

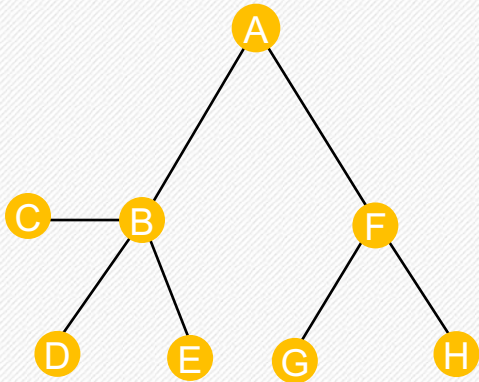
Spanning Tree

Fauzi Adi Rafrastara

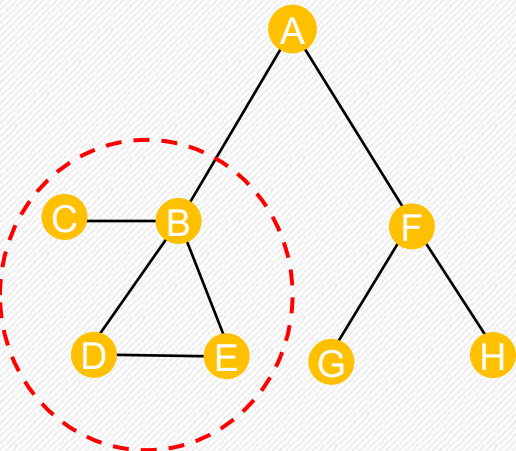
Contents

- Pohon (Tree)
- Pohon Rentang (Spanning Tree)
- Algoritma Kruskal
- Algoritma Prim

Pohon (Tree)

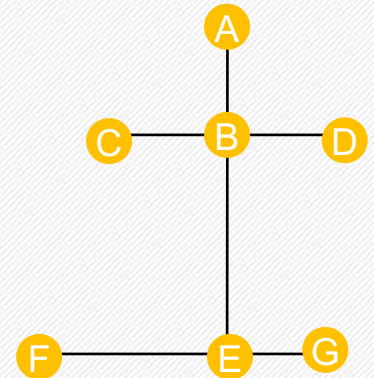


Pohon, karena tersusun dari graf terhubung dan tidak memiliki sirkuit

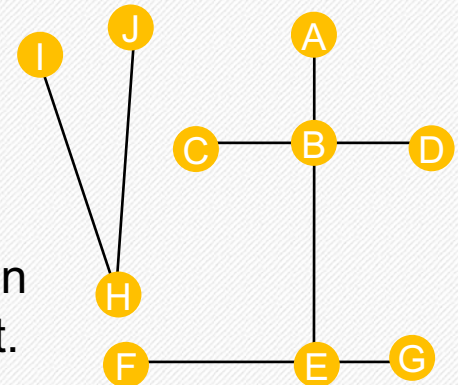


Bukan Pohon, karena meskipun graf terhubung, tapi dia memiliki sirkuit (B-D-E)

Pohon, karena tersusun dari graf terhubung dan tidak memiliki sirkuit. Pohon tidak harus berbentuk seperti tumpukan akar.



Hutan, karena terdiri dari dua pohon. Masing-masing adalah graf terhubung dan tidak punya sirkuit.

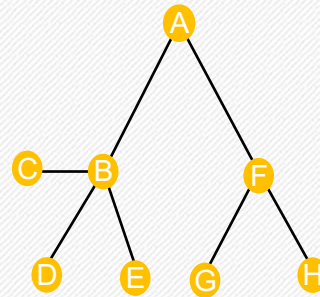


Pohon (Tree) - Jenis

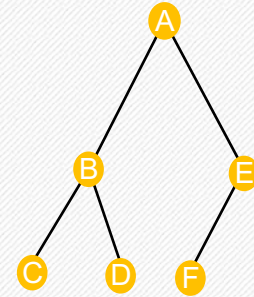


Empty/Null Tree (Pohon kosong):
Pohon yang tidak memiliki titik

Trivial Tree (Pohon semu):
Terdiri dari satu simpul saja, tanpa sisi.



Rooted Tree (Pohon berakar):
Pohon dengan satu titik yang dikhususkan dari yang lain. Titik tersebut disebut akar (Root).



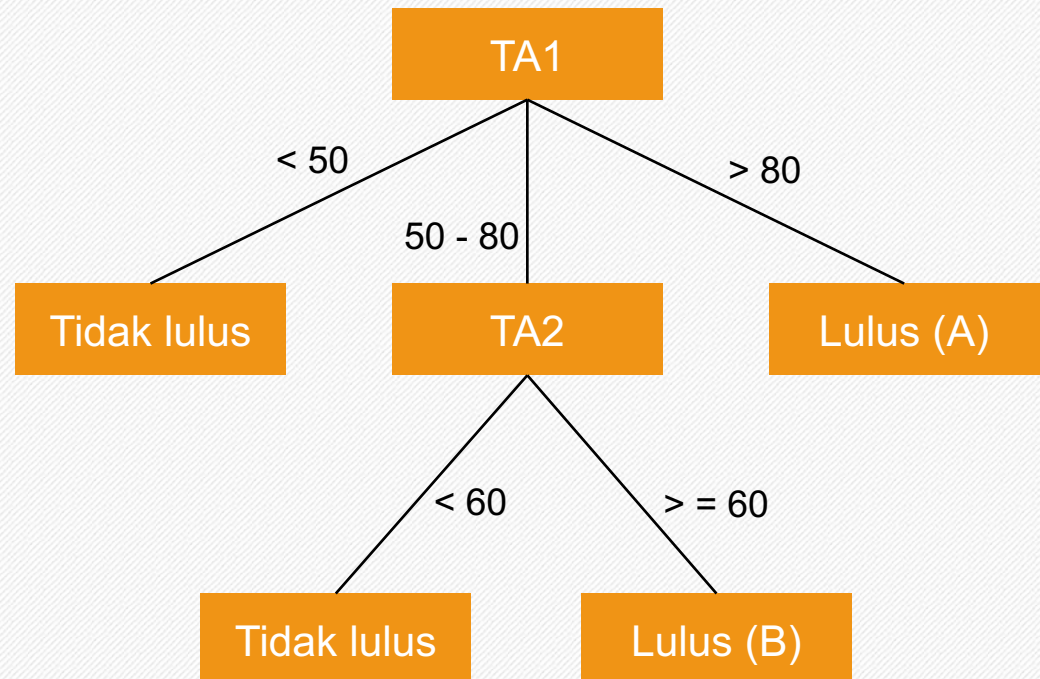
Binary Tree (Pohon biner):
Pohon berakar dengan maksimal dua anak dalam tiap tingkatnya.

Pohon (Tree) - Implementasi

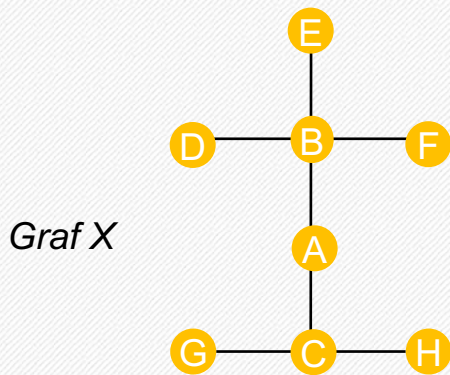
Syarat kelulusan sidang TA:

- Jika nilai ujian TA1 > 80 , maka mhs dinyatakan lulus dengan nilai A.
- Jika nilai ujian TA1 < 50 , maka mhs dinyatakan tidak lulus.
- Jika nilai ujian TA1 antara 50 – 80, maka mhs diwajibkan mengikuti sidang TA2.
- Jika nilai ujian TA2 ≥ 60 , maka nilainya B.
- Jika nilai ujian TA2 < 60 , maka mhs tidak lulus.

Nyatakan syarat kelulusan sidang TA tersebut dalam suatu pohon keputusan (Decision Tree)!



