Pohon (Bag. 1)

(Update 2023)

Bahan Kuliah

IF2120 Matematika Diskrit

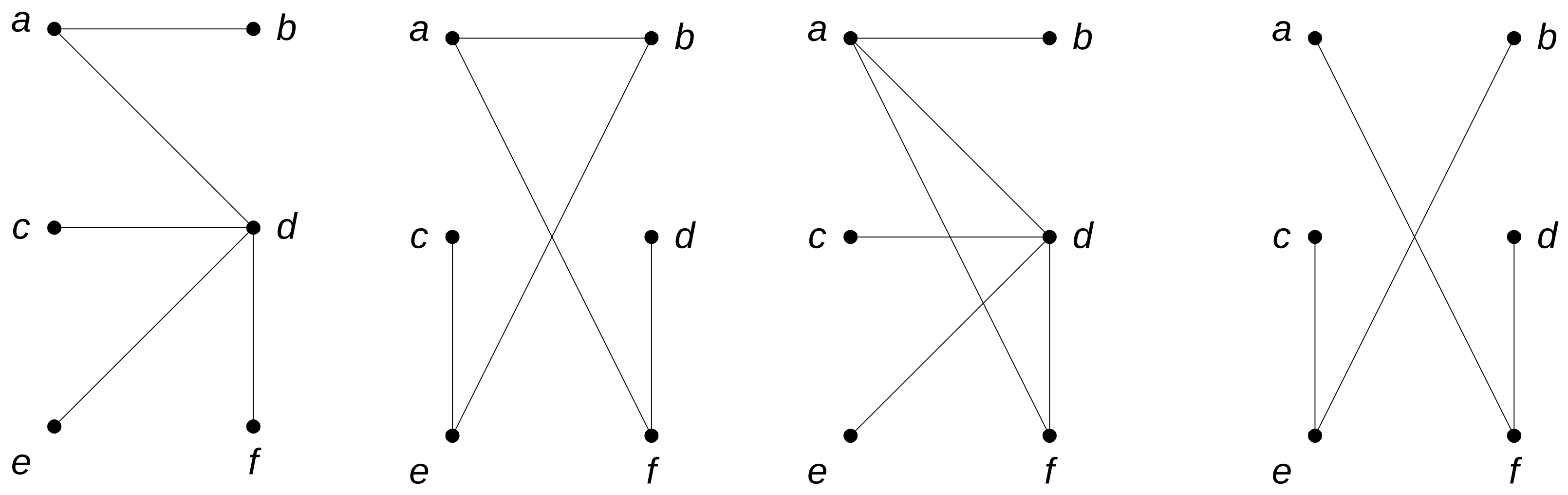
Oleh: Rinaldi Munir

Program Studi Teknik Informatika
STEI- ITB

Definisi Pohon

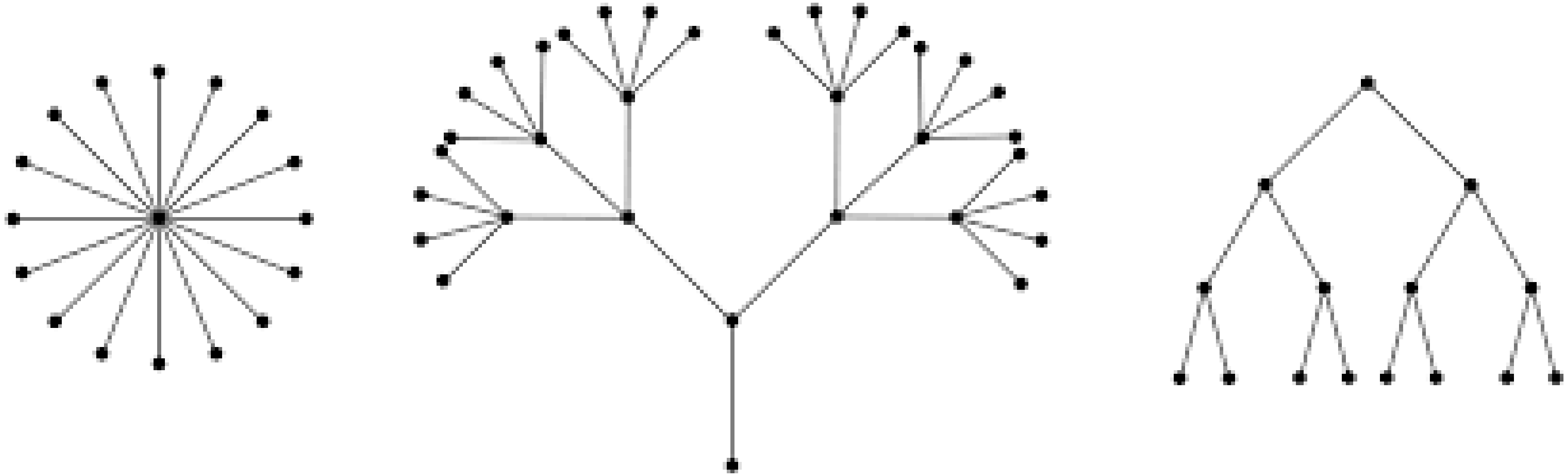
- **Pohon** adalah graf tak-berarah terhubung yang tidak mengandung sirkuit

Jadi, syarat pohon ada 3: graf tidak berarah, harus terhubung, dan tidak memiliki sirkuit (siklus)



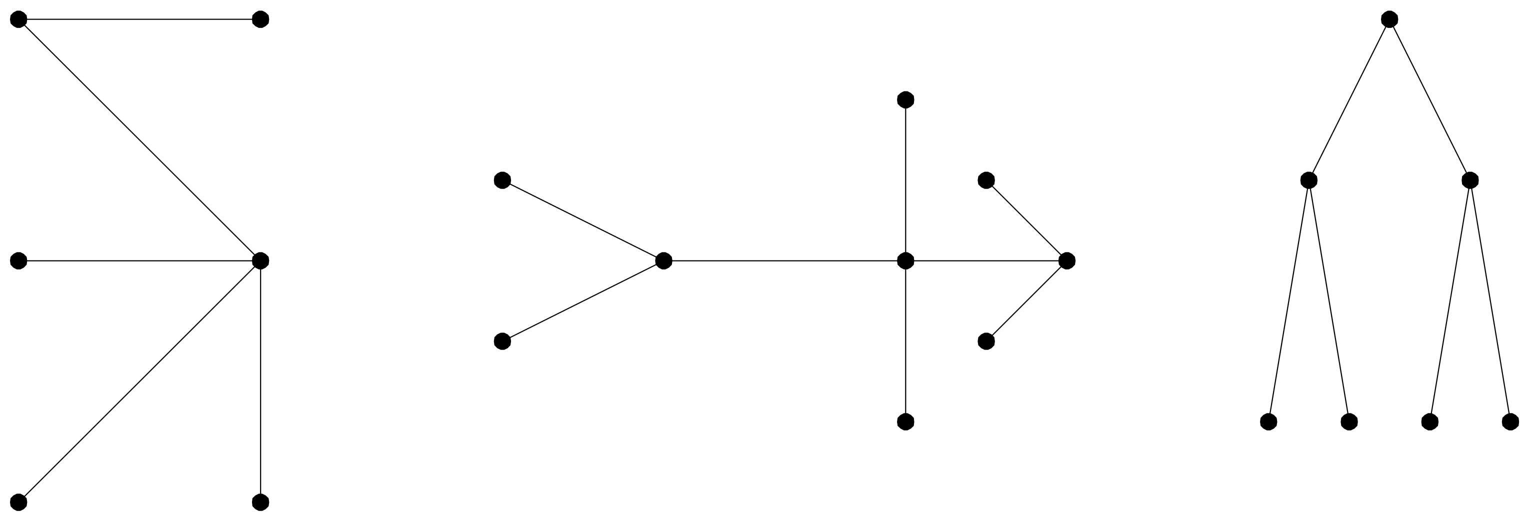
pohon pohon bukan pohon bukan pohon

- Bermacam-macam pohon:



Hutan (*forest*) adalah

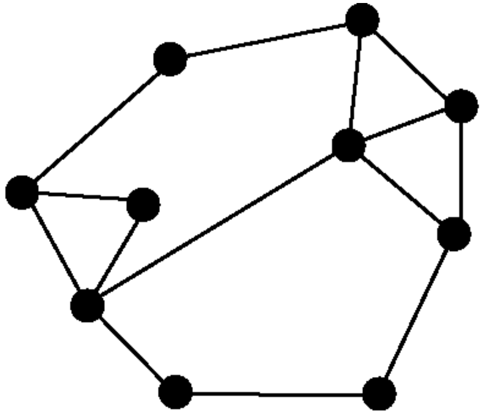
- kumpulan pohon yang saling lepas, atau
- graf tidak terhubung yang tidak mengandung sirkuit. Setiap komponen di dalam graf terhubung tersebut adalah pohon.



