

**APLIKASI ALGORITMA PRIME  
DALAM MENENTUKAN POHON  
PEMBANKIT MINIMUM SUATU GRAF  
(Study Kasus)**



Oleh :

**Drs. Emut, M.Si**

**(Dosen Jurusan Matematika FMIPA UNY)**

**JURUSAN PENDIDIKAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS NEGERI YOGYAKARTA**

**2008**

# APLIKASI ALGORITMA PRIME DALAM MENENTUKAN POHON PEMBANGKIT MINIMUM SUATU GRAF

( Drs. Emut, M.Si)

## A. PENDAHULUAN

**Graf** merupakan suatu metode untuk mencari solusi dari permasalahan diskret yang di temui dalam kehidupan sehari-hari. Untuk mencari solusi ini, di dalam graf terdapat banyak konsep. Salah satu konsep yang sering dipakai adalah **konsep pohon** (*tree*). Konsep pohon adalah kosep yang paling penting dan populer, karena konsep pohon mampu mendukung aplikasi graf dalam berbagai bidang ilmu. Aplikasi ilmu yang menggunakan konsep pohon diantaranya: pembagunan jalan dan rel kereta api, pembuatan jaringan komputer, pencarian jalur untuk penjual keliling.

(<http://72.14.235.132/search?q=cache:kRtKrfphikEJ:www.ittelkom.ac.id>)

**Pohon adalah suatu graf terhubung yang tidak berarah dan tidak memuat siklus.** **Sirkulus** adalah lintasan yang berawal dan berakhir pada simpul yang sama. Jika  $G$  adalah graf berbobot, maka bobot dari pohon pembangkit  $T$  dari  $G$  didefinisikan sebagai jumlah bobot pada semua sisi di  $T$ . Pohon pembangkit yang berbeda memiliki bobot yang berbeda pula. Di antara pohon pembangkit di  $G$ , pohon yang memiliki bobot paling minimum dinamakan **pohon pembangkit minimum**. Pohon ini merupakan pohon pembangkit yang paling penting.

(<http://72.14.235.132/search?q=cache:ZDjQZOBM5PYJ:www.informatika.org>).

Terdapat beberapa algoritma untuk membangun pohon pembangkit minimum, salah satunya adalah menggunakan Algoritma Prim. **Algoritma Prim** adalah sebuah algoritma dalam teori graf untuk mencari pohon pembangkit minimum untuk sebuah graf terhubung berbobot. Dengan kata lain sebuah himpunan bagian dari cabang-cabang yang membentuk suatu pohon yang terdiri dari semua simpul, di mana bobot keseluruhan semua cabang dalam pohon adalah paling kecil. Bila graf tersebut tidak terhubung, maka graf itu hanya memiliki satu pohon pembangkit minimum untuk satu dari komponen yang terhubung.

Algoritma ini ditemukan oleh matematikawan [Vojtěch Jarník](#) pada tahun 1930. Kemudian secara terpisah ditemukan oleh ahli komputer [Robert C. Prim](#) pada tahun 1957 dan ditemukan kembali oleh [Dijkstra](#) pada tahun 1959. Oleh karena itu algoritma ini sering dinamai **algoritma DJP** atau **algoritma Jarnik**.

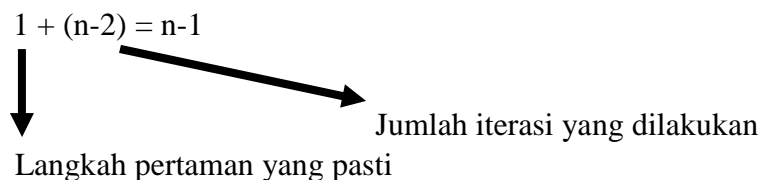
([http://id.wikipedia.org/wiki/Algoritma\\_Prim](http://id.wikipedia.org/wiki/Algoritma_Prim))

## B. ALGORITMA

Algoritma Prim digunakan untuk mencari pohon pembangkit minimum dari graf terhubung berbobot dengan cara mengambil sisi/ ruas garis yang memiliki bobot terkecil dari graf, dimana ruas garis tersebut bersisian dengan pohon terentang yang telah dibuat dan yang tidak membentuk siklus.

Untuk mencari pohon pembangkit minimum T dari graf G dengan Algoritma Prim dapat dilakukan dengan langkah-langkah sebagai berikut:

1. Ambil satu simpul sembarang (misalnya  $v_1 \in G$ ) dan masukkan simpul tersebut ke dalam graf T yang merupakan graf kosong.
2. Tambahkan satu rusuk yang terhubung dengan  $v_1$  dengan bobot yang paling minimum (misalnya  $e_1$ ) dan titik ujung lainnya ke T sehingga T terdiri dari sebuah rusuk  $e_1$  dan dua buah simpul yang merupakan titik-titik ujung dari rusuk  $e_1$  (salah satu titik ujung harus memuat simpul  $v_1$ ).
3. Pada langkah berikutnya, pilihlah sebuah rusuk dalam  $E(G)$  yang bukan  $E(T)$  dengan syarat yang harus dipenuhi sebagai berikut:
  - a. Rusuk tersebut menghubungkan salah satu simpul  $\in V(T)$ .
  - b. Rusuk tersebut mempunyai bobot minimal.
4. Ulangilah langkah tersebut (langkah 2-3) hingga diperoleh  $(n-1)$  rusuk dalam  $E(T)$  dengan  $n$  adalah banyaknya simpul dalam G. Dengan kata lain ulangi iterasi tersebut sebanyak  $(n-2)$ . ([Jong Jek Siang, 2002: 269](#)).



Adapun algoritma Prim yang digunakan dalam menentukan PPM (misal  $T$ ) dari graf terhubung berbobot ( $G$ ) secara formal dapat ditulis sebagai berikut:

1. Mula-mula  $T$  adalah graf kosong
2. Ambil sembarang simpul pada graf  $G$  misal  $v$  dengan  $v \in V(G)$ . Masukkan  $v$  dalam  $V(T)$ .
3.  $V(G) = V(G) - \{v\}$
4. Selanjutnya di dalam iterasinya:
  - a. Pilihlah garis  $e \in E(G)$  dan  $e \notin E(T)$  dengan syarat:
    - $e$  terhubung dengan satu simpul di  $T$  serta tidak membentuk siklus
    - $e$  mempunyai bobot terkecil dibandingkan dengan semua garis yang terhubung dengan simpul-simpul dalam  $T$ . Misal  $w$  adalah titik ujung  $e$  yang tidak berada dalam  $T$ .
  - b. Tambahkan garis dengan bobot  $e$  ke  $E(T)$  dan  $w$  ke  $V(T)$
5. Ulangi langkah ke-4 sebanyak  $(n - 2)$  kali.