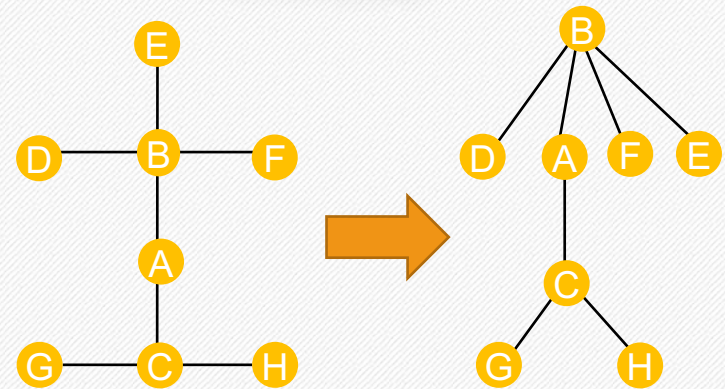
Pohon Berakar (Rooted Tree)



Jika pada Graf X **titik B bertindak sebagai akar**, maka tentukan:

- Tingkat tiap-tiap titik
- Tinggi pohon
- Anak, orang tua dan saudara dari titik A
- Apakah perintah a – c memiliki hasil yang sama **jika akarnya adalah titik A? Buktikan!**

Hasil pembuatan pohon dengan akar B dari Graf X.



Jawaban:

- Tentukan tingkat tiap-tiap titik!**

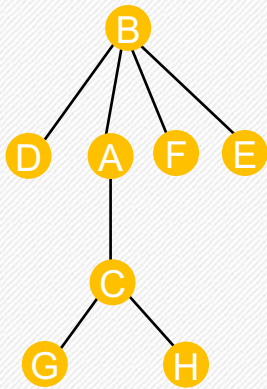
Makna tingkat adalah banyaknya garis yang menghubungkan titik tersebut hingga ke akar (B)

Tingkat D adalah jumlah garis dari D ke akar (B), yaitu 1. Dengan demikian, **maka tingkat D = tingkat A = tingkat F = tingkat E = 1.**

Tingkat C = 2, karena butuh dua garis untuk menghubungkan C ke B, yaitu melalui A.

Tingkat G = tingkat H = 3.

Pohon Berakar (Rooted Tree)



b. Tentukan tinggi pohon!

Tinggi pohon = tingkat maksimal yang dimiliki oleh pohon berakar, yaitu dari ujung atas (akar) sampai ujung bawah.

Dalam kasus ini, karena tingkat maksimalnya adalah 3, **maka tinggi pohonnya = 3.**

c. Tentukan anak, orang tua dan saudara dari A!

Berdasarkan pohon yang telah terbentuk, yang dimaksud dengan anak dari A, adalah titik yang persis ada satu level di bawah A, yaitu C.

Orang tua A, adalah titik yang ada tepat satu level di atas A, yaitu B.

Saudara A, adalah titik yang berada satu level dengan A dari orang tua yang sama. Dalam kasus ini adalah D, F, dan E.

Jadi,

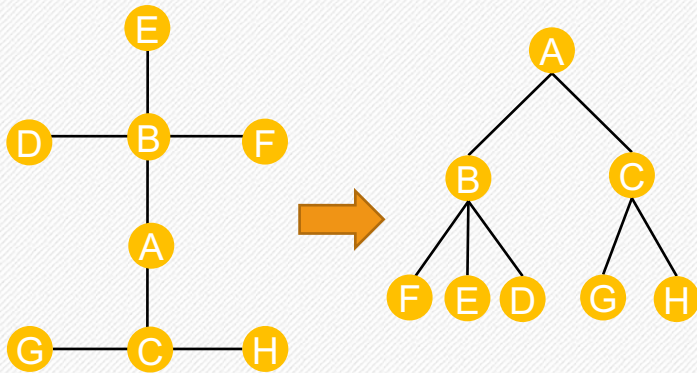
Anak A = C

Orang tua A = B

Saudara A = D, F, dan E

Pohon Berakar (Rooted Tree)

d. Jika akarnya adalah A, maka hasilnya adalah sebagai berikut!



Tingkat tiap-tiap titik!

- Tingkat B = Tingkat C = 1
- Tingkat D = Tingkat E =
Tingkat F = Tingkat G =
Tingkat H = 2

Tinggi pohon = 2

Orang tua A : Tidak ada (karena akar)

Anak A : B dan C

Saudara A : Tidak ada (karena akar)

Kesimpulan:

Dengan adanya akar yang berbeda, maka pohon yang terbentuk juga berbeda.

Dengan demikian, segala hal yang berhubungan dengan pohon tersebut pun juga berbeda, seperti tingkat, tinggi, orang tua, anak, juga saudara.

Pohon Rentang (Spanning Tree)

Pohon Rentang memiliki banyak aplikasi, khususnya dalam bidang riset operasi.

Riset operasi berhubungan dengan penerapan metode-metode ilmiah (matematis) pada permasalahan sehari-hari, baik terkait dengan bisnis, ekonomi, social, atau bidang lainnya untuk mendapatkan solusi yang optimal.

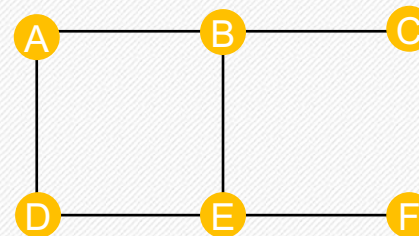
Pohon Rentang suatu graf terhubung G adalah subgraph G yang berbentuk pohon dan memuat semua titik dalam graf G .

1 graf G bisa memiliki banyak pohon rentang.

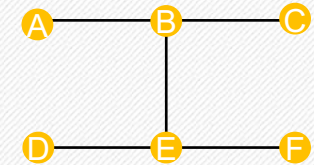
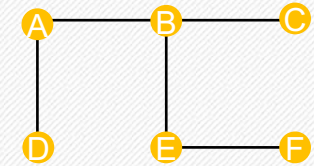
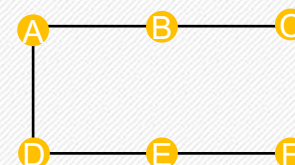
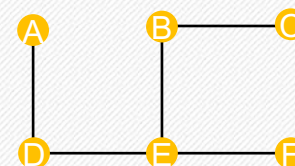
Pohon rentang tidak boleh memiliki sirkuit.

Pohon rentang berhubungan dengan graf berbobot.

Contoh:



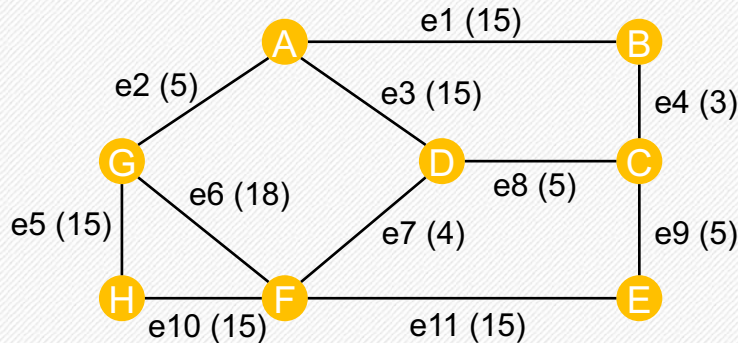
Pohon rentang yang terbentuk dari graf di atas adalah:



Pohon Rentang Minimum

Pohon Rentang Minimum

berhubungan dengan graf berbobot (berlabel).



Contoh implementasi:

Pemasangan pipa air, jaringan listrik, jaringan komputer yang paling efisien.

Untuk menghitung pohon rentang minimum, ada 2 algoritma yang populer digunakan, yaitu

1. Algoritma Kruskal
2. Algoritma Prim

Pada **algoritma kruskal**, penyusunan diurutkan dari bobot terkecil (tidak boleh membentuk sirkuit)

Pada **algoritma prim**, penyusunan diurutkan dari titik awal keberangkatan, dicari jalur2 dengan bobot terkecil, hingga semua selesai dijangkau (tidak boleh membentuk sirkuit)

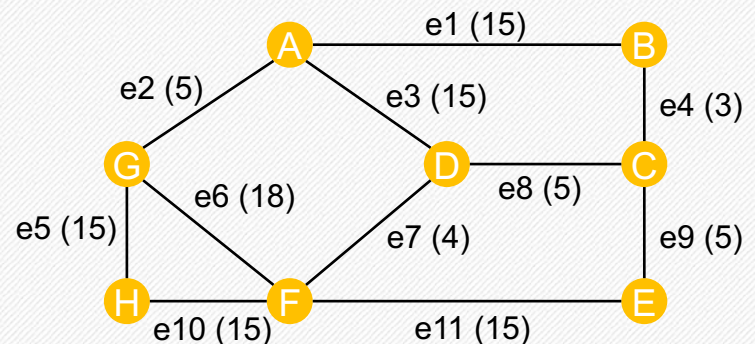
Pohon Rentang Minimum - *konversi*

Sisi	Titik yang dihubungkan	Bobot
e4	B – C	3
e7	D – F	4
e2	A – G	5
e8	C – D	5
e9	C – E	5
e1	A – B	15
e3	A – D	15
e10	F – H	15
e5	G – H	15
e11	E – F	15
e6	F – G	18

Berdasarkan tabel di samping, buatlah graf berbobotnya!

