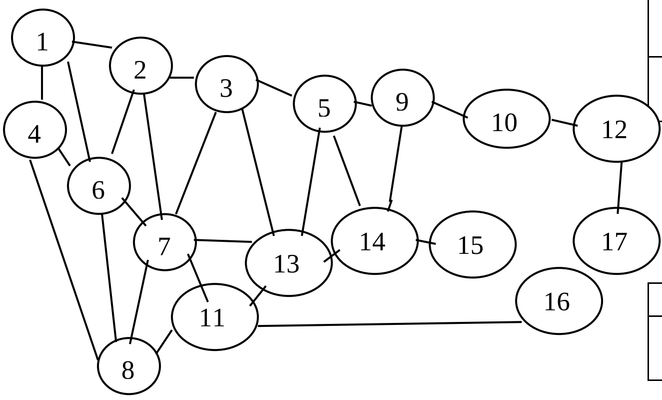
Bobot = 10 + 25 + 15 + 20 + 35 = 105

III. PEMBUATAN RUTE JALUR KERETA API

Stasiun-stasiun besar di pulau Jawa terdapat pada kota Jakarta, Bogor, Karawang, Bandung , Cirebon , Pekalongan, Semarang, Blora, Purwokerto, Cilacap, Purworejo, Yogyakarta, Klaten, Solo, Madiun, Nganjuk, Malang, dan Surabaya. Gambar berikut menunjukkan peta kota-kota yang telah disebutkan di atas.



Dengan peta tersebut, kota-kota tersebut dapat dimodelkan menjadi graf berbobot sebagai berikut.



1: Jakarta
2: Karawang
3: Cirebon
4: Bogor
5: Pekalongan
6: Bandung
7: Purwokerto
8: Cilacap
9: Semarang
10: Blora
11: Purworejo
12: Surabaya
13: Yogya
14: Klaten
15: Solo
16 : Madiun
17: Malang

Jarak (bobot) dalam kilometer:
1,2: 70,145
1,4: 52,5
1,6: 146,52
2,6: 89,939
2,3: 173,126
2,7: 316,482
3,5: 134,997
3,7: 147,105
3,13:309,889
4,6: 181,359
4,8: 432,097
5,9: 99,678
5,13: 177,724
8,11: 127,811
9,14: 112,731
5,14: 199.752
6,7: 244.119
10,12: 180.93
11,13: 65.96
6,8: 256.553
12,17: 92.802
13,14: 28.786
7,8: 53.895
7,11: 110.701
7,13: 164.74
11,16: 230.587
9,10:128.864
14,15: 35.62

Selanjutnya kita akan mengaplikasikan pohon merentang minimum yang telah dibahas pada bab sebelumnya agar pembuatan jalur kereta api menghabiskan biaya yang minimum dan jalur antar kota pendek tetapi semua kota tetap terhubung.

Dalam pembuatan pohon merentang minimum, terdapat 2 algoritma , yaitu algoritma Prim dan algoritma Kruskal. Dalam pembuatan jalan seperti ini, akan lebih efektif jika kita menggunakan algoritma Kruskal karena jumlah sisi yang sangat banyak. Algoritma Prim hanya efektif jika jumlah sisi dalam graf sedikit karena algoritma Prim memilih sisi berbobot minimum berdasarkan simpul, sedangkan algoritma Kruskal memilih berdasarkan sisi sehingga pada kasus ini akan lebih efektif apabila kita menggunakan algoritma Kruskal.

Langkah awal membuat pohon merentang minimum dengan algoritma Kruskal adalah mengurutkan semua sisi agar tersusun dari kecil ke besar.

13,14	14,15	1,4	7,8	11,13	1,2	2,6	12,17
28.786	35.62	52.5	53.895	65.96	70.145	89.939	92.802

5,9	7,11	9,14	8,11	9,10	3,5
99.678	110.701	112.731	127.811	128.864	134.997

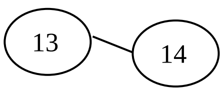
1,6	3,7	7,13	2,3	5,13	10,12
146.52	147.105	164.74	173.126	177.724	180.93

4,6	5,14	11,16	6,7	6,8	3,13
181.359	199.752	230.587	244.119	256.553	309.889

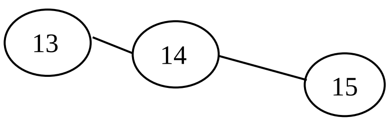
2,7	4,8
316,482	432.097

Setelah kita mengurutkan sisi berdasarkan bobotnya, selanjutnya kita membangun pohon minimum berdasarkan sisi-sisi yang telah kita urutkan. Pohon merentang yang kita buat kita namakan T. Pilih sisi dengan bobot minimum yang tidak membentuk sirkuit dan ulangi langkah ini sampai semua simpul pada graf termasuk dalam T.