### Bukti Algoritma Prim

Misalkan  $P$  adalah sebuah graf terhubung berbobot. Pada setiap iterasi algoritma Prim, suatu cabang harus ditemukan yang menghubungkan sebuah simpul di graf bagian ke sebuah simpul di luar graf bagian. Karena  $P$  terhubung, maka selalu ada jalur ke setiap simpul. Keluaran  $Y$  dari algoritma Prim adalah sebuah pohon, karena semua cabang dan simpul yang ditambahkan pada  $Y$  terhubung. Misalkan  $Y_I$  adalah pohon rentang minimum dari  $P$ . Bila  $Y_I = Y$  maka  $Y$  adalah pohon rentang minimum. Kalau tidak, misalkan  $e$  cabang pertama yang ditambahkan dalam konstruksi  $Y$  yang tidak berada di  $Y_I$ , dan  $V$  himpunan semua simpul yang terhubung oleh cabang-cabang yang ditambahkan sebelum  $e$ . Maka salah satu ujung dari  $e$  ada di dalam  $V$  dan ujung yang lain tidak. Karena  $Y_I$  adalah pohon rentang dari  $P$ , ada jalur dalam  $Y_I$  yang menghubungkan kedua ujung itu. Bila jalur ini ditelusuri, kita akan menemukan sebuah cabang  $f$  yang menghubungkan sebuah simpul di  $V$  ke satu simpul yang tidak di  $V$ . Pada iterasi ketika  $e$  ditambahkan ke  $Y$ ,  $f$  dapat juga ditambahkan jika dan hanya jika bobotnya lebih kecil daripada  $e$ . Karena  $f$  tidak ditambahkan, maka kesimpulannya

$$w(f) \geq w(e).$$

Misalkan  $Y_2$  adalah graf yang diperoleh dengan menghapus  $f$  dan menambahkan  $e$  dari  $Y_1$ . Dapat ditunjukkan bahwa  $Y_2$  terhubung, memiliki jumlah cabang yang sama dengan  $Y_1$ , dan bobotnya tidak lebih besar daripada  $Y_1$ , karena itu ia adalah pohon rentang minimum dari  $P$  dan ia mengandung  $e$  dan semua cabang-cabang yang ditambahkan sebelumnya selama konstruksi  $V$ . Ulangi langkah-langkah di atas dan kita akan mendapatkan sebuah pohon rentang minimum dari  $P$  yang identis dengan  $Y$ . Hal ini menunjukkan bahwa  $Y$  adalah pohon rentang minimum.

### C. DESKRIPSI

#### 1. Analisa Algoritma Prim

Algoritma Prim menggunakan strategi Greedy yaitu pada setiap langkah akan dipilih yang terbaik dan tidak mungkin mengulang langkah yang sebelumnya hingga akhir algoritma.

Langkah pertama pada Algoritma Prim adalah memilih sebarang simpul pada graf  $G$ , kemudian mencari sisi pada himpunan  $E$  yang menyatakan sisi-sisi pada graf  $G$  dengan bobot terkecil kemudian dimasukan pada himpunan  $T$ . Setelah itu akan dilakukan perulangan/iterasi sebanyak  $n-2$  untuk mencari sisi dengan bobot terkecil pada himpunan  $E$  yang bersisian dengan simpul yang telah dimasukan pada  $T$ . Hasil pencarian tersebut kemudian digabungkan atau ditambahkan pada himpunan  $T$ .

Kendala pada algoritma Prim	Solusi
<ul style="list-style-type: none"> <li>➤ tidak ada pengecekan secara eksplisit apakah sisi yang dipilih akan membentuk siklus atau tidak. Karena pada algoritma Prim sisi yang dimasukkan ke dalam <math>T</math> harus bersisian dengan sebuah simpul di <math>T</math></li> </ul>	<ul style="list-style-type: none"> <li>➤ dilihat apakah titik ujung/ simpul dari sisi tersebut sudah ada dalam <math>T</math> atau belum. Jika sudah ada maka tidak boleh memilih sisi tersebut karena pasti akan membentuk siklus</li> </ul>
<ul style="list-style-type: none"> <li>➤ tidak mampu menentukan sisi mana yang akan dipilih jika mempunyai bobot yang sama</li> </ul>	<ul style="list-style-type: none"> <li>➤ sisi yang dimasukkan harus terurut dari kecil ke besar.</li> </ul>

Selain itu dapat dilihat perbedaan prinsip antara algoritma Prim dan algoritma kruskal seperti tertera pada tabel berikut ini.

<b>Algoritma Prim</b>	<b>Algoritma Kruskal</b>
Sisi yang dimasukkan pada pohon T harus bersisian atau terhubung dengan sebuah simpul di T, sehingga akan meminimalkan waktu	Semua sisi boleh dimasukkan asal tidak membentuk siklus.
Pohon pembangkit yang dihasilkan berbeda dengan yang dihasilkan oleh algoritma kruskal	Pohon pembangkit yang dihasilkan berbeda dengan yang dihasilkan oleh algoritma Prim

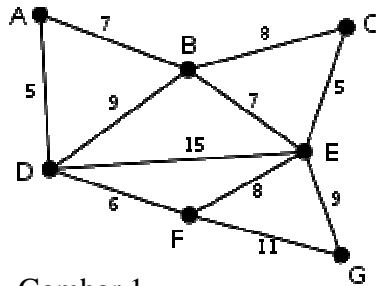
Algoritma Prim termasuk salah satu algoritma Greedy sedangkan pendekatan untuk mengecek siklus tersebut sama dengan kasus pemecahan masalah siklus Hamilton menggunakan algoritma Exhaustive Search yang notabene termasuk algoritma Brute Force, walaupun demikian algoritma Prim tetap masuk ke algoritma Greedy karena dalam pengecekan tersebut kita hanya menambahkan prasyarat agar terbentuk pohon pembangkit.

Algoritma Prim akan selalu berhasil menemukan pohon pembangkit minimum tetapi pohon pembangkit yang dihasilkan tidak selalu unik, maksudnya mungkin akan lebih dari 1 pohon yang dihasilkan dengan bobot yang sama hanya bentuknya saja yang berbeda.