Jawaban

Graf berbobot:



Pohon Rentang Minimum - *konversi*

Sisi	Titik yang dihubungkan	Bobot
e4	B – C	3
e7	D – F	4
e2	A – G	5
e8	C – D	5
e9	C – E	5
e1	A – B	15
e3	A – D	15
e10	F – H	15
e5	G – H	15
e11	E – F	15
e6	F – G	18

Berdasarkan tabel di samping, buatlah matriks hubungannya!

Jawaban:

A =

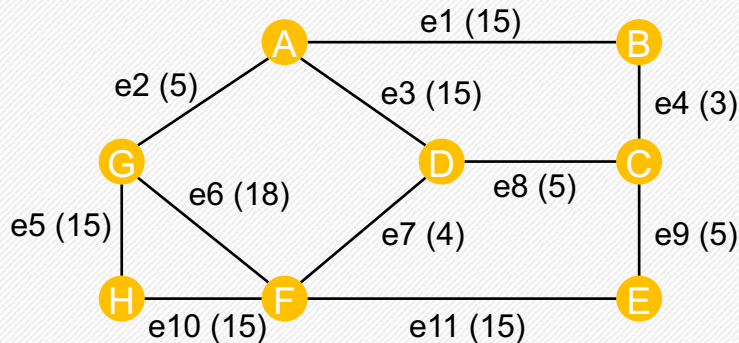
	A	B	C	D	E	F	G	H
A	0	15		15			5	
B	15	0	3					
C		3	0	5	5			
D	15		5	0		4		
E			5		0	15		
F				4	15	0	18	15
G	5					18	0	15
H						15	15	0

Algoritma Kruskal

Fauzi Adi Rafrastara

Algoritma Kruskal

Pohon Rentang Minimum berhubungan dengan graf berbobot (berlabel).



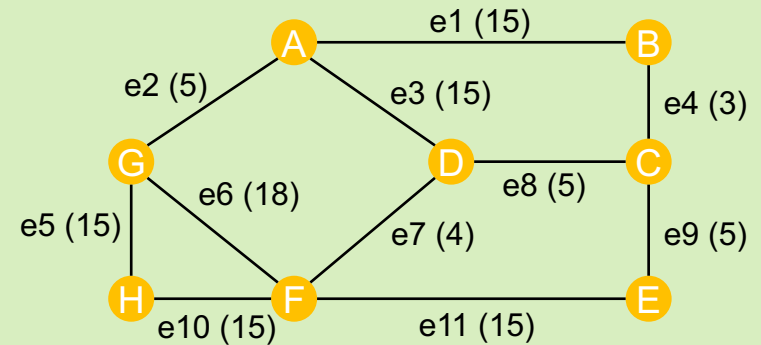
Carilah bobot total minimum graf di atas menggunakan algoritma kruskal!

Langkah 1.

Gambar ulang grafnya, namun tanpa garis-garis sisi dan tersisa titik-titiknya saja, seperti berikut ini.



Algoritma Kruskal



Langkah 2.

Buatlah tabel seperti berikut ini, lalu masukkan data-data sisi dan bobot yang ada:

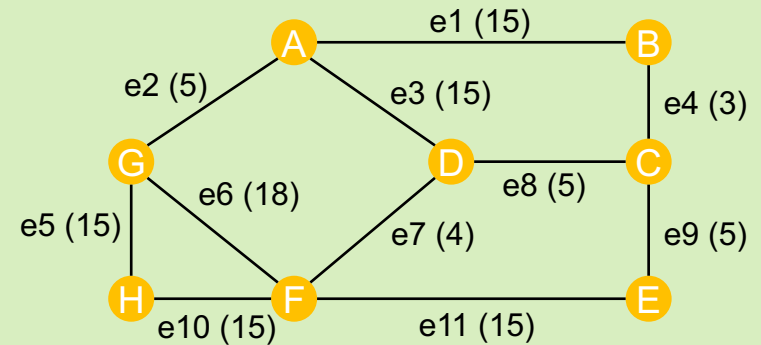
Sisi	Bobot	No sirkuit?
e1	15	
e2	5	
e3	15	
e4	3	
e5	15	
e6	18	
e7	4	
e8	5	
e9	5	
e10	15	
e11	15	

Langkah 3.

Urutkan data di tabel mulai dari bobot terkecil!

Sisi	Bobot	No sirkuit?
e4	3	
e7	4	
e2	5	
e8	5	
e9	5	
e1	15	
e3	15	
e5	15	
e10	15	
e11	15	
e6	18	

Algoritma Kruskal



Langkah 4.

Isi kolom ketiga pada tabel berikut dan pastikan tidak ada sirkuit yang terbentuk.

Sisi	Bobot	No sirkuit?
e4	3	✓
e7	4	
e2	5	
e8	5	
e9	5	
e1	15	
e3	15	
e5	15	
e10	15	
e11	15	
e6	18	

Dalam tahap ini, kita akan gunakan graf kosong yang ada pada langkah 1, lalu kita gambarkan jalurnya satu per satu.

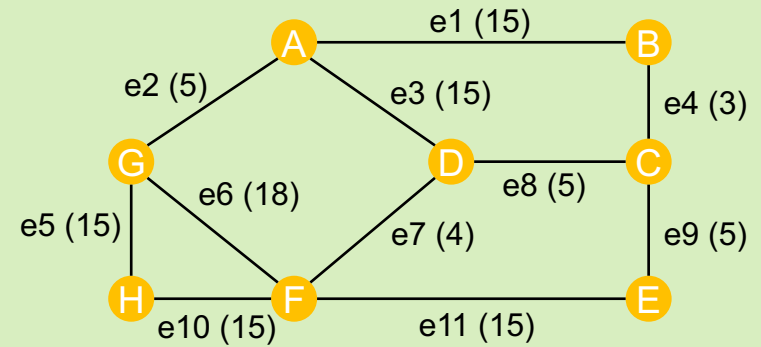
Pertama kita pastikan bahwa e4 adalah sisi dengan bobot terkecil.

Karena ini adalah sisi pertama, jadi tidak mungkin ada sirkuit.

Dengan begitu, kita bisa gambarkan garis e4, yaitu dari B ke C.



Algoritma Kruskal



Langkah 5.

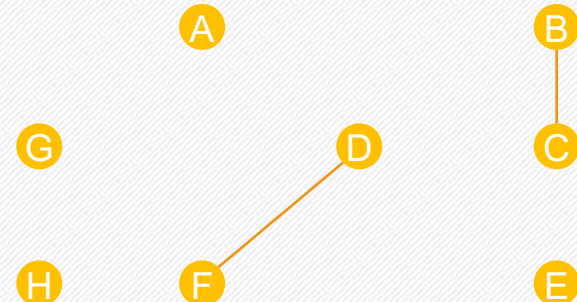
Isi kolom ketiga pada tabel berikut dan pastikan tidak ada sirkuit yang terbentuk.

Sisi	Bobot	No sirkuit?
e4	3	✓
e7	4	✓
e2	5	
e8	5	
e9	5	
e1	15	
e3	15	
e5	15	
e10	15	
e11	15	
e6	18	

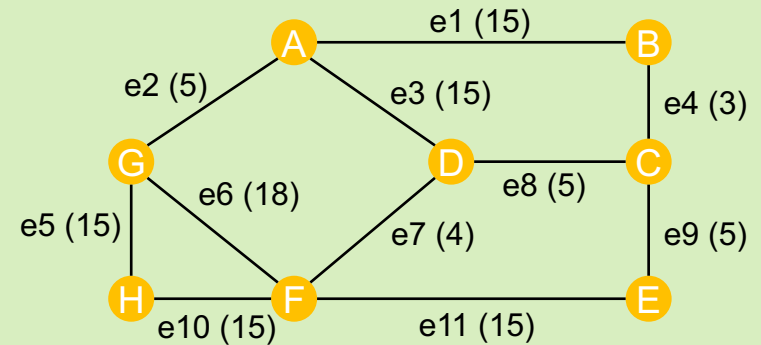
Sisi yang paling kecil berikutnya adalah sisi e7, dengan bobot 4.