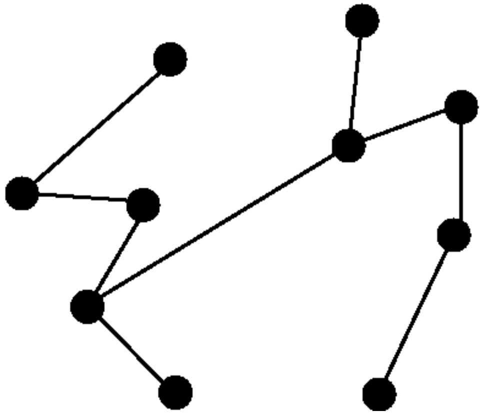
Hutan yang terdiri dari tiga buah pohon

Perbedaan graf, pohon, dan hutan:

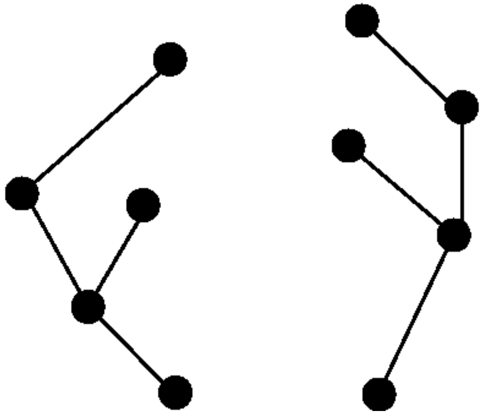
Graph
(with cycles)



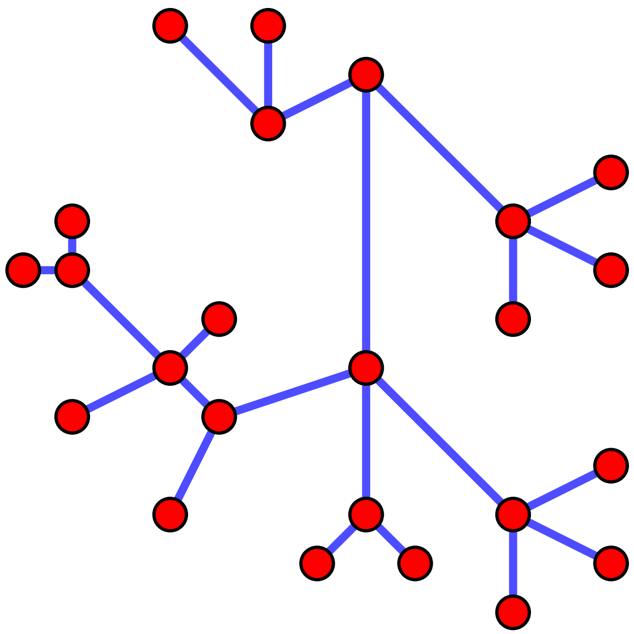
Tree
(no cycles, connected)



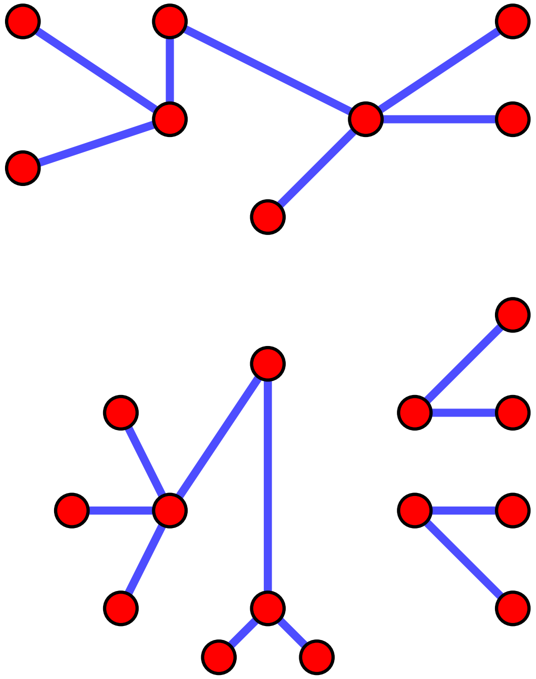
Forest
(no cycles, not connected)



Sumber: https://www.cambridge.org/core/books/abs/applying-graph-theory-in-ecological-research/shapes-of-graphs-trees-to-triangles/6F7A9487D10EF5ED28A83F7B035E7FDC



Tree



Forest

Sumber: https://www.mathreference.org/index/page/id/393/lg/en

Hutan

Sifat-sifat (properti) pohon

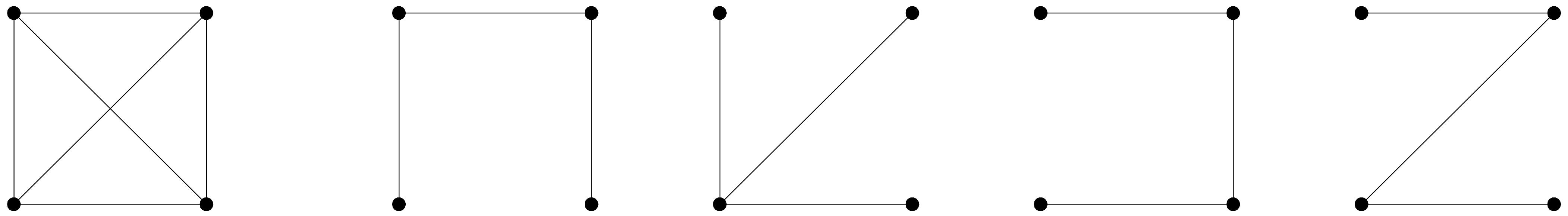
Teorema. Misalkan $G = (V, E)$ adalah graf tak-berarah sederhana dan jumlah simpulnya n. Maka, semua pernyataan di bawah ini adalah ekivalen:

1. G adalah pohon.
2. Setiap pasang simpul di dalam G terhubung dengan lintasan tunggal.
3. G terhubung dan memiliki $m = n - 1$ buah sisi.
4. G tidak mengandung sirkuit dan memiliki $m = n - 1$ buah sisi.
5. G tidak mengandung sirkuit dan penambahan satu sisi pada graf akan membuat hanya satu sirkuit.
6. G terhubung dan semua sisinya adalah jembatan.

- Teorema di atas dapat dikatakan sebagai definisi lain dari pohon.

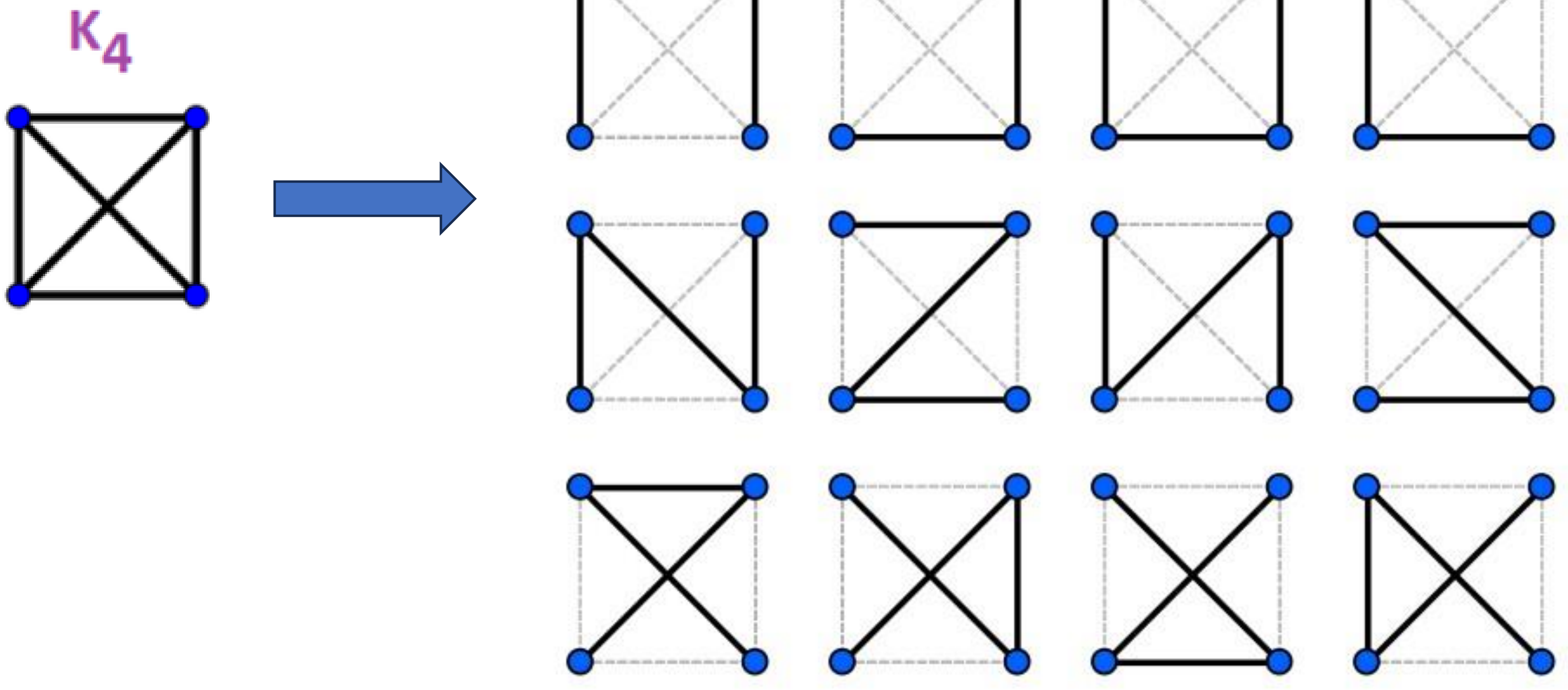
Pohon Merentang (*Spanning Tree*)