#### **D. APLIKASI**

##### **Contoh 1:**

Gambar 1 adalah graf terhubung berbobot awal. Graf ini bukan pohon karena ada siklus. Nama yang lebih tepat untuk diagram ini adalah graf atau jaringan. Angka-angka dekat garis penghubung adalah bobotnya. Maka carilah pohon pembangkit minimum dari graf pada gambar 1



Gambar 1

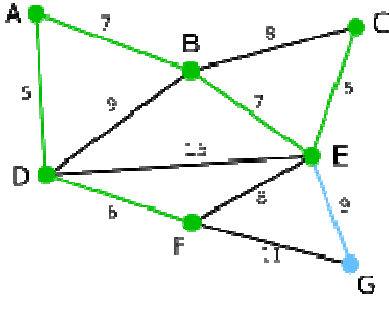
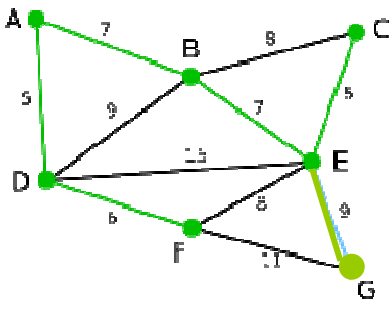
Penyelesaian:

*Petunjuk: simpul dan garis yang telah terpilih diberi warna hijau sebagai penanda telah menjadi anggota dari pohon pembangkit minimum, sehingga simpul dan ruas garis tidak terpilih kembali.*

	<p><b>Langkah pertama:</b> pilih salah satu simpul secara sebarang, misal simpul <b>D</b>, maka simpul tersebut beri warna hijau untuk menandakan bahwa simpul tersebut sudah terpilih menjadi anggota dari pohon.</p>
	<p><b>Langkah ke-2:</b> Setelah dipilih titik <b>D</b>, selanjutnya pilih garis yang terhubung pada simpul <b>D</b> dengan bobot terkecil. Garis-garis tersebut dengan bobotnya adalah <math>\{D,A\} = 5</math>, <math>\{D,B\} = 7</math>, <math>\{D,E\} = 15</math>, dan <math>\{D,F\} = 6</math>. Dari keempatnya, 5 adalah bobot yang terkecil, jadi garis yang dipilih adalah garis <b>DA</b>, kemudian diberi warna hijau untuk menandai bahwa garis tersebut sudah terpilih.</p>

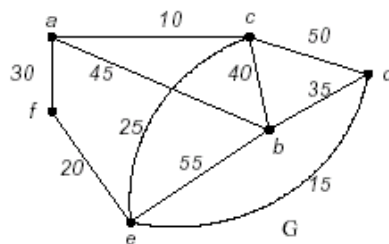
	<p>Pohon yang telah dihasilkan memiliki 1 ruas garis <b>DA</b> dengan simpul <b>A</b> dan <b>D</b>.</p> <p><b>Langkah ke-3:</b> Garis berikutnya yang dipilih adalah yang terhubung dengan <b>D</b> atau <b>A</b>.  <math>\{A,B\} = 7</math>, <math>\{D,B\} = 9</math>, <math>\{D,E\} = 15</math> dan <math>\{D,F\} = 6</math> adalah bobot yang terkecil, jadi kita pilih ruas garis <b>DF</b> kemudian beri warna hijau.</p>
	<p><b>Langkah ke-4:</b> Algoritma ini berlanjut seperti di atas. Garis <math>\{A,B\} = 7</math> merupakan salah satu dari banyak garis yang terhubung dengan simpul <b>A</b>, <b>D</b> atau <b>F</b> yang memiliki bobot terkecil. Maka ruas garis <b>AB</b> yang dipilih kemudian beri warna hijau.</p>
	<p>Di sini, ruas garis yang terhubung dengan simpul <b>A</b>, <b>D</b> atau <b>F</b> adalah <math>\{B,C\} = 8</math>, <math>\{B,D\} = 9</math>, <math>\{B,E\} = 7</math>, <math>\{D,E\} = 15</math>, <math>\{F,E\} = 8</math> dan <math>\{F,G\} = 11</math>.</p> <p>Ruas garis <b>BD</b> jangan dipilih, karena akan menghasilkan siklus.</p> <p>Ruas garis <b>BE</b> yang dipilih, karena memiliki bobot terkecil (7), kemudian beri warna hijau.</p>
	<p>Ruas garis yang terhubung dengan simpul, <b>B</b>, <b>E</b> atau <b>F</b> adalah <math>\{B,C\} = 8</math>, <math>\{E,C\} = 5</math>, <math>\{D,E\} = 15</math>, <math>\{E,F\} = 8</math>, <math>\{E,G\} = 9</math>, dan <math>\{F,G\} = 11</math></p> <p>Ruas garis <b>DE</b> dan <b>EF</b> jangan dipilih karena akan menghasilkan siklus, maka ruas garis yang dapat dipilih adalah <b>BC</b>, <b>EC</b>, <b>EG</b>, dan <b>FG</b>. Karena <b>EC</b> memiliki bobot terkecil yaitu 5</p>



	maka garis <b>EC</b> lah yang dipilih.
	Simpul <b>G</b> adalah satu-satunya yang tersisa. Maka dengan mudah cari ruas garis yang terhubung dengan simpul <b>G</b> dan tidak membentuk siklus dan memiliki bobot terkecil. Ruas garis $\{E,G\} = 9$ , dan $\{F,G\} = 11$ , maka kita pilih ruas garis <b>EG</b> .
	Sekarang semua simpul telah terhubung, dan pohon pembangkit minimum ditunjukkan dengan warna hijau, sehingga bobotnya = $6 + 5 + 7 + 7 + 5 + 9 = 39$ .

**Contoh 2:**

Studi kasus ini menggunakan gambar graf G pada gambar 2 di bawah. Kemudian akan kita lihat, apakah hasil dari algoritma Prim akan menghasilkan  $T_1$ ,  $T_2$ , ataukah T yang lain?

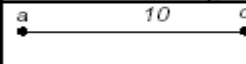
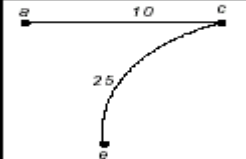
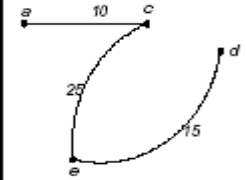
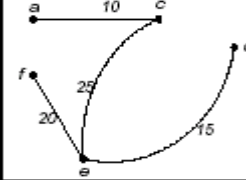
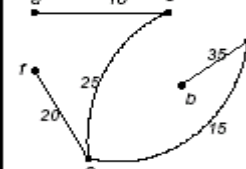


Gambar 2

**Penyelesaian:**

Langkah pertama pilih salah satu simpul, di sini kita pilih simpul **a**, iterasi selanjutnya dapat dilihat pada tabel berikut ini.