Karena ini adalah sisi kedua, jadi belum bisa membentuk sirkuit.

Jadi, kita gambar garis e7, yaitu dari D ke F.



Algoritma Kruskal



Langkah 6.

Isi kolom ketiga pada tabel berikut dan pastikan tidak ada sirkuit yang terbentuk.

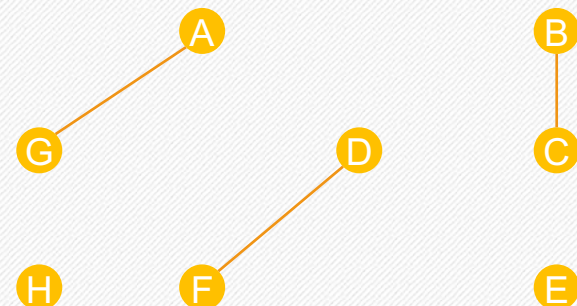
Sisi	Bobot	No sirkuit?
e4	3	✓
e7	4	✓
e2	5	✓
e8	5	
e9	5	
e1	15	
e3	15	
e5	15	
e10	15	
e11	15	
e6	18	

Sisi dengan bobot terkecil berikutnya adalah e2, e8 dan e9 dengan bobot yang sama yaitu 5.

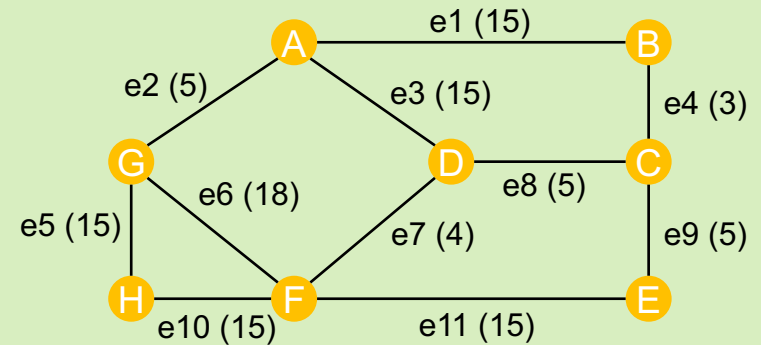
Di sini kita putuskan untuk mengerjakannya satu persatu secara urut dari e2, e8 lalu terakhir e9.

Kita pastikan jika e2 dipasang, tidak akan membentuk sirkuit.

Jika sudah, maka kita gambar garis e2 yang membentang dari A ke G ke graf berikut.



Algoritma Kruskal



Langkah 7.

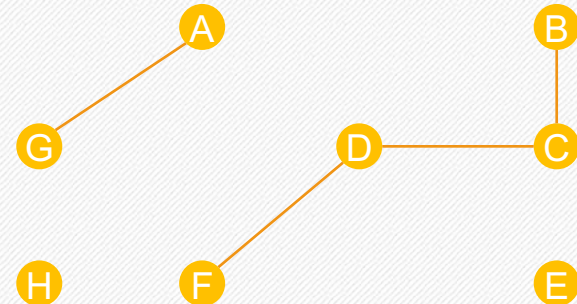
Isi kolom ketiga pada tabel berikut dan pastikan tidak ada sirkuit yang terbentuk.

Sisi	Bobot	No sirkuit?
e4	3	✓
e7	4	✓
e2	5	✓
e8	5	✓
e9	5	
e1	15	
e3	15	
e5	15	
e10	15	
e11	15	
e6	18	

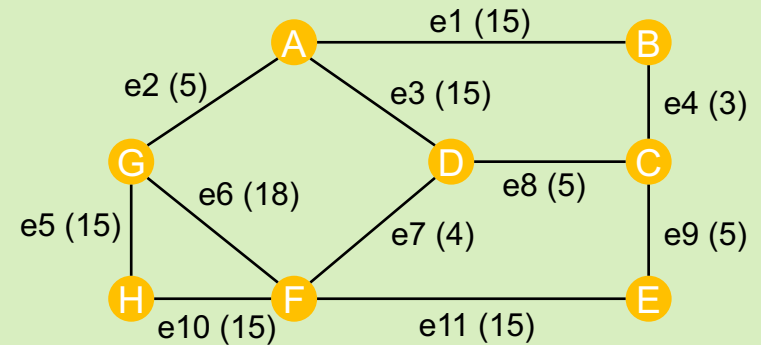
Sisi dengan bobot terkecil berikutnya adalah e8 dengan bobot 5.

Kita pastikan jika e8 dipasang, tidak akan membentuk sirkuit.

Jika sudah, maka kita gambar garis e8 yang membentang dari C ke D ke graf berikut.



Algoritma Kruskal



Langkah 8.

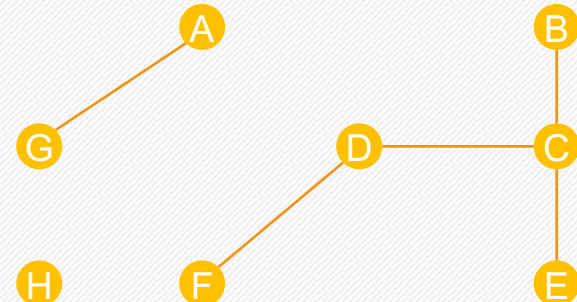
Isi kolom ketiga pada tabel berikut dan pastikan tidak ada sirkuit yang terbentuk.

Sisi	Bobot	No sirkuit?
e4	3	✓
e7	4	✓
e2	5	✓
e8	5	✓
e9	5	✓
e1	15	
e3	15	
e5	15	
e10	15	
e11	15	
e6	18	

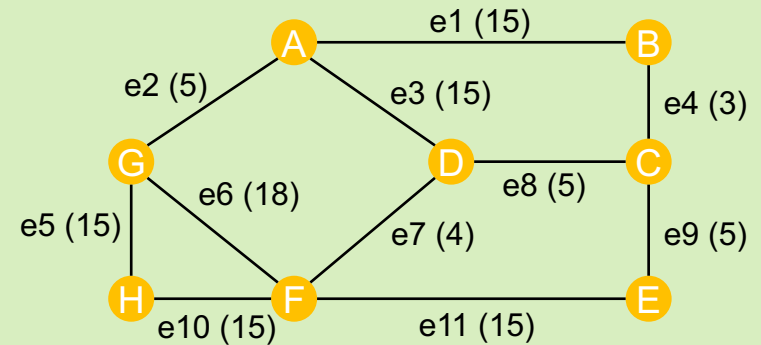
Sisi dengan bobot terkecil berikutnya adalah e9 dengan bobot 5.

Kita pastikan jika e9 dipasang, tidak akan membentuk sirkuit.

Jika sudah, maka kita gambar garis e9 yang membentang dari C ke E ke graf berikut.



Algoritma Kruskal



Langkah 9.

Isi kolom ketiga pada tabel berikut dan pastikan tidak ada sirkuit yang terbentuk.

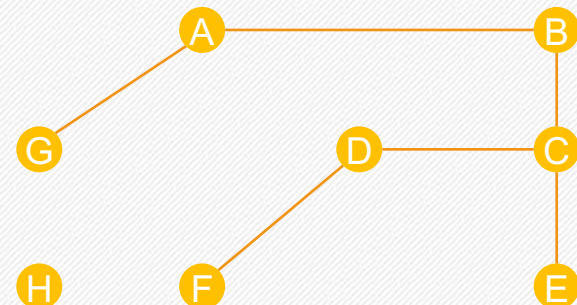
Sisi	Bobot	No sirkuit?
e4	3	✓
e7	4	✓
e2	5	✓
e8	5	✓
e9	5	✓
e1	15	✓
e3	15	
e5	15	
e10	15	
e11	15	
e6	18	

Sisi dengan bobot terkecil berikutnya adalah e1, e3, e5, e10 dan e11 dengan bobot 15.