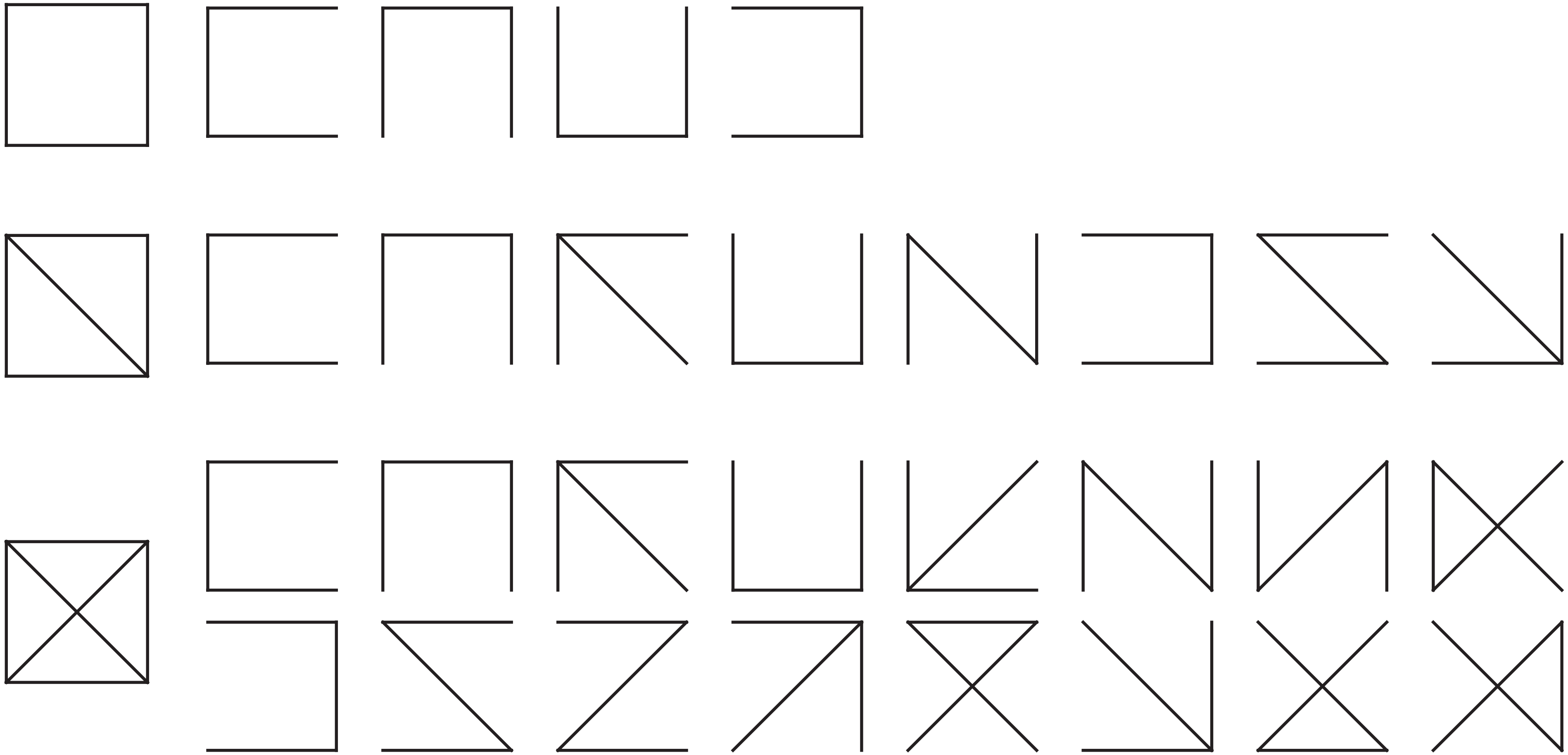
- Pohon merentang dari sebuah graf terhubung adalah upagraf merentang yang berupa pohon.
- Upagraf merentang (*spanning subgraph*) adalah upagraf dari graf G yang mengandung semua simpul di dalam graf G
- Pohon merentang dari graf G diperoleh dengan memutus sirkuit di dalam graf.
- Contoh: Sebagian pohon merentang dari Graf G



G T_1 T_2 T_3 T_4

Pohon merentang lengkap dari graf K_4:



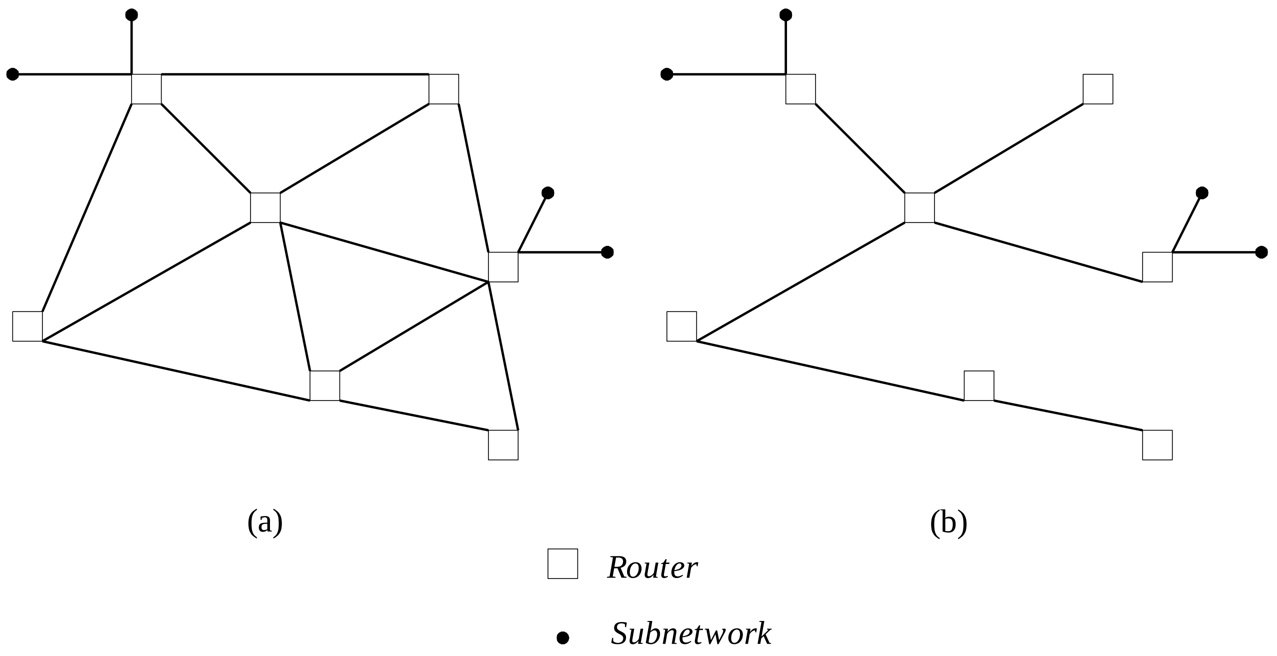


- Setiap graf terhubung mempunyai paling sedikit satu buah pohon merentang.

- Graf tak-terhubung dengan *k* komponen mempunyai *k* buah hutan merentang yang disebut hutan merentang (*spanning forest*).