**Tabel pembentukan pohon pembangkit minimum dari graf G.**

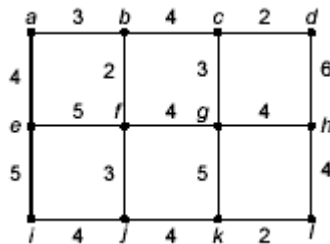
Langkah	Sisi	Bobot	Pohon Merentang
1	(a,e)	10	
2	(c,e)	25	
3	(d,e)	15	
4	(e,f)	20	
5	(b,d)	35	

Bobot pohon pembangkit minimum ini adalah  $10 + 25 + 15 + 20 + 35 = 105$

Pohon pembangkit yang dihasilkan hanya satu, karena graf di atas memiliki ruas garis dengan bobot yang berbeda-beda.

**Contoh 3:**

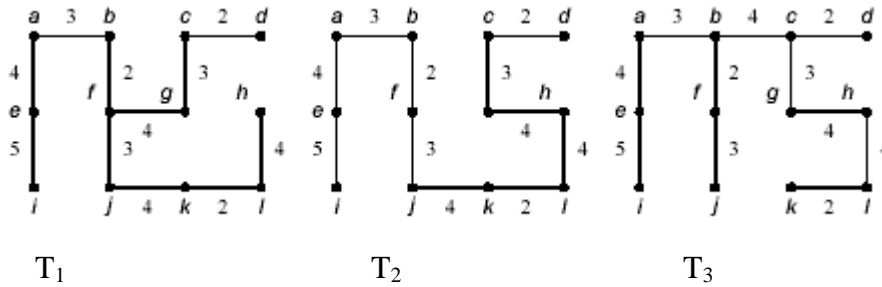
Gambarkan 3 buah pohon pembangkit minimum yang berbeda beserta bobotnya untuk graf pada Gambar 3 dengan menggunakan algoritma Prim.



Gambar 3

**Penyelesaian:**

Graf di atas memiliki beberapa sisi yang berbobot sama, maka ada kemungkinan pohon pembangkit minimumnya (T) lebih dari satu. Tiga buah di antaranya di adalah seperti di bawah ini:



$$T_1 = 5 + 4 + 3 + 2 + 3 + 4 + 2 + 4 + 4 + 3 + 2 = 36$$

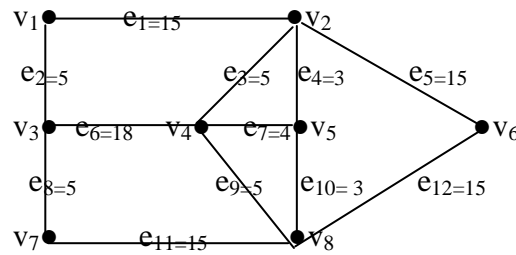
$$T_2 = 5 + 4 + 3 + 2 + 3 + 4 + 2 + 4 + 4 + 3 + 2 = 36$$

$$T_3 = 5 + 4 + 3 + 4 + 2 + 2 + 3 + 3 + 4 + 4 + 2 = 36$$

Dengan demikian dapat disimpulkan bahwa ketiga buah pohon pembangkit minimum di atas memiliki bentuk yang berbeda, namun jumlah bobot seluruh sisinya tetap sama, yaitu 36.

**Contoh 4:**

Gunakanlah algoritma Prim untuk mencari pohon rentang minimum pada graf berikut:


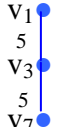
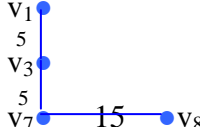
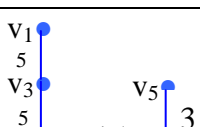
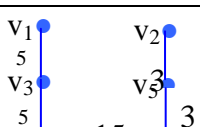
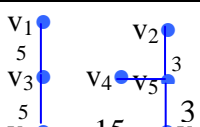
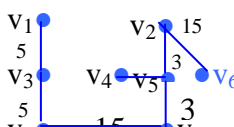


Gambar 4

**Penyelesaian:**

*Keterangan: garis yang dievaluasi adalah garis yang terhubung dengan simpul yang terdapat pada PPM dan yang tidak membentuk siklus*

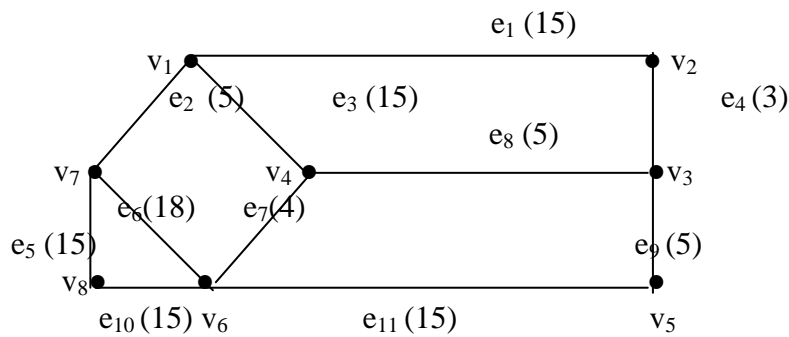
Iterasi	Garis yang di	Garis	Simpul	Keteranga	PPM
---------	---------------	-------	--------	-----------	-----

	evaluasi	yang terpilih	pada T	n	
Mula-mula	-	-	$v_3$	-	$v_3$
1	$e_{2=5}, e_{6=18}, e_{8=5}$	$e_{2=5}$	$\{v_1, v_3\}$	5 adalah bobot terkecil	
2	$e_{1=15}, e_{6=18}, e_{8=5}$	$e_{8=5}$	$\{v_1, v_3, v_7\}$	5 adalah bobot terkecil,	
3	$e_{1=15}, e_{6=18}, e_{11=15}$	$e_{11=15}$	$\{v_1, v_3, v_7, v_8\}$	15 adalah bobot terkecil, sehingga dapat memilih antara $e_1, e_{11}$	
4	$e_{1=15}, e_{6=18}, e_{9=5}, e_{10=3}, e_{12=15}$	$e_{10=3}$	$\{v_1, v_3, v_7, v_8, v_5\}$	3 adalah bobot terkecil	
5	$e_{1=15}, e_{4=3}, e_{6=18}, e_{7=4}, e_{12=15}$	$e_{4=3}$	$\{v_1, v_3, v_7, v_8, v_5, v_2\}$	3 adalah bobot terkecil	
6	$e_{1=15}, e_{3=5}, e_{5=15}, e_{6=18}, e_{7=4}, e_{12=15}$	$e_{7=4}$	$\{v_1, v_3, v_7, v_8, v_5, v_2, v_4\}$	4 adalah bobot terkecil	
7	$e_{5=15}, e_{12=15}$	$e_{5=15}$	$\{v_1, v_3, v_7, v_8, v_5, v_2, v_4, v_6\}$	15 adalah bobot terkecil	

8	Iterasi selesai	Banyak rusuk telah memenuhi $(n-1) = (8-1) = 7$	$T = 5 + 5 + 15 + 3 + 3 + 4 + 15 = 50$
---	-----------------	---	--