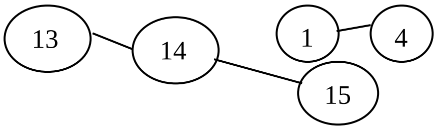
1)



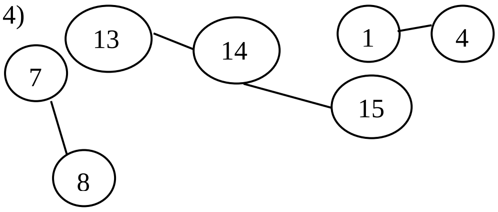
2)



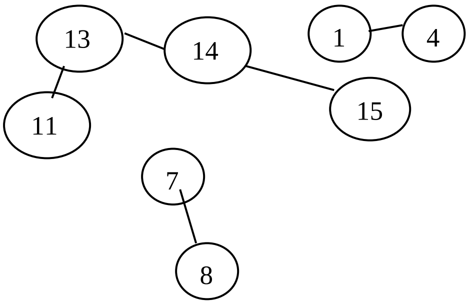
3)



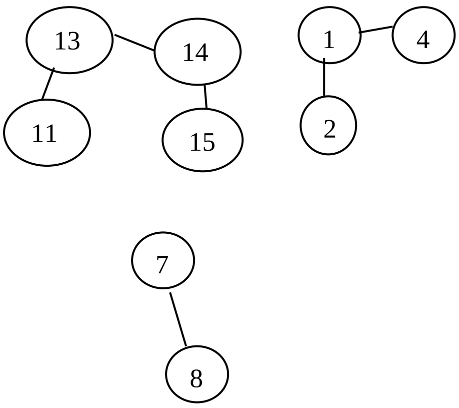
4)



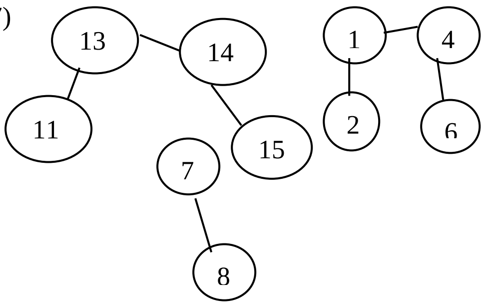
5)



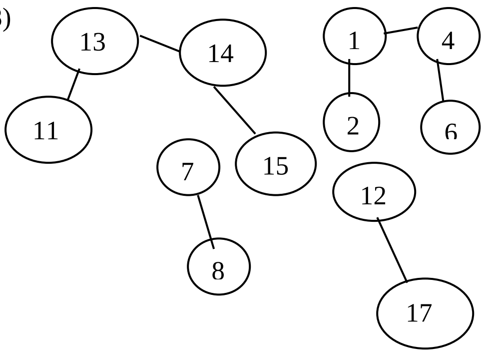
6)



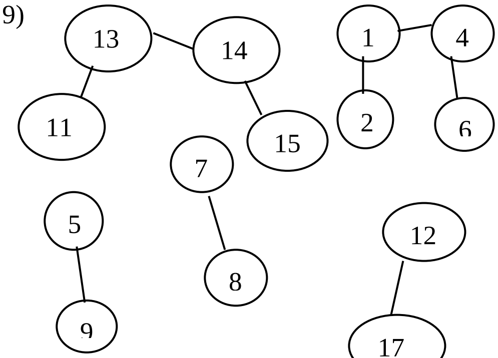
7)



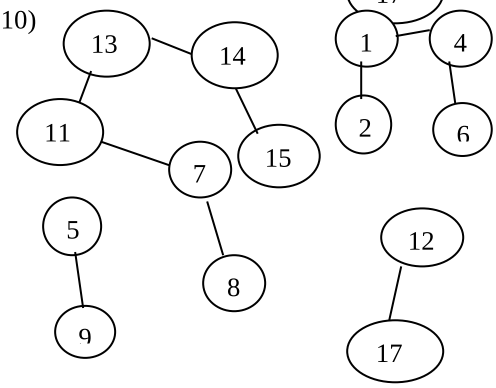
8)



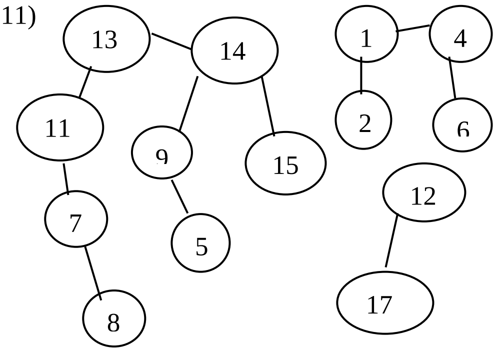
9)



10)

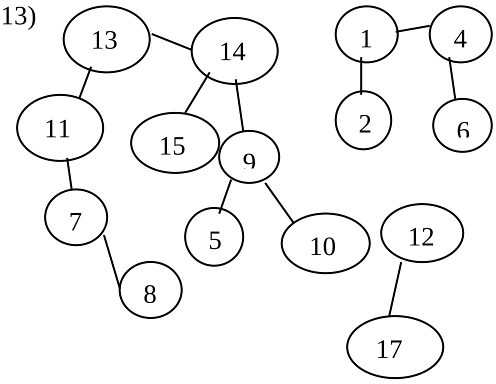


11)

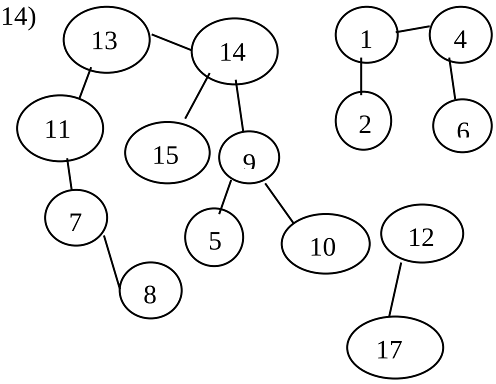


12) 8,11 ditolak karena membentuk sirkuit pada graf.