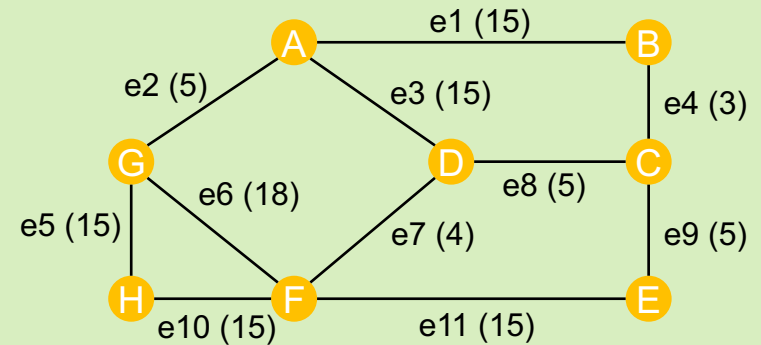
Kita sepakati dulu, proses penghitungan kita lakukan berdasarkan urutan e1, e3, e5, e10 dan e11.

Kita cek e1, apakah akan membentuk sirkuit?

Ternyata tidak! Maka kita gambar e1 ke graf berikut ini.



Algoritma Kruskal



Langkah 10.

Isi kolom ketiga pada tabel berikut dan pastikan tidak ada sirkuit yang terbentuk.

Sisi	Bobot	No sirkuit?
e4	3	✓
e7	4	✓
e2	5	✓
e8	5	✓
e9	5	✓
e1	15	✓
e3	15	✗
e5	15	
e10	15	
e11	15	
e6	18	

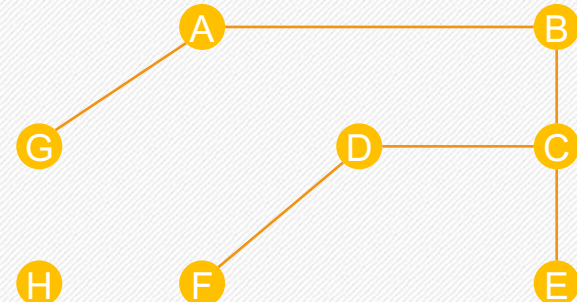
Sisi dengan bobot terkecil berikutnya adalah e3 dengan bobot 15.

Kita cek e3, apakah akan membentuk sirkuit?

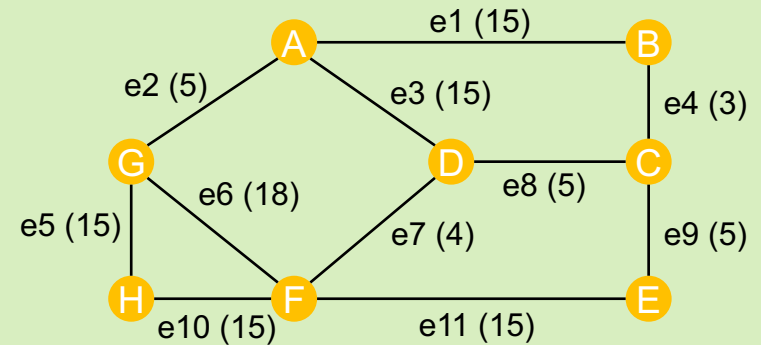
Ternyata iya! Jika sisi e3 dipasang, maka akan membentuk sirkuit, yaitu pada titik A-B-C-D.

Maka dari itu, sisi ini bisa diabaikan.

Kita beri tanda silang di tabel, untuk sisi e3.



Algoritma Kruskal



Langkah 11.

Isi kolom ketiga pada tabel berikut dan pastikan tidak ada sirkuit yang terbentuk.

Sisi	Bobot	No sirkuit?
e4	3	✓
e7	4	✓
e2	5	✓
e8	5	✓
e9	5	✓
e1	15	✓
e3	15	✗
e5	15	✓
e10	15	✗
e11	15	
e6	18	

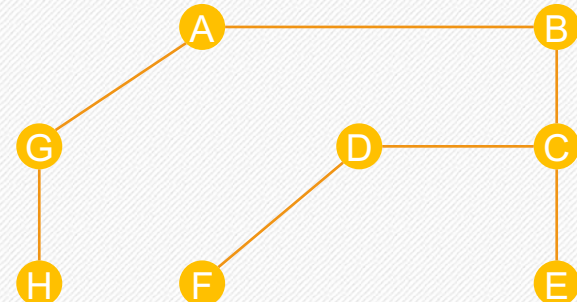
Sisi dengan bobot terkecil berikutnya adalah e5 dan e10 dengan bobot sama-sama 15.

Kita cek e5 dan e10, apakah salah satunya akan membentuk sirkuit?

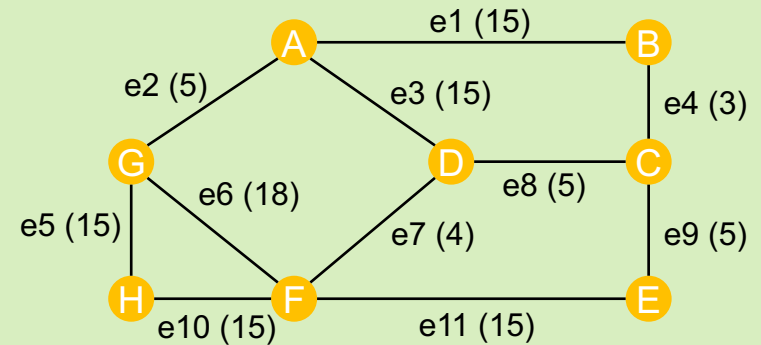
Ternyata tidak!

Maka bisa kita pilih salah satu, yaitu e5. Selanjutnya kita gambar jalur e5 yang menghubungkan G dan H.

Sisi e10 kita abaikan (coret) karena akan membentuk sirkuit jika e5 dipasang.



Algoritma Kruskal



Langkah 12.

Isi kolom ketiga pada tabel berikut dan pastikan tidak ada sirkuit yang terbentuk.

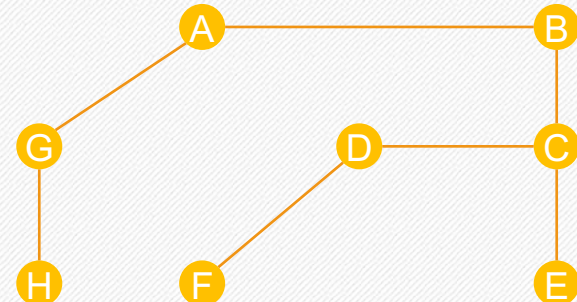
Sisi	Bobot	No sirkuit?
e4	3	✓
e7	4	✓
e2	5	✓
e8	5	✓
e9	5	✓
e1	15	✓
e3	15	✗
e5	15	✓
e10	15	✗
e11	15	✗
e6	18	

Sisi dengan bobot terkecil berikutnya adalah e11 dengan bobot 15.

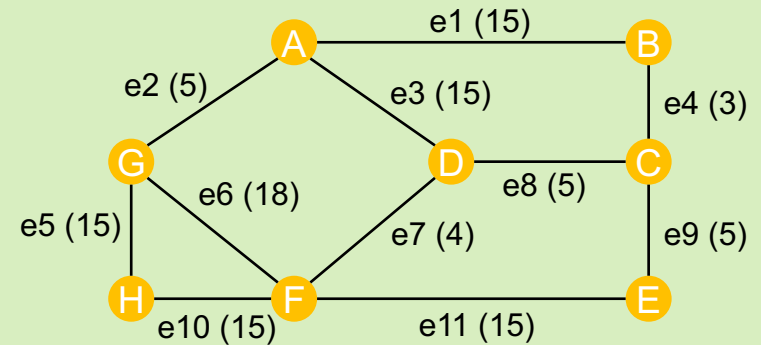
Kita cek e11, apakah akan membentuk sirkuit?

Ternyata iya!

Maka e11 kita abaikan, dan kita beri tanda silang pada tabel.



Algoritma Kruskal



Langkah 13.

Isi kolom ketiga pada tabel berikut dan pastikan tidak ada sirkuit yang terbentuk.

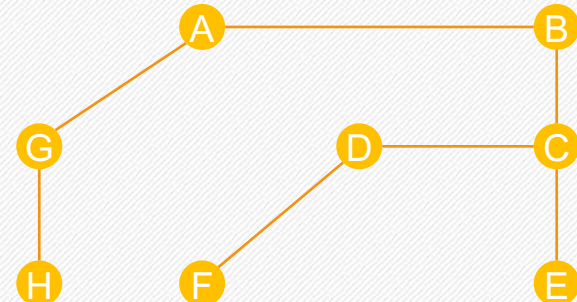
Sisi	Bobot	No sirkuit?
e4	3	✓
e7	4	✓
e2	5	✓
e8	5	✓
e9	5	✓
e1	15	✓
e3	15	✗
e5	15	✓
e10	15	✗
e11	15	✗
e6	18	✗

Sisi dengan bobot terkecil berikutnya adalah e6 (sisi terakhir) dengan bobot 18.