**Contoh 5:**

Gunakanlah algoritma Prim untuk mencari pohon rentang minimum pada graf berikut:

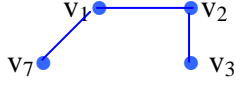
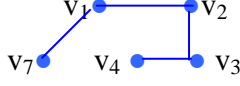
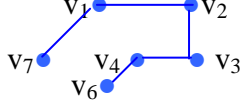
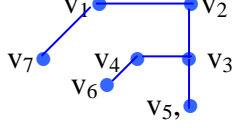
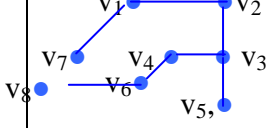


Gambar 5

**Penyelesaian:**

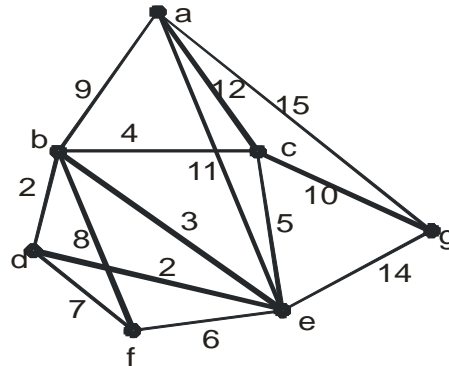
*Keterangan: garis yang dievaluasi adalah garis yang terhubung dengan simpul yang terdapat pada PPM dan yang tidak membentuk siklus*

Iterasi	Garis yang dievaluasi	Garis yang terpilih	Simpul pada T	Keterangan	PPM
Mula-mula	-	-	$v_1$	-	$v_1$
1	$e_2(5), e_1(15),$ dan $e_3(15)$	$e_2(5)$	$\{v_1, v_7\}$	5 adalah bobot terkecil	$v_1$ $v_7$
2	$e_1(15), e_3(15),$ $e_5(15), e_6(18)$	$e_1(15)$	$\{v_1, v_7, v_2\}$	15 adalah bobot terkecil,	$v_1$ $v_7$ $v_2$

				sehingga dapat memilih $e_5, e_3, e_1$	
3	$e_3 (15), e_4 (3), e_5 (15), e_6(18)$	$e_4 (3)$	$\{v_1, v_7, v_2, v_3\}$	3 adalah bobot terkecil	
4	$e_8 (5), e_5 (15), e_6(18), e_8 (5), e_9 (5)$	$e_8 (5)$	$\{v_1, v_7, v_2, v_3, v_4, v_6\}$	5 adalah bobot terkecil, sehingga dapat memilih antara $e_8$ dan $e_9$	
5	$e_7(4), e_5 (15), e_6(18), e_8 (5), e_9 (5)$	$e_7(4)$	$\{v_1, v_7, v_2, v_3, v_4, v_6\}$	4 adalah bobot terkecil	
6	$e_5 (15), e_{10} (15), e_9 (5), e_{11} (15)$	$e_9 (5)$	$\{v_1, v_7, v_2, v_3, v_4, v_6, v_5\}$	5 adalah bobot terkecil	
7	$e_5 (15), e_{10} (15)$	$e_{10} (15)$	$\{v_1, v_7, v_2, v_3, v_4, v_6, v_5, v_8\}$		
8	Iterasi selesai			Banyak rusuk telah memenuhi $(n-1) = (8-1) = 7$	$T = 5 + 15 + 3 + 5 + 4 + 5 + 15 = 52$

### Contoh 6

Tentukan suatu pohon perentang minimum dari graf berikut ini dengan menggunakan algoritma prim.



Gambar 6

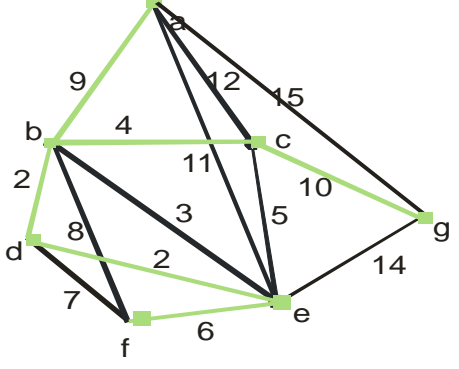
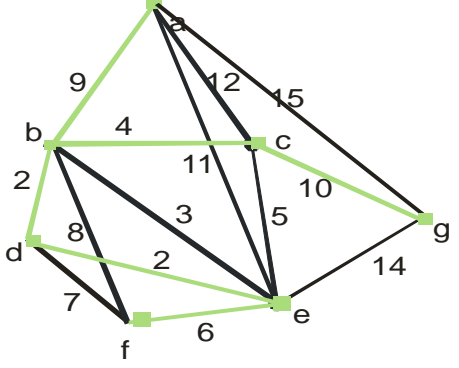
Penyelesaian :

*Petunjuk: simpul dan garis yang telah terpilih diberi warna hijau sebagai penanda telah menjadi anggota dari pohon perentang minimum, sehingga simpul dan ruas garis tidak terpilih kembali.*

	<p><b>Langkah pertama:</b> pilih salah satu simpul secara sebarang, misal simpul <b>a</b>, maka simpul tersebut beri warna hijau untuk menandakan bahwa simpul tersebut sudah terpilih menjadi anggota dari pohon.</p>
	<p><b>Langkah ke-2:</b> Setelah dipilih simpul <b>a</b>, selanjutnya pilih garis yang terhubung pada simpul <b>a</b> dengan bobot terkecil. Garis-garis tersebut dengan bobotnya adalah <math>\{a,b\} = 9</math>, <math>\{a,c\} = 12</math>, <math>\{a,e\} = 11</math>, dan <math>\{a,g\} = 15</math>. Dari keempatnya, <b>9</b> adalah bobot yang terkecil, jadi garis yang dipilih adalah garis <b>ab</b>, kemudian diberi warna hijau untuk menandai bahwa garis tersebut sudah terpilih.</p>