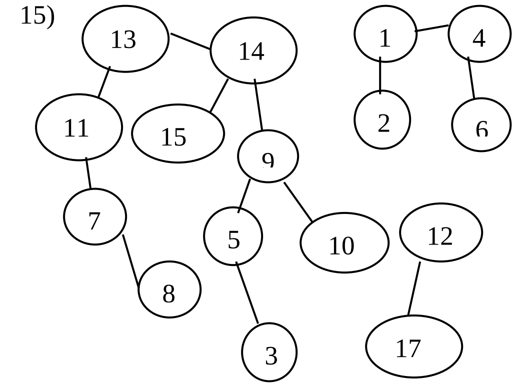
13)



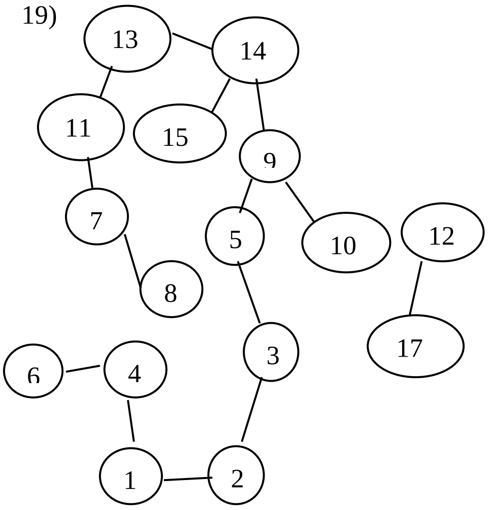
14)



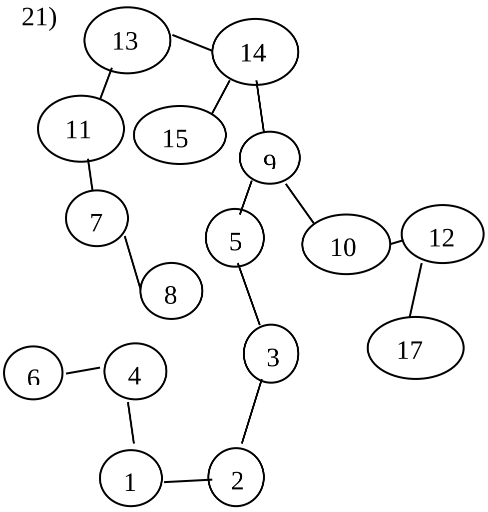
15)



16) 1,6 ditolak karena membentuk sirkuit
17) 3,7 ditolak karena membentuk sirkuit
18) 7,13 ditolak karena membentuk sirkuit
19)



20) 5,13 ditolak karena membentuk sirkuit
21)



Semua graf telah terhubung simpulnya, dengan demikian telah didapatkan pohon merentang minimum bagi pembuatan jalur kereta api agar lebih efisien dengan bobot graf = 1522.081

IV. Kesimpulan

Dalam penulisan makalah ini, saya mendapatkan beberapa kesimpulan :

1) Graf merupakan satu pemodelan yang sangat krusial dalam menyelesaikan permasalahan sehari-hari.
2) Algoritma Kruskal efektif digunakan jika jumlah sisinya banyak, algoritma Prim efektif digunakan jika jumlah simpul banyak dan jumlah sisi sedikit.
3) Aplikasi pohon merentang minimum sangat penting dalam konstruksi jalan karena akan membuat biaya yang dikeluarkan minimum dan jarak antar kota juga minimum.
4) Di bidang Informatika, graf berguna bagi perutean jaringan pengiriman pesan dan sejenisnya.

Referensi

[1] Munir,Rinaldi. Diktat Kuliah IF2120 Matematika Diskrit,Program Studi Teknik Informatika ITB.2008 : ITB.

[2] Graf dan permasalahan jembatan Konigsberg. http://www.jcu.edu/math/vignettes/bridges.htm, 14 Desember 2013.

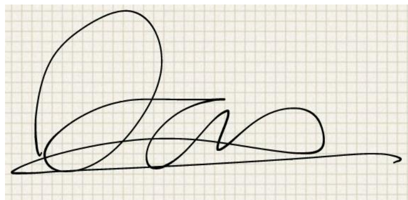
[3] Google Maps. Maps.google.com, 15 Desember 2013.

[4] http://jaraktempuh.com/hitung-jarak-dan-rute-antar-lokasi, 15 Desember 2013.

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, 16 Desember 2013



Darwin Prasetio
13512001