Aplikasi Pohon Merentang

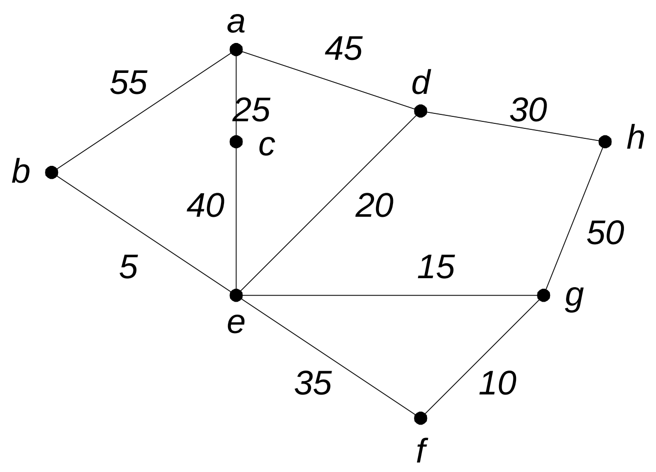
1. Jumlah ruas jalan seminimum mungkin yang menghubungkan semua kota sehingga setiap kota tetap terhubung satu sama lain.
2. Perutean (*routing*) pesan pada jaringan komputer.



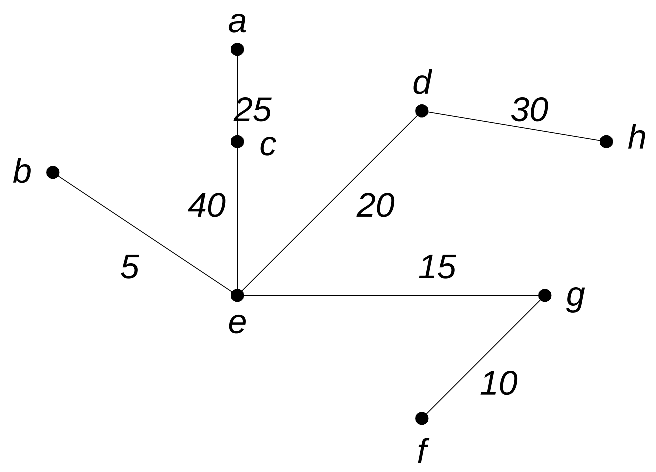
(a) Jaringan komputer, (b) Pohon merentang *multicast*

Pohon Merentang Minimum

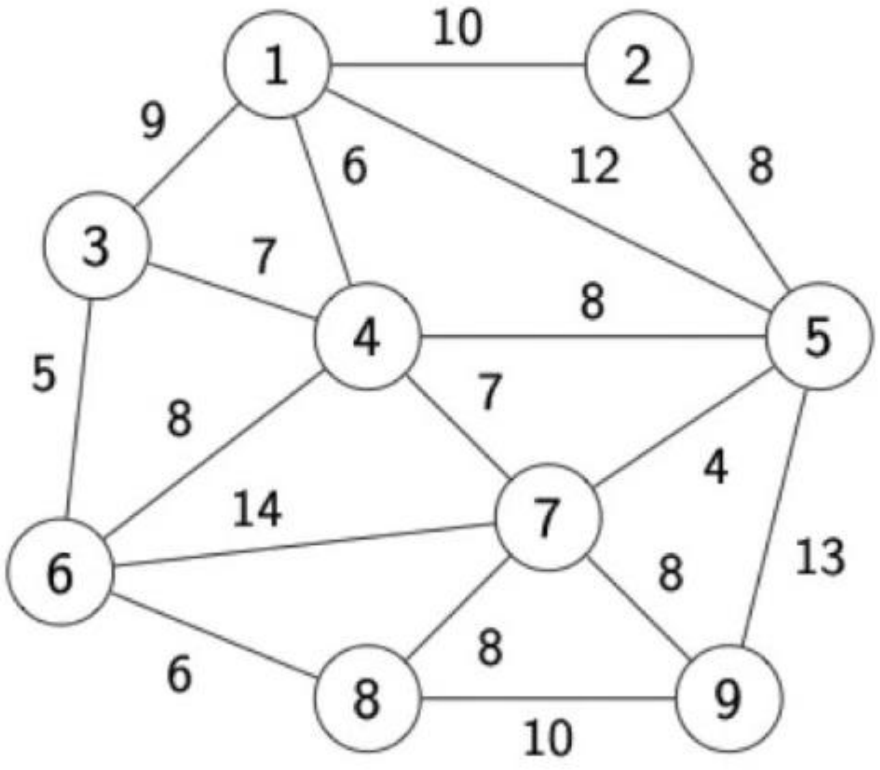
- Graf terhubung-berbobot mungkin mempunyai lebih dari 1 pohon merentang.
- Pohon merentang yang berbobot minimum –dinamakan **pohon merentang minimum** (*minimum spanning tree*).



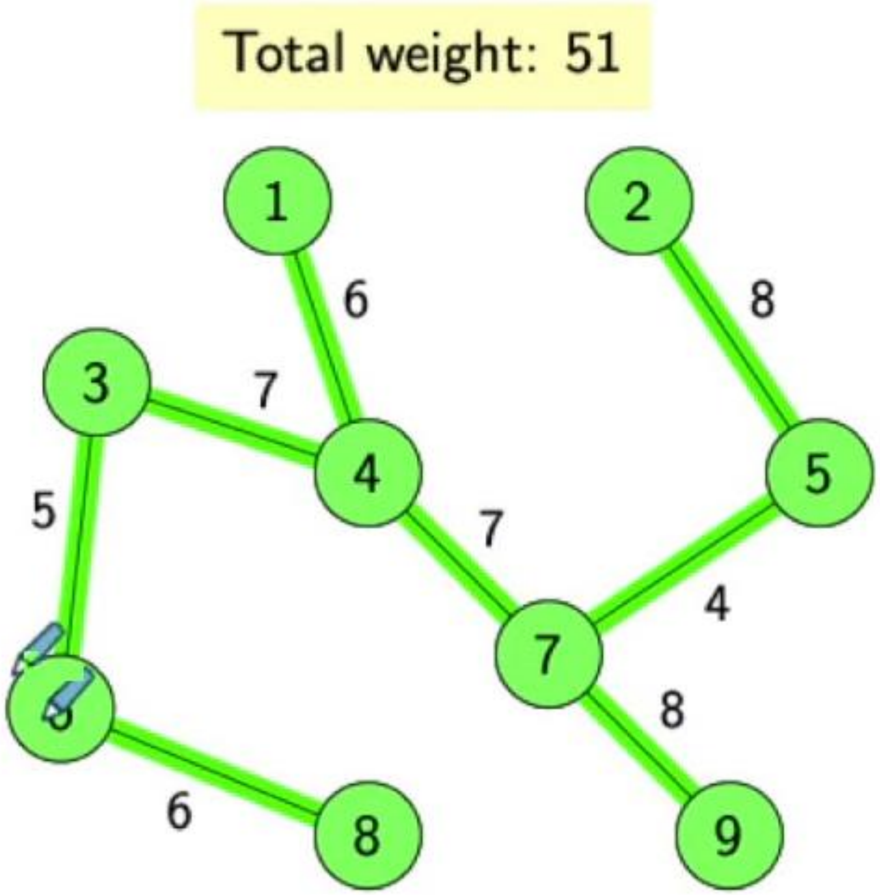
Graf berbobot (*weighted graph*)



