TEORI DASAR GRAF 1

Obyektif :

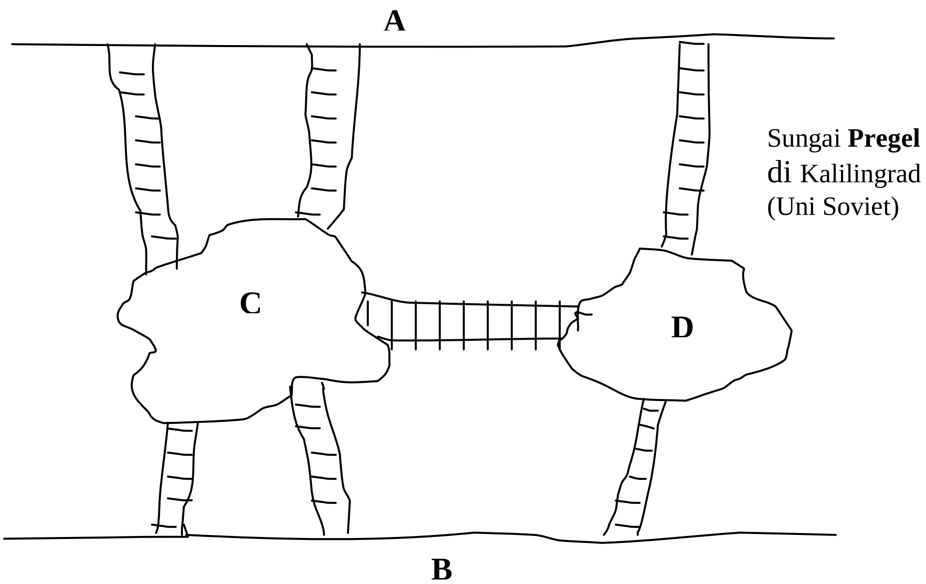
1. Mengerti apa yang dimaksud dengan Graf

2. Memahami operasi yang dilakukan pada Graf

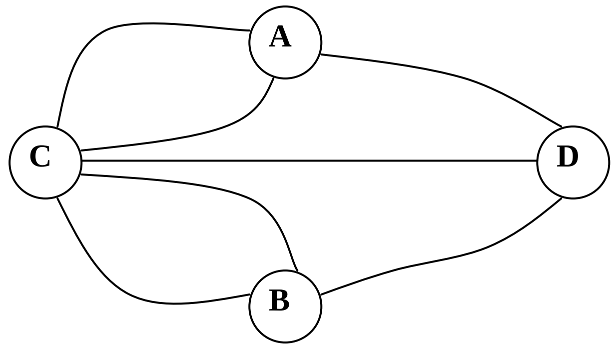
3. Mengerti derajat dan keterhubungan Graf

Teori Graf

Teori Graf mulai dikenal pada saat seorang matematikawan bangsa Swiss, bernama Leonhard Euler, berhasil mengungkapkan *Misteri Jembatan Konigsberg* pada tahun 1736. Di Kota Konigsberg (sekarang bernama Kalilingrad, di Uni Soviet) mengalir sebuah sungai bernama sungai Pregel. Di tengah sungai tersebut terdapat dua buah pulau. Dari kedua pulau tersebut terdapat jembatan yang menghubungi ke tepian sungai dan diantara kedua pulau. Jumlah jembatan tersebut adalah 7 buah seperti gambar berikut :



Secara singkat, dalam tulisannya, Euler menyajikan keadaan jembatan Konigsberg tersebut seperti gambar berikut :



Dalam masalah di atas, daratan (tepian A dan B, serta pulau C dan D) disajikan sebagai titik dan jembatan disajikan sebagai ruas garis. Euler mengemukakan teoremanya yang mengatakan bahwa perjalanan yang diinginkan di atas (yang kemudian dikenal sebagai perjalanan Euler) akan ada apabila *graf terhubung* dan banyaknya garis yang datang pada setiap titik (*derajat simpul*) adalah genap.

Problema & Model Graf

Secara umum, langkah-langkah yang perlu dilalui dalam penyelesaian suatu masalah dengan bantuan komputer adalah sebagai berikut :