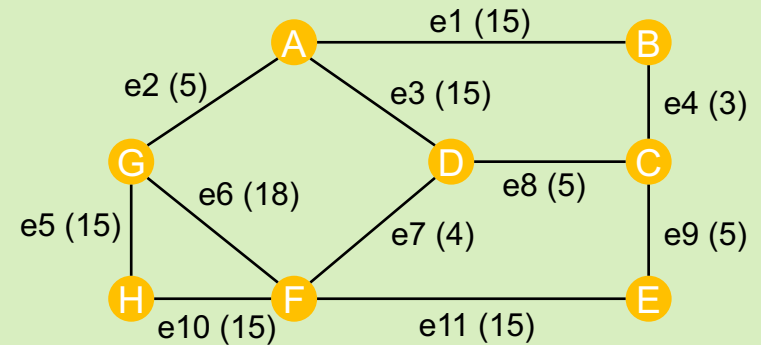
Kita cek e6, apakah akan membentuk sirkuit?

Ternyata iya!

Maka e6 kita abaikan, dan kita beri tanda silang pada tabel.



Algoritma Kruskal



Langkah 14.

Sampai di sini, algoritma kruskan sudah selesai diproses.

Sisi	Bobot	No sirkuit?
e4	3	✓
e7	4	✓
e2	5	✓
e8	5	✓
e9	5	✓
e1	15	✓
e3	15	x
e5	15	✓
e10	15	x
e11	15	x
e6	18	x

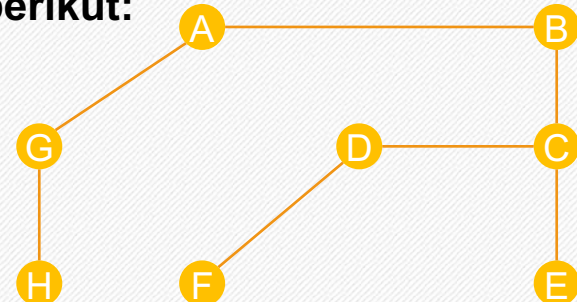
Langkah berikutnya adalah menghitung bobot-bobot yang terpilih, sesuai pada graf yang terbentuk.

Atau bisa juga menghitungnya dari tabel di samping. Jumlahkan semua bobot dengan tanda centang di sampingnya.

Totalnya adalah 52.

Dengan menggunakan algoritma Kruskal, maka total bobot minimum yang diperoleh adalah 52.

Sedangkan gambar pohon rentangnya adalah sebagai berikut:

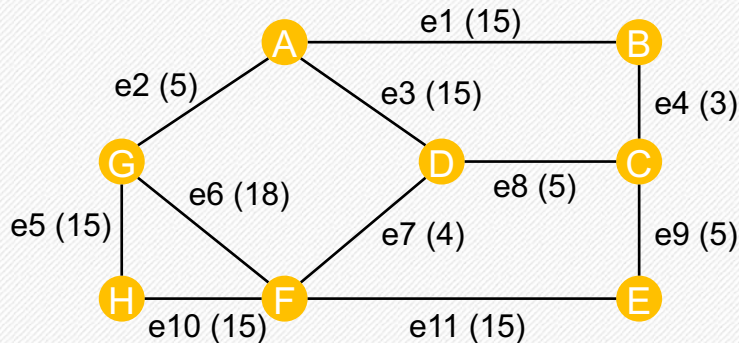


Algoritma Prim

Fauzi Adi Rafrastara

Algoritma Prim

Carilah bobot minimum graf berikut menggunakan algoritma Prim!

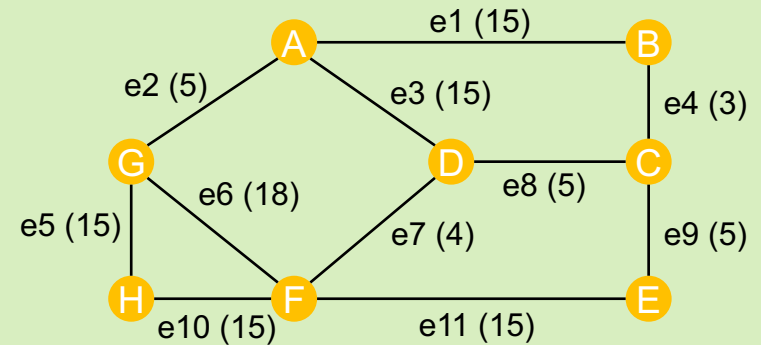


Langkah 1.

Gambar ulang grafnya, namun tanpa garis-garis sisi, seperti berikut ini.



Algoritma Prim



Langkah 2.

Buatlah tabel seperti berikut ini, lalu masukkan data-data sisi dan bobot yang ada:

Sisi	Bobot	No sirkuit?
e1	15	
e2	5	
e3	15	
e4	3	
e5	15	
e6	18	
e7	4	
e8	5	
e9	5	
e10	15	
e11	15	

Langkah 3.

Urutkan data di tabel mulai dari bobot terkecil!

Sisi	Bobot	No sirkuit?
e4	3	
e7	4	
e2	5	
e8	5	
e9	5	
e1	15	
e3	15	
e5	15	
e10	15	
e11	15	
e6	18	

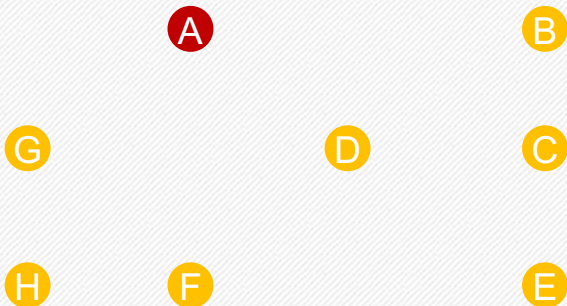
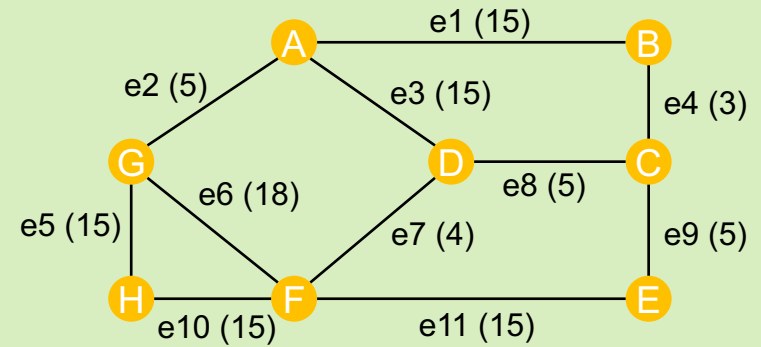
Algoritma Prim

Langkah 4.

Pilih salah satu node/titik untuk dijadikan sebagai titik awal (Start)!

Pemilihan titik awal ini bersifat bebas. Dalam latihan kali ini, kita memilih titik A sebagai titik permulaan.

Kita tandai A dengan warna merah.



Algoritma Prim

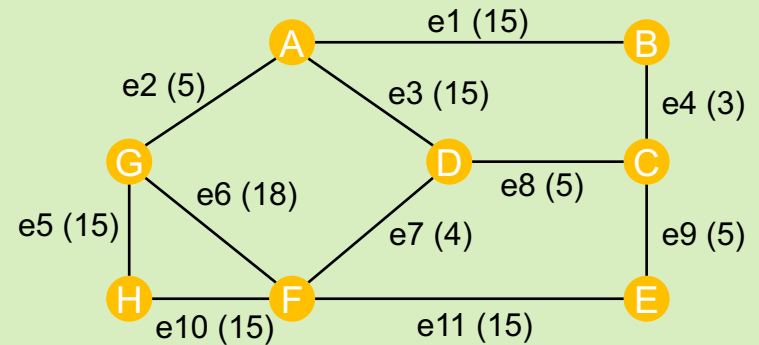
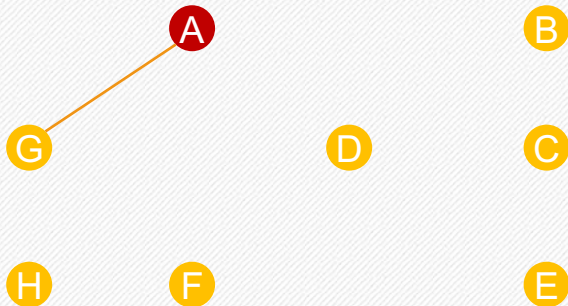
Langkah 5.