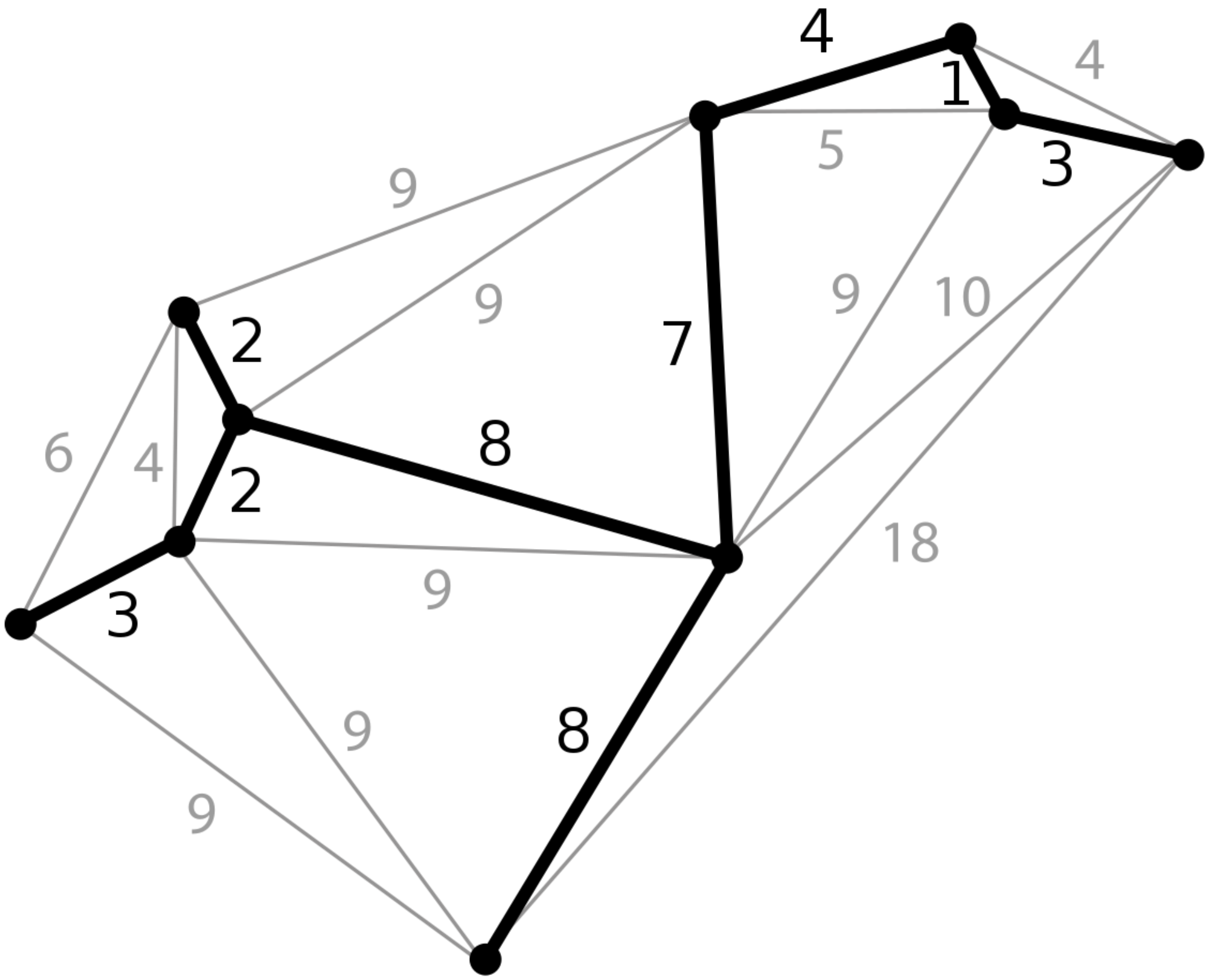
Pohon merentang minimum



Graf berbobot (*weighted graph*)



Pohon merentang minimum



Graf berbobot dan pohon merentang minimum (garis tebal)

- Algoritma untuk menentukan pohon merentang minimum:
    1. Algoritma Prim
    2. Algoritma Kruskal

Algoritma Prim

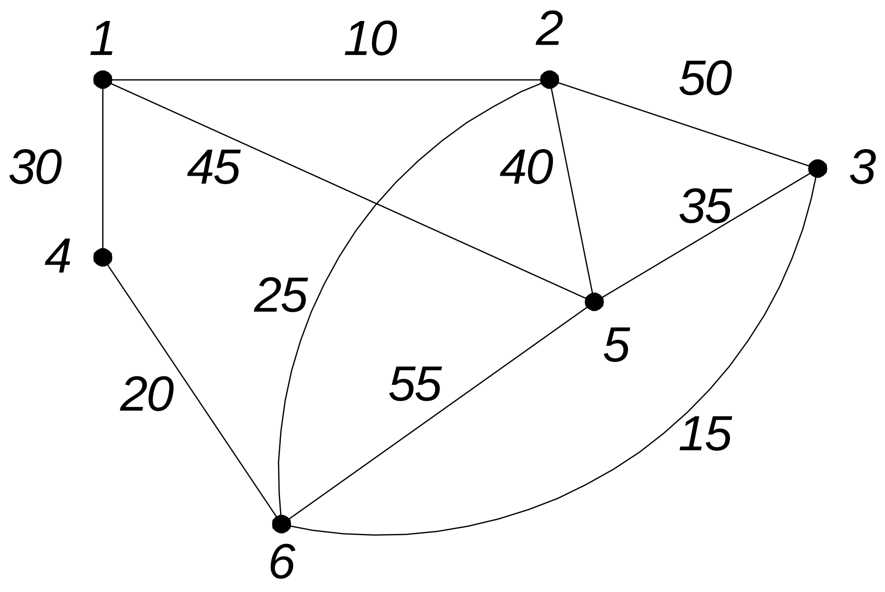
Langkah 1: ambil sisi dari graf *G* yang berbobot minimum, masukkan ke dalam *T*.

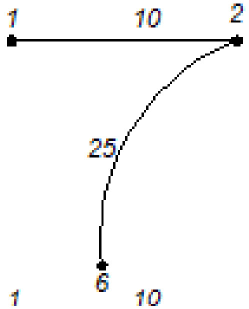
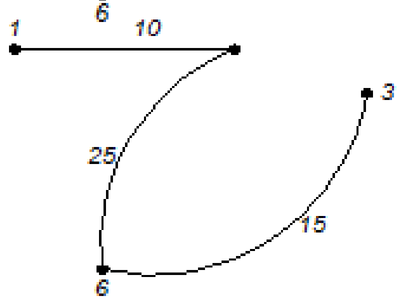
Langkah 2: pilih sisi (*u*, *v*) yang mempunyai bobot minimum dan bersisian dengan simpul di *T*, tetapi (*u*, *v*) tidak membentuk sirkuit di *T*. Masukkan (*u*, *v*) ke dalam *T*.

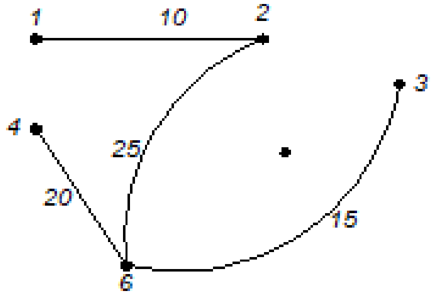
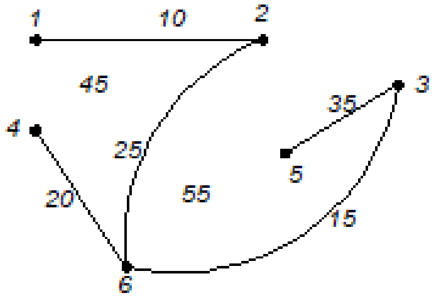
Langkah 3: ulangi langkah 2 sebanyak $n - 2$ kali.

```
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 ← {(p,q)}
  for i←1 to n-2 do
    Pilih sisi (u,v) dari E yang bobotnya terkecil namun
    bersisian dengan simpul di T dan (u,v) tidak membentuk
    sirkuit di T
    T ← T ∪ {(u,v)}
  endfor
```

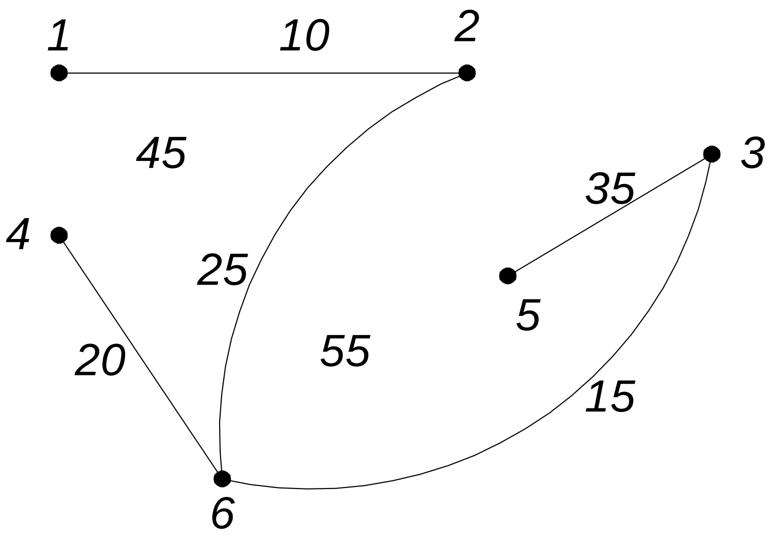
Contoh:



Langkah	Sisi	Bobot	Pohon rentang
1	(1, 2)	10	
2	(2, 6)	25	
3	(3, 6)	15	

Langkah	Sisi	Bobot	Pohon rentang
4	(4, 6)	20	
5	(3, 5)	35	

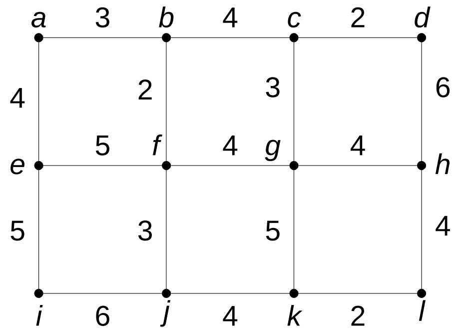
Pohon merentang minimum yang dihasilkan:



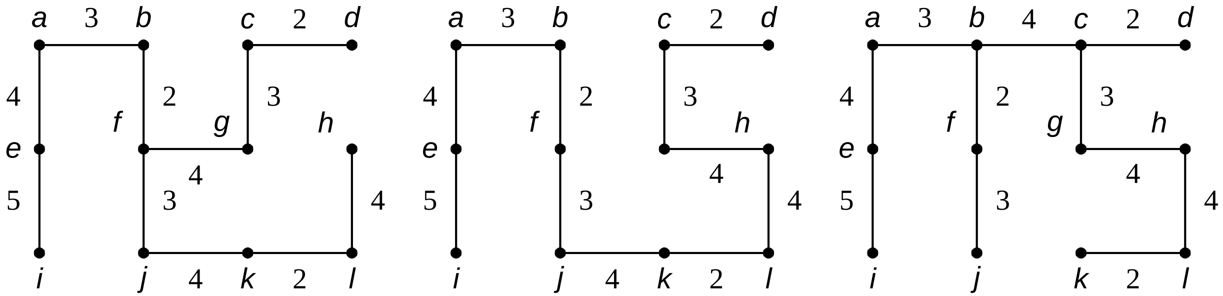
Bobot = 10 + 25 + 15 + 20 + 35 = 105

- Pohon merentang yang dihasilkan tidak selalu unik meskipun bobotnya masih tetap sama.
- Hal ini terjadi apabila terdapat beberapa sisi yang akan dipilih berbobot sama.

Contoh:



Tiga buah pohon merentang minimumnya:



Bobotnya sama yaitu = 36

Algoritma Kruskal

(Langkah 0: sisi-sisi dari graf sudah diurut menaik berdasarkan bobotnya – dari bobot kecil ke bobot besar)

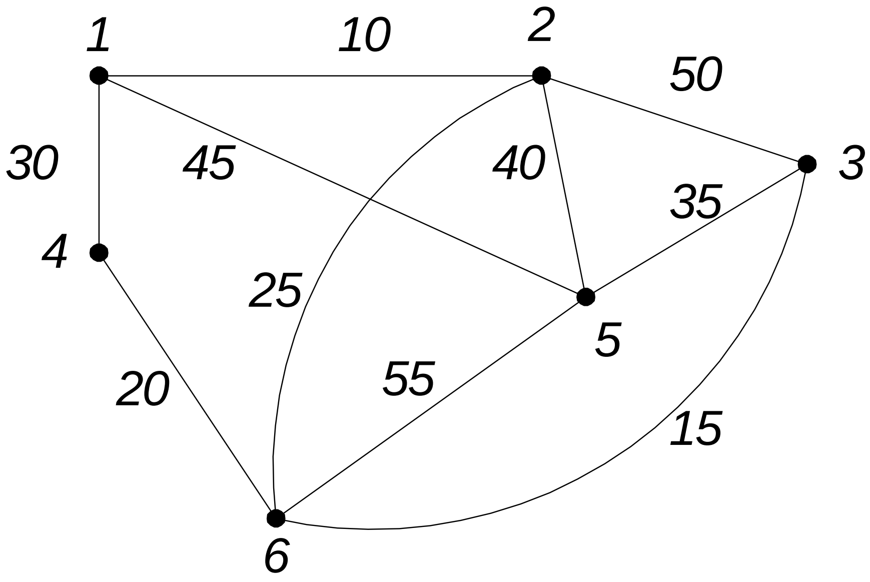
Langkah 1: *T* masih kosong

Langkah 2: pilih sisi (*u*, *v*) dengan bobot minimum yang tidak membentuk sirkuit di *T*. Tambahkan (*u*, *v*) ke dalam *T*.

Langkah 3: ulangi langkah 2 sebanyak *n* – 1 kali.

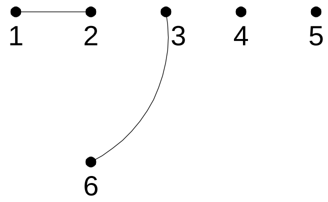
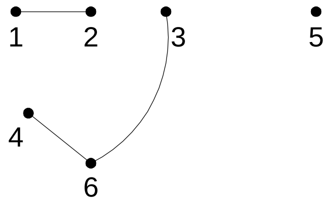
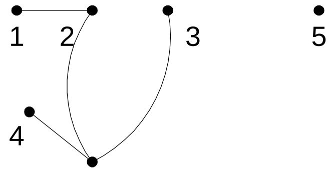
```
procedure Kruskal(input G : graf, output T : pohon)
{ Membentuk pohon merentang minimum T dari graf terhubung –
berbobot G.
Masukan: graf-berbobot terhubung G = (V, E), dengan |V|= n
Keluaran: pohon rentang minimum T = (V, E')
}
Deklarasi
  i, p, q, u, v : integer

Algoritma
  ( Asumsi: sisi-sisi dari graf sudah diurut menaik
     berdasarkan bobotnya – dari bobot kecil ke bobot
     besar)
  T ← {}
  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 ← T ∪ {(u,v)}
    endif
  endfor
```

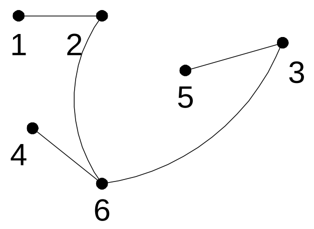
Contoh:



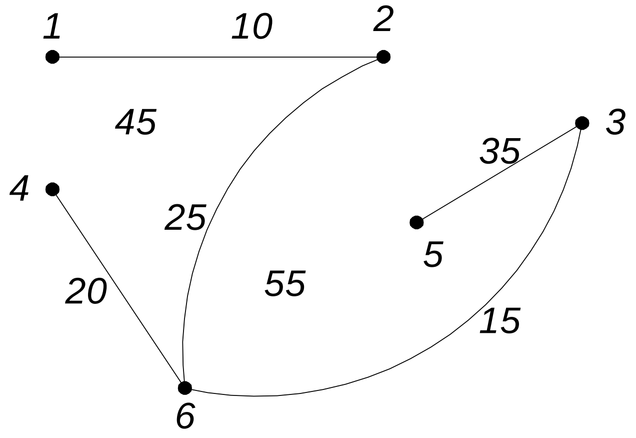
Sisi-sisi diurut menaik:

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	
4	(2, 6)	25	

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

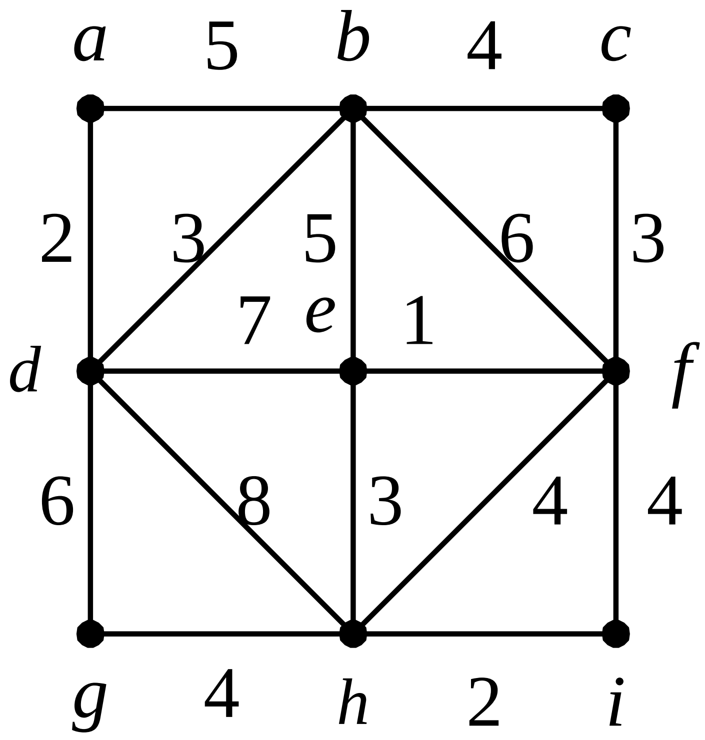
Pohon merentang minimum yang dihasilkan:



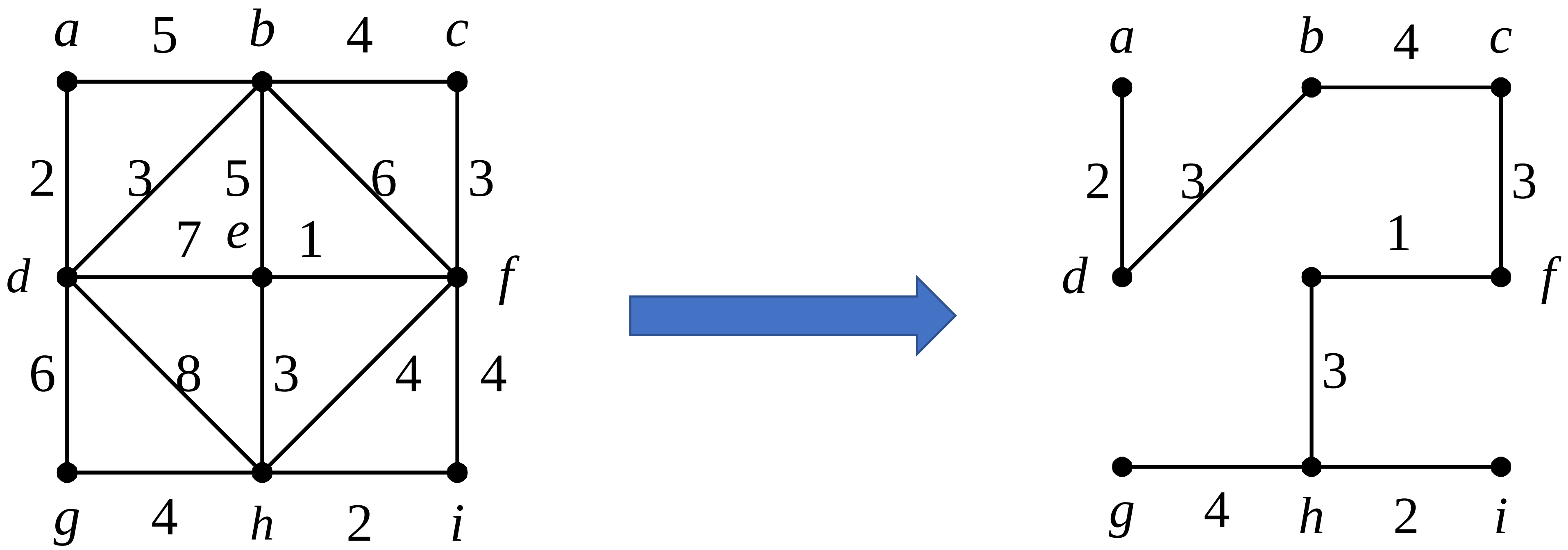
Bobot = 10 + 25 + 15 + 20 + 35 = 105

Latihan

Tentukan dan gambarkan pohon merentang minimum dari graf di bawah ini (tahapan pembentukannya tidak perlu ditulis).



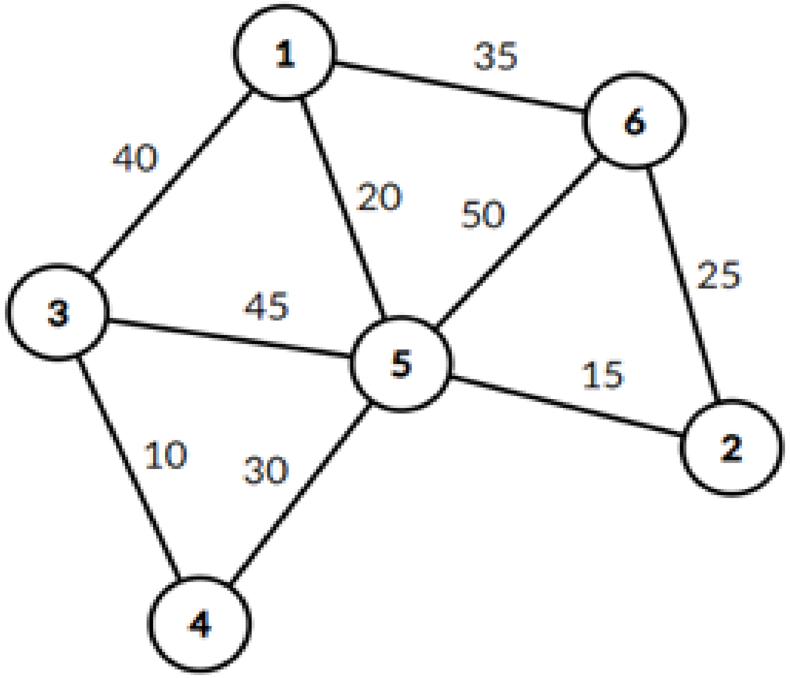
Jawaban:

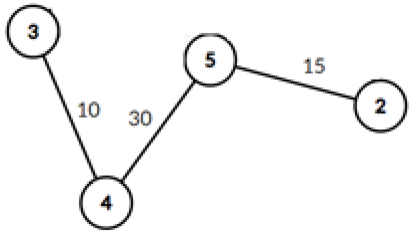


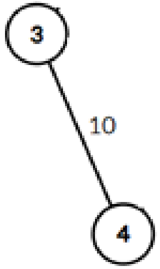
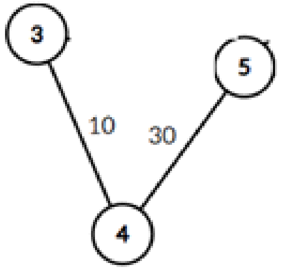
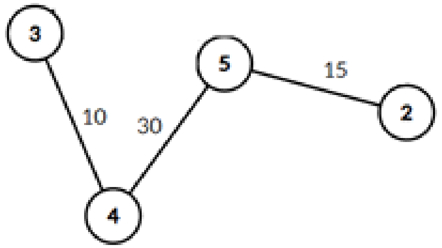
Bobot pohon merentang minimum: 1 + 3 + 3 + 2 + 4 + 4 + 3 + 2 = 22

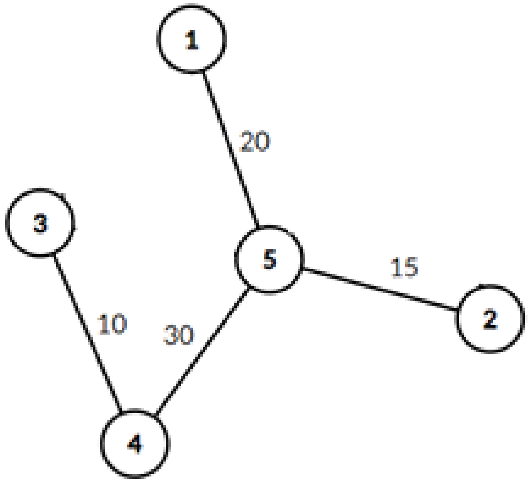
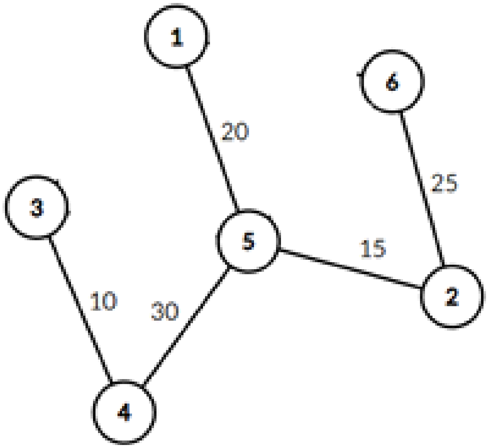
Latihan (Kuis 2021)

Tentukan Pohon Merentang Minimum dari graf berikut ini dengan algoritma Prim (lengkap dengan langkah-langkahnya) dan sebutkan berapa bobot minimumnya.