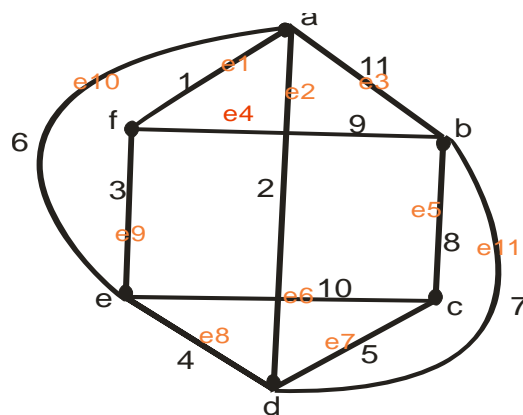
	<p>Pohon yang telah dihasilkan memiliki 1 ruas garis <b>ab</b> dengan simpul <b>a</b> dan <b>b</b>.</p> <p><b>Langkah ke-3:</b> Garis berikutnya yang dipilih adalah yang terhubung dengan <b>a</b> atau <b>b</b>.</p> <p><b>{b,d} = 2, {b,f} = 8, {D,E} = 15, {b,e}= 3, {b,c} = 4, {a,e} = 11, {a,c} = 12, dan {a,g} = 15</b></p> <p>2 adalah bobot yang terkecil, jadi kita pilih ruas garis <b>bd</b> kemudian beri warna hijau.</p>
	<p><b>Langkah ke-4:</b> Algoritma ini berlanjut seperti di atas. Garis <b>{d,e} = 2</b> merupakan salah satu dari banyak garis yang terhubung dengan simpul <b>a, b</b> atau <b>d</b> yang memiliki bobot terkecil. Maka ruas garis <b>de</b> yang dipilih kemudian beri warna hijau.</p>
	<p>Ruas garis yang terhubung dengan simpul, <b>a, b, d,</b> atau <b>e</b> adalah <b>{a,g} = 15, {a,c} = 12, {a,e} = 11, {b,c} = 4, {b,e} = 3, {b,f} = 8, {d,f} = 7, {e,f} = 6, {e,c} = 5, dan {e,g} = 14</b></p> <p>Ruas garis <b>be</b> dan <b>ae</b> jangan dipilih karena akan menghasilkan siklus, maka ruas garis yang dapat dipilih adalah <b>ag, ac, bc, bf, df, ef, ec,</b> dan <b>eg</b>. Karena <b>bc</b> memiliki bobot terkecil yaitu 4 maka garis <b>bc</b> lah yang dipilih.</p>
	<p>Ruas garis yang terhubung dengan simpul, <b>a, b, c, d,</b> atau <b>e</b> adalah <b>{a,g} = 15, {a,c} = 12, {a,e} = 11, {b,e} = 3, {b,f} = 8, {c,e} = 5, {c,g} = 10, {d,f} = 7, {e,f} = 6, dan {e,g} = 14</b></p> <p>Ruas garis <b>ac, ae, be</b> dan <b>ce</b> jangan dipilih karena akan menghasilkan siklus, maka ruas garis yang dapat dipilih adalah <b>ag, bf, cg, df, ef,</b> dan <b>eg</b>. Karena <b>ef</b></p>



	<p>memiliki bobot terkecil yaitu 4 maka garis <b>bc</b> lah yang dipilih.</p>
	<p>Simpul <b>g</b> adalah satu-satunya yang tersisa. Maka dengan mudah cari ruas garis yang terhubung dengan simpul <b>g</b> dan tidak membentuk siklus dan memiliki bobot terkecil. Ruas garis <b>{a,g} = 15</b>, <b>{c,g} = 10</b>, dan <b>{e,g} = 14</b> maka kita pilih ruas garis <b>cg</b>.</p>
	<p>Sekarang semua simpul telah terhubung, dan pohon perentang minimum ditunjukkan dengan warna hijau, sehingga bobotnya = <math>9 + 2 + 2 + 4 + 6 + 10 = 33</math>.</p>

### Contoh 7


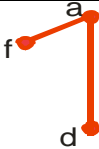

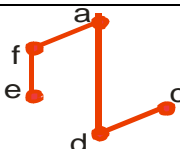
Tentukan suatu pohon perentang minimum dari gambar dibawah ini dengan menggunakan algoritma prim.

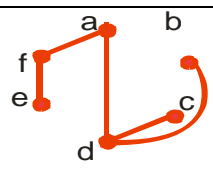


Gambar 7

**Penyelesaian:**

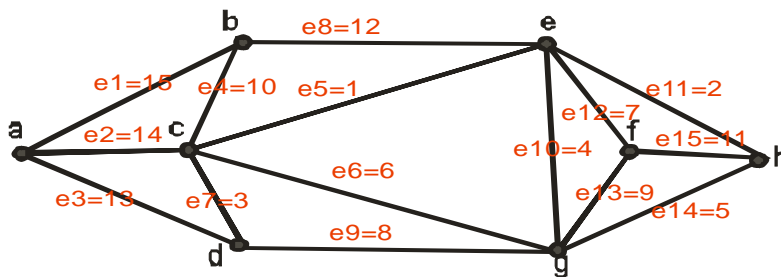
*Keterangan: garis yang dievaluasi adalah garis yang terhubung dengan simpul yang terdapat pada PPM dan yang tidak membentuk siklus*

Iterasi	Garis yang di evaluasi	Garis yang terpilih	Simpul pada T	Keterangan	PPM
Mula-mula	-	-	d	-	d ●
1	$e_8$ (4), $e_2$ (2), $e_7$ (5) dan $e_{11}$ (7)	$e_2$ (2)	{a, d}	2 adalah bobot terkecil	
2	$e_1$ (1), $e_3$ (11), $e_{10}$ (6), $e_8$ (4), $e_7$ (5), $e_{11}$ (7),	$e_1$ (1)	{a, d, f}	1 adalah bobot terkecil, sehingga dipilih $e_1$	
3	$e_3$ (11), $e_4$ (9), $e_{10}$ (6), $e_8$ (4), $e_7$ (5), $e_{11}$ (7), $e_9$ (3),	$e_9$ (3)	{a, d, e, f}	3 adalah bobot terkecil	
4	$e_3$ (11), $e_4$ (9), $e_{10}$ (6), $e_8$ (4), $e_7$ (5), $e_{11}$ (7), $e_6$ (10)	$e_7$ (5)	{a, c, d, e, f}	4 adalah bobot terkecil namun karena $e_8$ membentuk siklus maka dipilih $e_7$ dengan	

				bobot 5.	
5	$e_3(11),$ $e_4(9),$ $e_{10}(6),$ $e_8(4),$ $e_5(8),$ $e_6(10),$ $e_{11}(7),$	$e_{11}(7)$	{a, b, c, d, e, f}	$e_3$ dan $e_{10}$ tidak boleh diambil karena akan membentuk siklus, yang diambil adalah $e_{11}$ dengan bobot 7.	
6	Iterasi selesai			Banyak rusuk telah memenuhi $(n-1) = (6-1) = 5$	$T = 2+1+3+5+7=18$

### Contoh 8


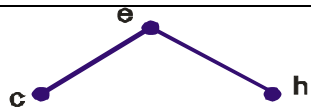
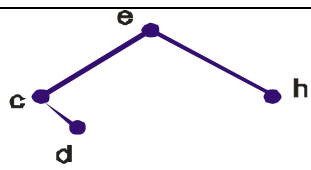
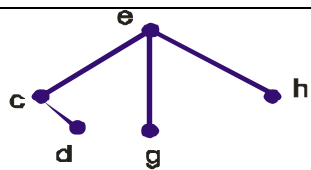
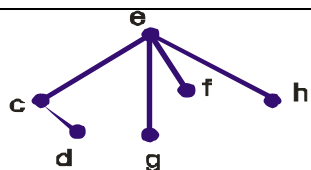
Tentukan suatu pohon perentang minimum dari gambar dibawah ini dengan menggunakan algoritma prim.