Perhatikan pada graf utuh di samping!

Tentukan titik/sisi yang terhubung dengan A, lalu tandai dan pilih yang bobotnya yang paling kecil!

Ada 3 sisi yang terhubung, yaitu e1, e2, dan e3. Dan sisi dengan bobot terkecil adalah e2.

Maka kita buat garis e2, yaitu dari A ke G.



Sisi	Bobot	Sisi terhubung	Sisi yang diambil
e4	3		
e7	4		
e2	5	↔	✓
e8	5		
e9	5		
e1	15	↔	
e3	15	↔	
e5	15		
e10	15		
e11	15		
e6	18		

Algoritma Prim

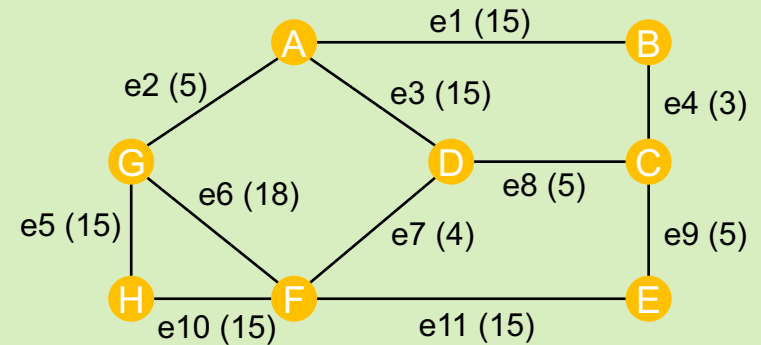
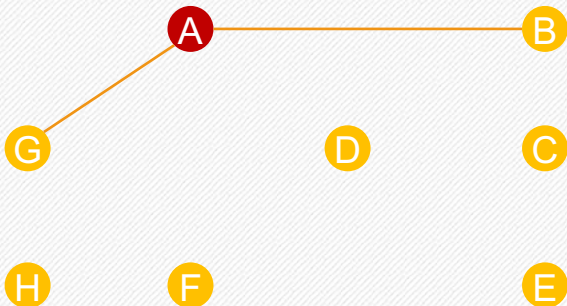
Langkah 6.

Perhatikan pada graf utuh di samping!

Tentukan titik/sisi yang terhubung dengan A atau G, lalu tandai dan pilih yang bobotnya yang paling kecil!

Ada 4 sisi yang terhubung, yaitu e1, e3, e5, dan e6.

Karena bobot terkecilnya adalah 15 dan dimiliki oleh 3 sisi, kita pilih salah satu saja, yaitu e1. Selanjutnya kita buat garis e1 (A - B)



Sisi	Bobot	Sisi terhubung	Sisi yang diambil
e4	3		
e7	4		
e2	5	✓	✓
e8	5		
e9	5		
e1	15	⇐	✓
e3	15	⇐	
e5	15	⇐	
e10	15		
e11	15		
e6	18	⇐	

Algoritma Prim

Langkah 7.

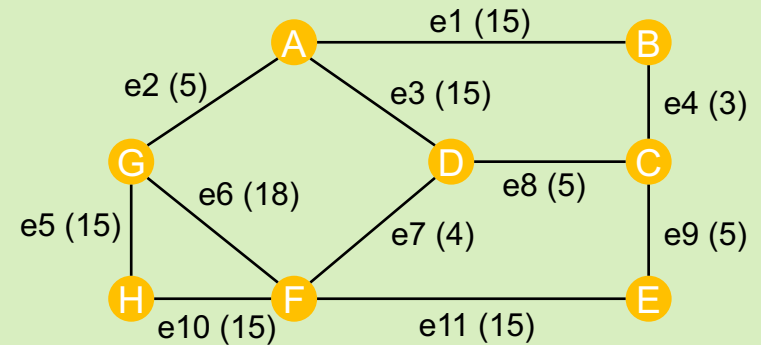
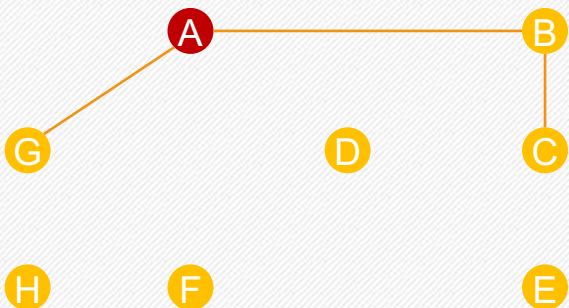
Perhatikan pada graf utuh di samping!

Tentukan titik/sisi yang terhubung dengan A, B atau G, lalu tandai dan pilih yang bobotnya yang paling kecil!

Ada 4 sisi yang terhubung, yaitu e3, e4, e5, dan e6.

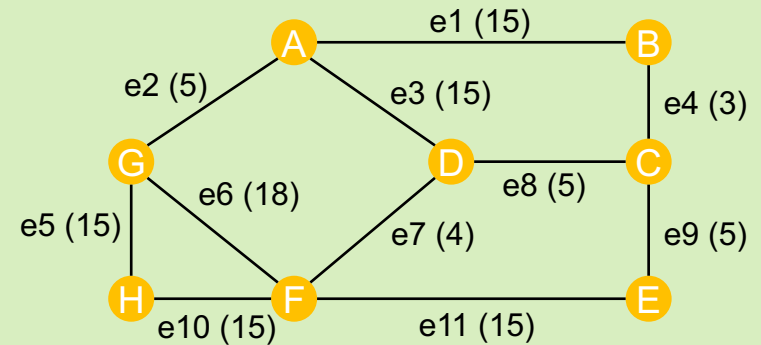
Bobot terkecilnya adalah 3, ada pada e4.

Maka kita gambar garis e4 (B – C)



Sisi	Bobot	Sisi terhubung	Sisi yang diambil
e4	3	↔	✓
e7	4		
e2	5	✓	✓
e8	5		
e9	5		
e1	15	✓	✓
e3	15	↔	
e5	15	↔	
e10	15		
e11	15		
e6	18	↔	

Algoritma Prim



Langkah 8.

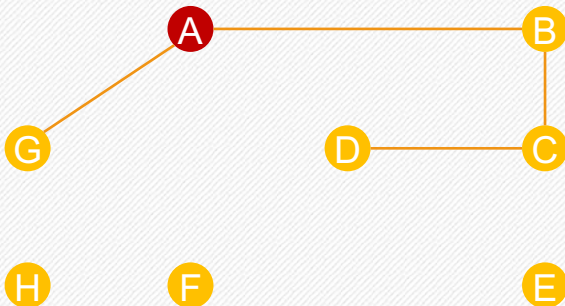
Tentukan titik/sisi yang terhubung dengan A, C atau G, lalu tandai dan pilih yang bobotnya yang paling kecil!

Titik B tidak lagi dicari karena sudah habis terpakai.

Ada 5 sisi yang terhubung, yaitu e3, e5, e6, e8 dan e9.

Bobot terkecilnya adalah 5, kita pilih yang e8.

Maka kita gambar garis e8 (C – D)



Sisi	Bobot	Sisi terhubung	Sisi yang diambil
e4	3	✓	✓
e7	4		
e2	5	✓	✓
e8	5	⇐	✓
e9	5	⇐	
e1	15	✓	✓
e3	15	⇐	
e5	15	⇐	
e10	15		
e11	15		
e6	18	⇐	

Algoritma Prim

Langkah 9.