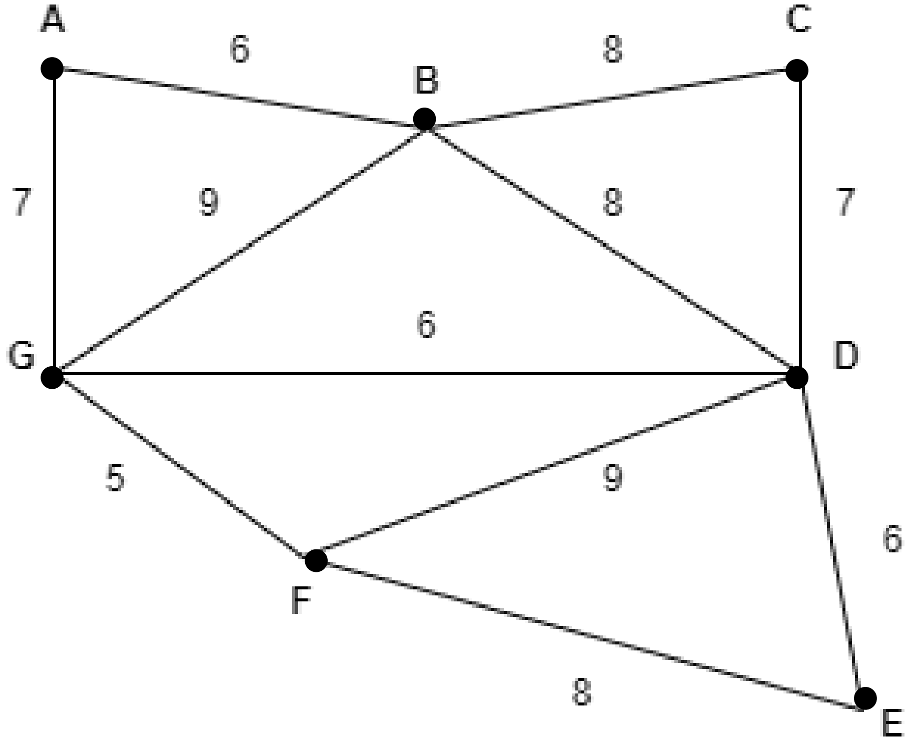
Langkah	Sisi	Bobot	Pohon Rentang
1	(3, 4)	10	
2	(4, 5)	30	
3	(2, 5)	15	

<table>
<tr><td>4</td><td>(1, 5)</td><td>20</td><td>
</td></tr>
<tr><td>5</td><td>(2, 6)</td><td>25</td><td>
</td></tr>
</table>

Total bobot minimumnya adalah 10 + 30 + 15 + 20 + 25 = 100

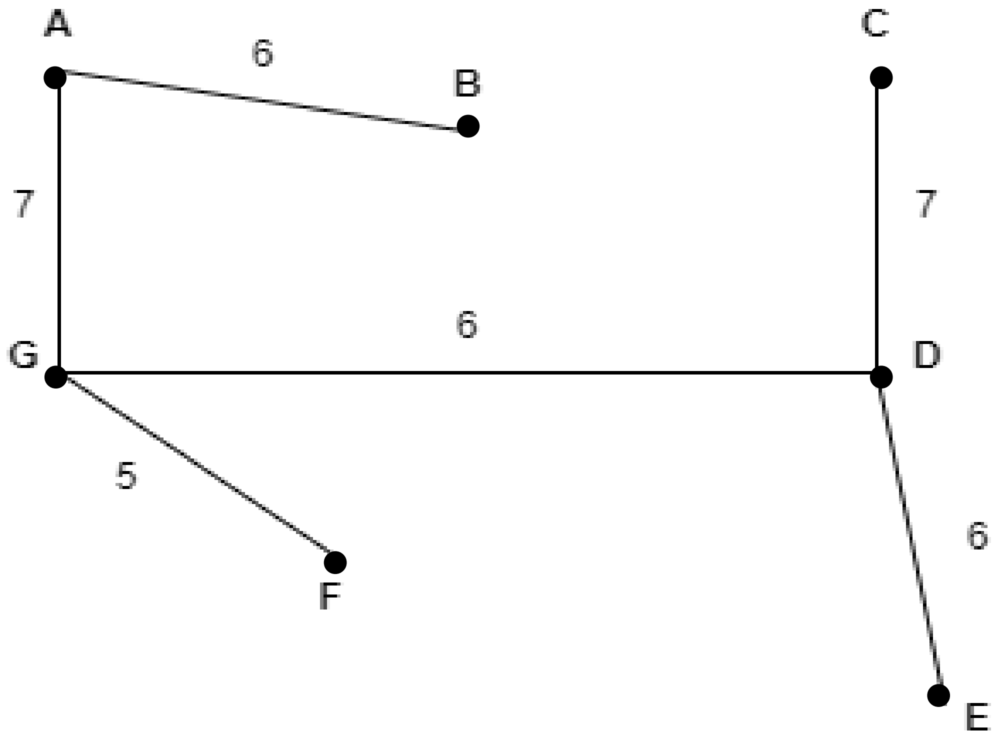
Latihan (Kuis 2022)

- Tentukan pohon merentang minimum dari graf berikut ini dengan algoritma Kruskal (lengkap dengan total bobot dan tiap sisi yang terbentuk di setiap langkah). Di akhir, sebutkan berapa bobot minimumnya beserta pohon akhir yang terbentuk. (Catatan: urutan sisi yang terbentuk berdasarkan simpul dengan abjad terendah).



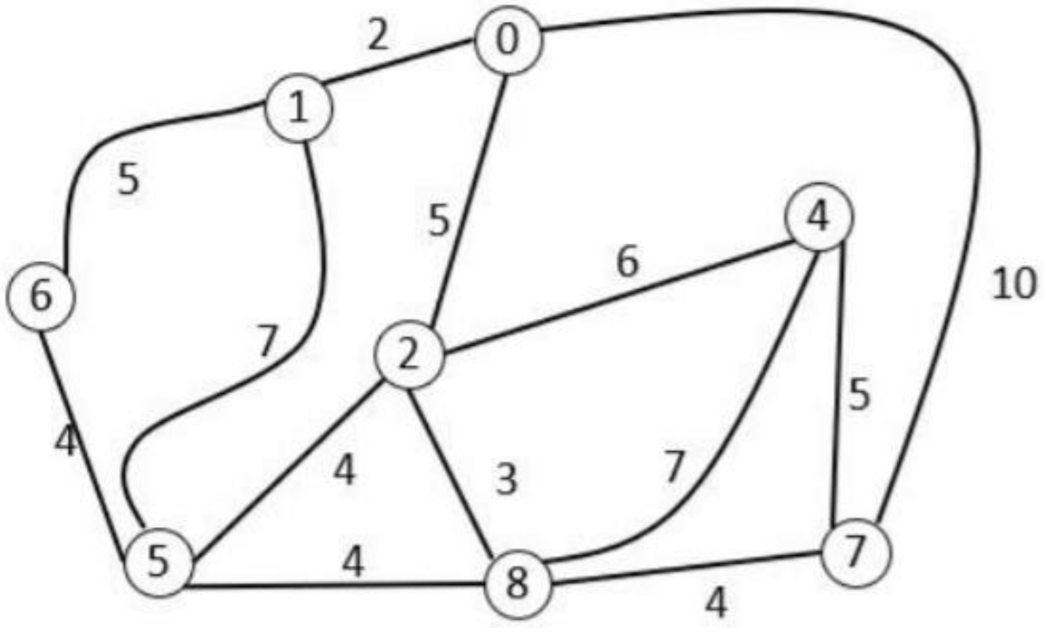
Jawaban:

Langkah	Sisi yang terbentuk	Total Bobot
1	F - G (5)	5
2	A - B (6)	11
3	D - E (6)	17
4	D - G (6)	23
5	A - G (7)	30
6	C - D (7)	37

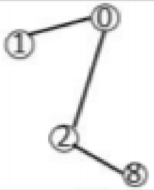
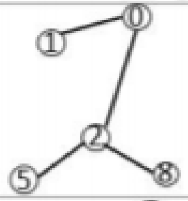
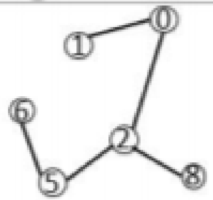
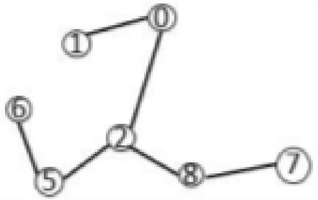
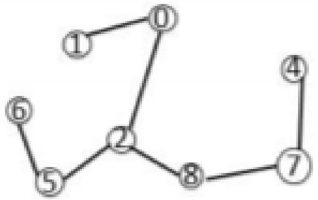


Latihan (Kuis 2020)

Bentuklah minimum spanning tree dari graf berikut dengan menggunakan Algoritma Prim dan tentukan bobot totalnya. Jika terdapat sisi dengan bobot yang sama, utamakan sisi dengan nomor simpul terkecil.



Jawaban:

Langkah	Sisi	Bobot Total	Pohon Rentang
1	(0,1)	2	
2	(0,2)	7	
3	(2,8)	10	
4	(2,5)	14	
5	(5,6)	18	
6	(7, 8)	22	
7	(4,7)	26	

Bersmbung ke Bagian 2