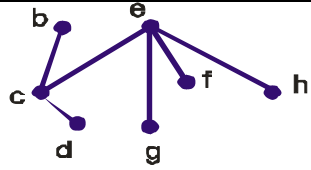
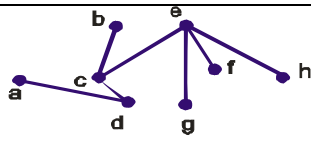


Gambar 8

### Penyelesaian:

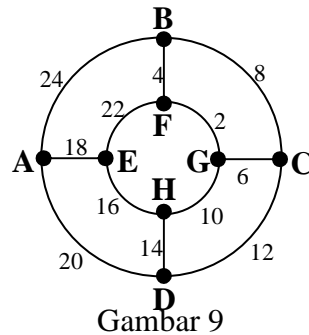
*Keterangan: garis yang dievaluasi adalah garis yang terhubung dengan simpul yang terdapat pada PPM dan yang tidak membentuk siklus*

Iterasi	Garis yang di evaluasi	Garis yang terpilih	Simpul pada T	Keterangan	PPM
Mula-mula	-	-	h	-	h●
1	$e_{11}=2,$ $e_{14}=5,$ $e_{15}=11$	$e_{11}=2$	{e,h}	2 adalah bobot terkecil	
2	$e_5=1, e_8=12,$ $e_{10}=4, e_{12}=7,$ $e_{14}=5,$ $e_{15}=11$	$e_5 = 1$	{c,e,h}	1 adalah bobot terkecil, sehingga dipilih $e_5$	
3	$e_2=14, e_4=10,$ $e_6=6,$ $e_7=3, e_8=12,$ $e_{10}=4, e_{12}=7,$ $e_{14}=5,$ $e_{15}=11$	$e_7=3$	{c,d,e,h}	3 adalah bobot terkecil	
4	$e_2=14,$ $e_3=13, e_4=10,$ $e_6=6, e_8=12,$ $e_9=8, e_{10}=4$ $e_{12}=7,$ $e_{14}=5,$ $e_{15}=11$	$e_{10}=4$	{c,d,e,g,h}	4 adalah bobot terkecil sehingga dipilih $e_{10}$ .	
5	$e_2=14,$ $e_3=13, e_4=10,$ $e_6=6, e_8=12,$ $e_9=8, e_{12}=7,$ $e_{13}=9,$ $e_{14}=5,$ $e_{15}=11$	$e_{12}=7$	{ c, d, e, f, g, h }	$e_6, e_{14}$ tidak boleh diambil karena akan membentuk siklus, yang diambil	

				adalah $e_{12}$ dengan bobot 7.	
6	$e_2=14,$ $e_3=13, e_4=10,$ $e_6=6, e_8=12,$ $e_9=8, e_{13}=9,$ $e_{14}=5,$ $e_{15}=11$	$e_4=10$	{ b, c, d, e, f, g, h }	$e_9, e_{13}$ tidak boleh diambil karena akan membentuk siklus, yang diambil adalah $e_4$ dengan bobot 10.	
7	$e_1=15, e_2=14,$ $e_3=13,$ $e_6=6, e_8=12,$ $e_9=8, e_{13}=9,$ $e_{14}=5,$ $e_{15}=11$	$e_3=13$	{a,b, c, d, e, f, g, h }	$e_9, e_{13}$ tidak boleh diambil karena akan membentuk siklus, yang diambil adalah $e_3$ dengan bobot 13.	
	Iterasi selesai			Banyak rusuk telah memenuhi $(n-1)= (8-1)=7$	$T=$ $2+1+3+4+7+10+13=40$

**Contoh 9:**

Tentukanlah pohon pembangkit minimum untuk graf berikut ini (gunakan algoritma prim).

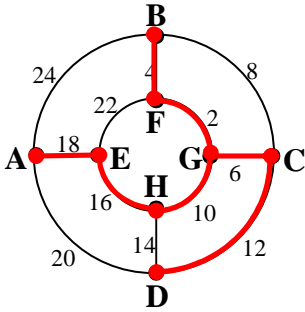
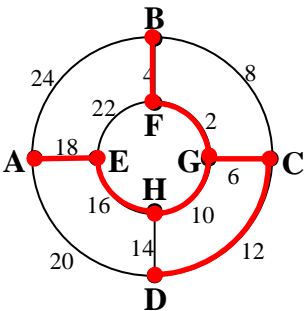


Penyelesaian :

*Petunjuk: simpul dan garis yang telah terpilih diberi warna merah sebagai penanda telah menjadi anggota dari pohon perentang minimum, sehingga simpul dan ruas garis tidak terpilih kembali.*

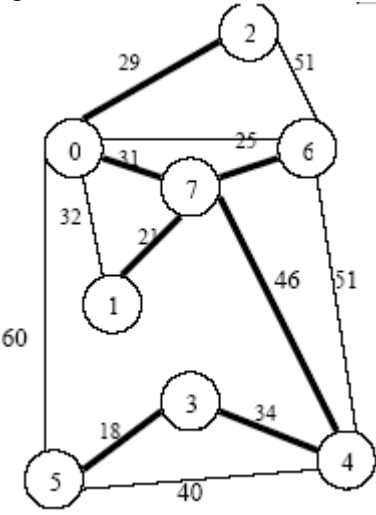
	<p><b>Langkah pertama:</b> pilih salah satu simpul secara sebarang, misal simpul <b>A</b>, maka simpul tersebut beri warna hijau untuk menandakan bahwa simpul tersebut sudah terpilih menjadi anggota dari pohon.</p>
	<p><b>Langkah ke-2:</b> Setelah dipilih simpul <b>A</b>, selanjutnya pilih garis yang terhubung pada simpul <b>A</b> dengan bobot terkecil. Garis-garis tersebut dengan bobotnya adalah <math>\{A,B\} = 24</math>, <math>\{A,E\} = 18</math>, dan <math>\{A,D\} = 20</math>. Dari ketiganya, <b>18</b> adalah bobot yang terkecil, jadi garis yang dipilih adalah garis <b>AE</b>, kemudian diberi warna merah untuk menandai bahwa garis tersebut sudah terpilih.</p>
	<p>Pohon yang telah dihasilkan memiliki 1 ruas garis <b>AE</b> dengan simpul <b>A</b> dan <b>E</b>.</p> <p><b>Langkah ke-3:</b> Garis berikutnya yang dipilih adalah</p>