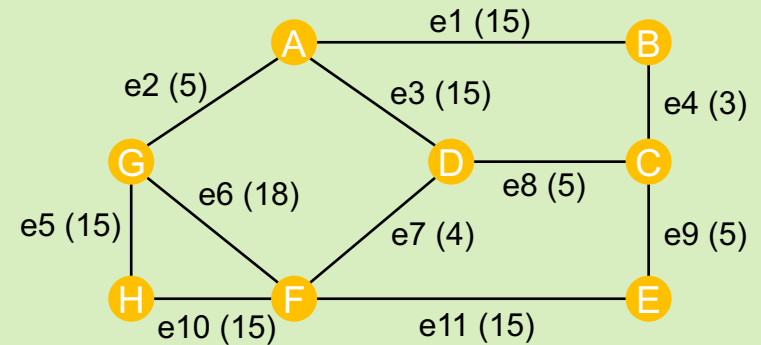
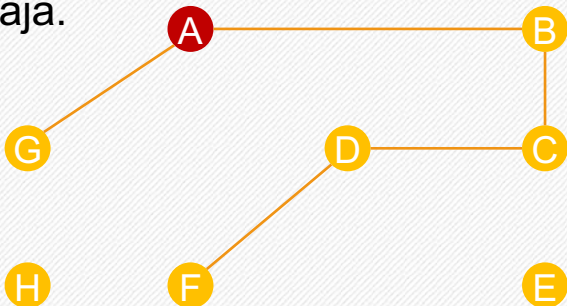
Tentukan titik/sisi yang terhubung dengan A, C, D atau G, lalu tandai dan pilih yang bobotnya yang paling kecil!

Ada 5 sisi yang terhubung, yaitu e3, e5, e6, e7 dan e9.

Bobot terkecilnya adalah 4, ada pada e7.

Maka kita gambar garis e7 (D – F)

Note: e3 tidak bisa dipakai karena akan membentuk sirkuit, sehingga kita abaikan saja.



Sisi	Bobot	Sisi terhubung	Sisi yang diambil
e4	3	✓	✓
e7	4	⇐	✓
e2	5	✓	✓
e8	5	✓	✓
e9	5	⇐	
e1	15	✓	✓
e3	15	✗	
e5	15	⇐	
e10	15		
e11	15		
e6	18	⇐	

Algoritma Prim

Langkah 10.

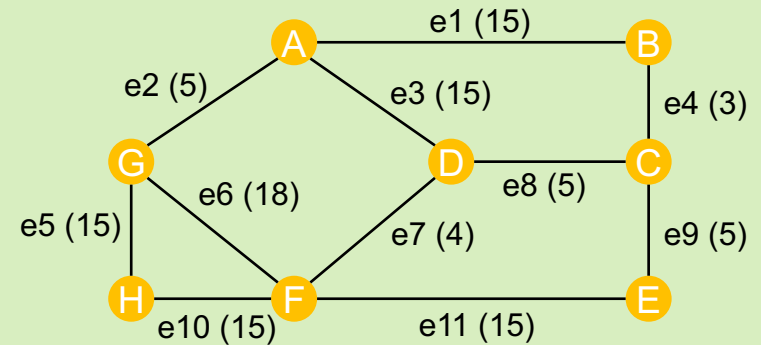
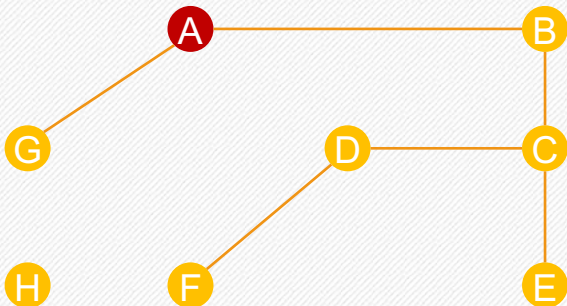
Tentukan titik/sisi yang terhubung dengan A, C, D atau G, lalu tandai dan pilih yang bobotnya yang paling kecil!

Ada 5 sisi yang terhubung, yaitu e5, e6, e9, e10 dan e11.

Sisi e6 tidak bisa dipakai karena akan membentuk sirkuit. Kita abaikan!

Bobot terkecilnya adalah 5, ada pada e9.

Maka kita gambar garis e7 (D – F)



Sisi	Bobot	Sisi terhubung	Sisi yang diambil
e4	3	✓	✓
e7	4	✓	✓
e2	5	✓	✓
e8	5	✓	✓
e9	5	⇐	✓
e1	15	✓	✓
e3	15	✗	
e5	15	⇐	
e10	15	⇐	
e11	15	⇐	
e6	18	✗	

Algoritma Prim

Langkah 11.

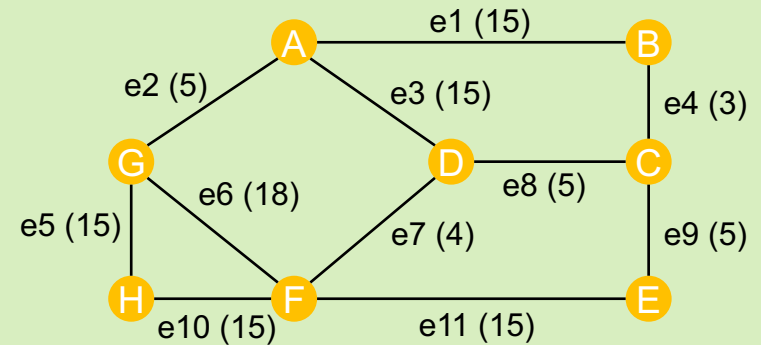
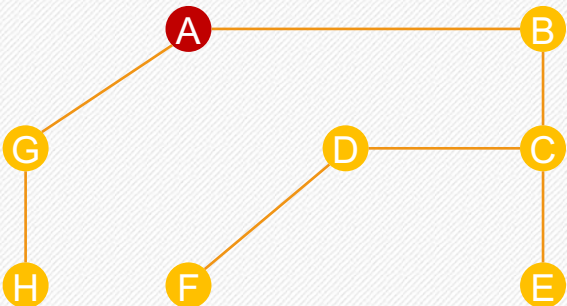
Sekarang, hanya ada dua titik saja yang memungkinkan tidak terbentuk sirkuit, yaitu titik F dan G.

Ada 2 sisi yang terhubung, yaitu e5, e10.

Sisi e11 tidak bisa dipakai karena sirkuit.

Bobot kedua sisi itu sama-sama 15, sehingga kita pilih salah satu, yaitu e5.

Maka kita gambar garis e5 (G – H)



Sisi	Bobot	Sisi terhubung	Sisi yang diambil
e4	3	✓	✓
e7	4	✓	✓
e2	5	✓	✓
e8	5	✓	✓
e9	5	✓	✓
e1	15	✓	✓
e3	15	✗	
e5	15	⇐	✓
e10	15	⇐	
e11	15	✗	
e6	18	✗	

Algoritma Prim

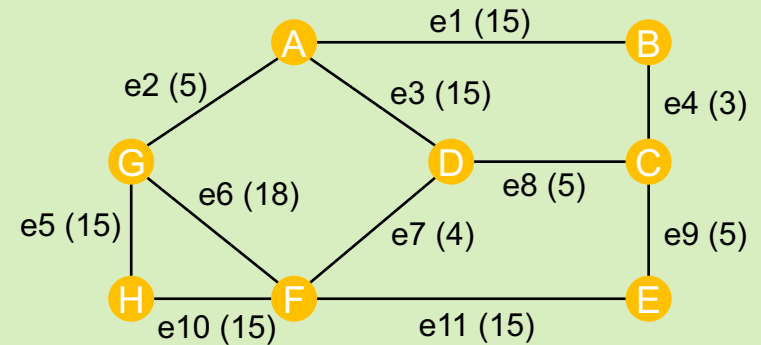
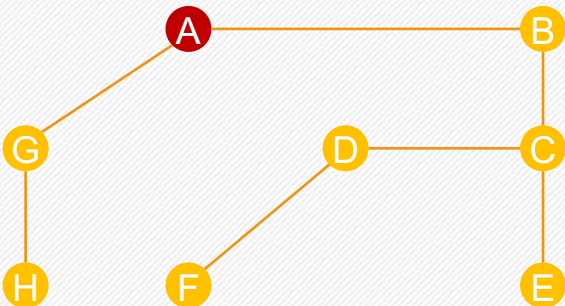
Langkah 12.

Sampai di sini, sudah tidak ada lagi titik yang bisa dihubungkan tanpa membentuk sirkuit.

Dengan demikian, maka proses kalkulasi dengan algoritma Prim sudah selesai.

Bentuk pohon rentangnya tampak seperti di bawah.

Sedangkan total bobot minimumnya adalah **52**



Sisi	Bobot	Sisi terhubung	Sisi yang diambil
e4	3	✓	✓
e7	4	✓	✓
e2	5	✓	✓
e8	5	✓	✓
e9	5	✓	✓
e1	15	✓	✓
e3	15	✗	
e5	15	✓	✓
e10	15	✗	
e11	15	✗	
e6	18	✗	