	<p>yang terhubung dengan <b>A</b> atau <b>E</b>.</p> <p><math>\{A,B\} = 24</math>, <math>\{A,D\} = 20</math>, <math>\{E,F\} = 22</math>, <math>\{E,H\} = 16</math>,  <b>16</b> adalah bobot yang terkecil, jadi kita pilih ruas garis <b>EH</b> kemudian beri warna merah.</p>
	<p><b>Langkah ke-4:</b> Algoritma ini berlanjut seperti di atas. Garis <math>\{H,G\} = 10</math> merupakan salah satu dari banyak garis yang terhubung dengan simpul <b>A</b>, <b>E</b> atau <b>H</b> yang memiliki bobot terkecil. Maka ruas garis <b>HG</b> yang dipilih kemudian beri warna merah.</p>
	<p><b>Langkah ke-5:</b> Algoritma ini berlanjut seperti di atas. Garis <math>\{G,F\} = 2</math> merupakan salah satu dari banyak garis yang terhubung dengan simpul <b>A</b>, <b>E</b>, <b>H</b> atau <b>G</b> yang memiliki bobot terkecil. Maka ruas garis <b>GF</b> yang dipilih kemudian beri warna merah.</p>
	<p><b>Langkah ke-6:</b> Algoritma ini berlanjut seperti di atas. Garis <math>\{F,B\} = 4</math> merupakan salah satu dari banyak garis yang terhubung dengan simpul <b>A</b>, <b>E</b>, <b>H</b>, <b>G</b> atau <b>F</b> yang memiliki bobot terkecil. Maka ruas garis <b>FB</b> yang dipilih kemudian beri warna merah.</p>
	<p><b>Langkah ke-7:</b> Ruas garis yang terhubung dengan simpul <b>A</b>, <b>E</b>, <b>H</b>, <b>G</b>, <b>F</b> atau <b>B</b> adalah <math>\{A,B\} = 24</math>, <math>\{A,D\} = 20</math>, <math>\{E,F\} = 22</math>, <math>\{H,D\} = 14</math>, <math>\{G,C\} = 6</math>, dan <math>\{B,C\} = 8</math>.</p> <p>Ruas garis <b>AB</b> dan <b>EF</b> jangan dipilih karena akan menghasilkan siklus, maka ruas garis yang dapat dipilih adalah <b>AD</b>, <b>HD</b>, <b>GC</b> dan <b>BC</b>. Karena <b>GC</b></p>

	<p>memiliki bobot terkecil yaitu 6 maka garis <b>GC</b> lah yang dipilih dan diberi warna merah.</p>
	<p><b>Langkah ke-8:</b> Simpul <b>D</b> adalah satu-satunya yang tersisa. Maka dengan mudah cari ruas garis yang terhubung dengan simpul <b>D</b> dan tidak membentuk siklus dan memiliki bobot terkecil. Ruas garis <math>\{C,D\} = 12</math>, dan <math>\{H,D\} = 14</math> maka kita pilih ruas garis <b>CD</b>.</p>
	<p>Sekarang semua simpul telah terhubung, dan pohon perentang minimum ditunjukkan dengan warna merah, sehingga bobotnya = <math>18 + 16 + 10 + 2 + 4 + 6 + 12 = 68</math>.</p>

**Contoh 10**

Gambar dibawah ini merupakan gambar sebuah jaringan kabel telepon. Tentukan pohon pembangkit mimum(T) dari jaringan di bawah ini.



Gambar 11



Langkah	E	V-e	bobot
1	{0}	{1,2,3,4,5,6,7}	29
2	{0,2}	{1,3,4,5,6,7}	31
3	{0,2,7}	{1,3,4,5,6}	25
4	{0,2,6,7}	{1,3,4,5}	21
5	{0,1,2,6,7}	{3,4,5}	46
6	{0,1,2,4,6,7}	{3,5}	34
7	{0,1,2,3,4,6,7}	{5}	18
8	{0,1,2,3,4,5,6,7}	{}	204

## E. KESIMPULAN

Dari penjelasan dan contoh soal di atas dapat disimpulkan bahwa algoritma Prim tidak menentukan sisi mana yang dipilih jika terdapat lebih dari satu buah sisi yang berbobot sama. Satu cara untuk mengatasi hal ini adalah dengan mengurutkan sisi-sisi itu berdasarkan bobotnya dari kecil ke besar. Lagi pula, pohon pembangkit minimum yang dihasilkan tidak selalu unik. Graf sederhana terhubung dan berbobot dapat memiliki lebih dari satu buah pohon pembangkit minimum yang berbeda tetapi jumlah bobot minimumnya tetap sama. Selain itu, algoritma Prim lebih berorientasi pada pencarian simpul-simpul yang dihubungkan oleh sisi dengan bobot minimum.

## F. DAFTAR PUSTAKA

- Hernawati, Reni. *Algoritma Prim dengan Strategi Greedy untuk Membangun Pohon Pembangkit Minimum*. Available. from: <http://www.stttelkom.ac.id/staf/FAY/kuliah/DAA/20052/Tugas1/pdfs/26DAA%2020052%20113030010%20Algoritma%20Prim%20Dengan%20Strategi%20Greedy%20untuk%20Membangun%20Minimum%20Spanning%20Tree.pdf>. Accessed November 26, 2008.
- Siang, Jek Jong. 2002. *Matematika Diskrit dan Aplikasinya Pada Ilmu Komputer*. Yogyakarta: Andi.
- Wikipedia bahasa Indonesia, ensiklopedia bebas. 2008. *Algoritma Prim*. Available. From: [http://id.wikipedia.org/wiki/Algoritma\\_Prim](http://id.wikipedia.org/wiki/Algoritma_Prim). Accessed November 26, 2008.
- Anonim. 1999. *Masalah Pencarian Pohon Rentangan Minimum*. Available. From: <http://www2.toki.or.id/sda/archive/1998/handout/handout23.html>. Accessed November 26, 2008.
- Anonim. *Metode Greedy*. Available. From: [jkw1.files.wordpress.com/2008/06/pertemuan-algoritma-12-s-d-13.doc](http://jkw1.files.wordpress.com/2008/06/pertemuan-algoritma-12-s-d-13.doc). Accessed November 26, 2008.
- Anonim. *Pohon*. Available. From: [www.stttelkom.ac.id/staf/ZKA/ASD/download.php?file=Pohon.pdf](http://www.stttelkom.ac.id/staf/ZKA/ASD/download.php?file=Pohon.pdf). Accessed November 26, 2008.