Problema → Model Yang Tepat → Algoritma → Program Komputer

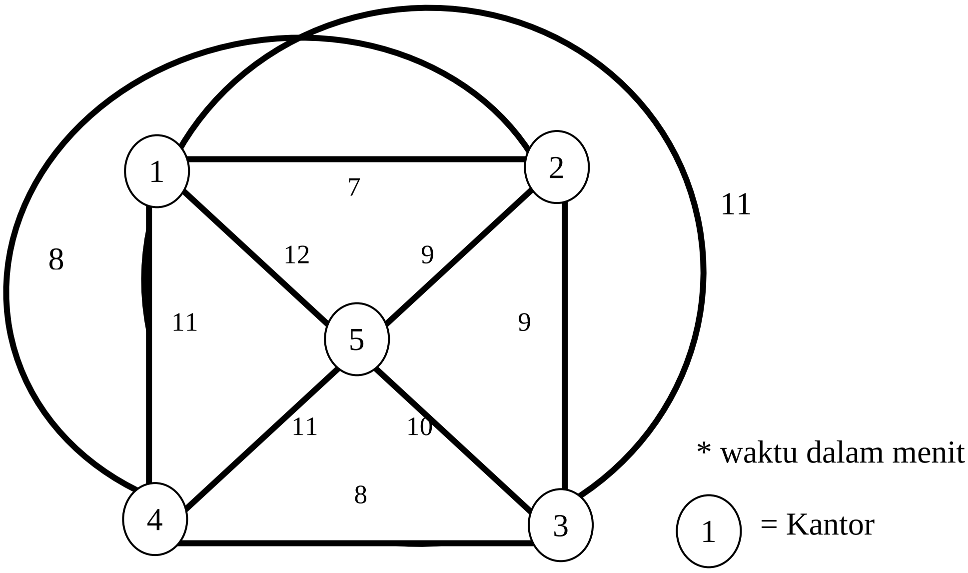
Contoh problema graf :

1. Petugas kantor telepon yang ingin mengumpulkan koin-koin dari telepon umum. Berangkat dari kantor & kembali ke kantornya lagi.

   Yang diharapkan → suatu <u>rute</u> perjalanan dengan <u>waktu minimal</u>.

   Masalah di atas dikenal sebagai *Travelling Salesman Problem*

   Sebagai contoh :

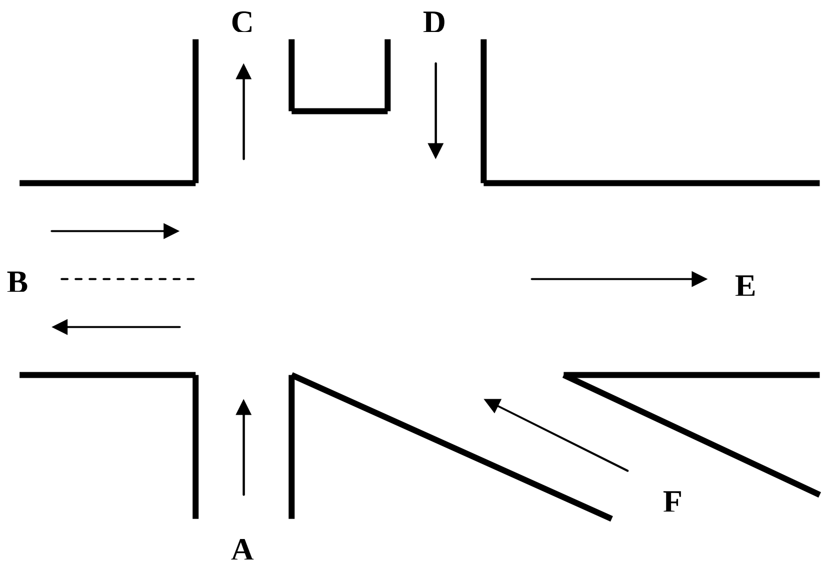


Untuk menyelesaikan masalah di atas dapat dipakai Algoritma Tetangga Terdekat (yakni menggunakan Metode Greedy)

2. Perancangan Lampu Lalu Lintas.

Yang diharapkan $\rightarrow$ pola lampu lalu lintas dengan <u>jumlah fase minimal.</u>

Sebagai contoh :

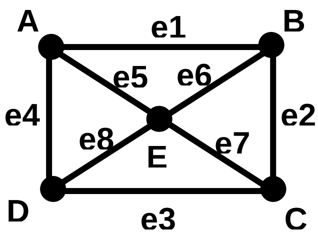


A $\rightarrow$ B			F $\rightarrow$ B
A $\rightarrow$ C	B $\rightarrow$ C	D $\rightarrow$ C	F $\rightarrow$ C
A $\rightarrow$ E	B $\rightarrow$ E	D $\rightarrow$ E	F $\rightarrow$ E

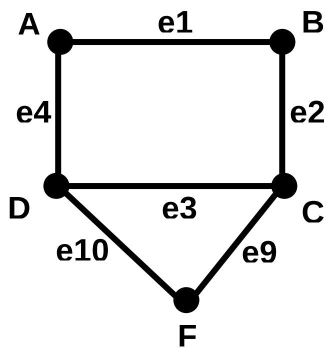
Untuk menyelesaikan masalah di atas dapat dipakai Algoritma Pewarnaan Graf (juga dikenal sebagai Graph Coloring, yakni menggunakan Metode Greedy)

Contoh :

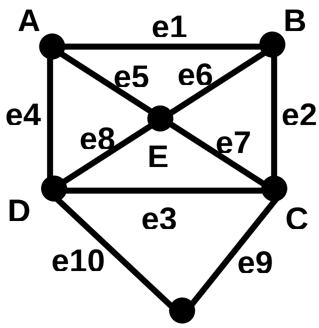
G_1



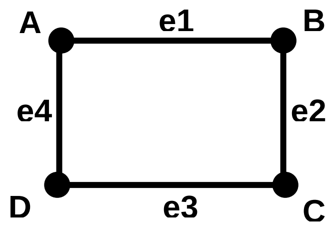
G_2



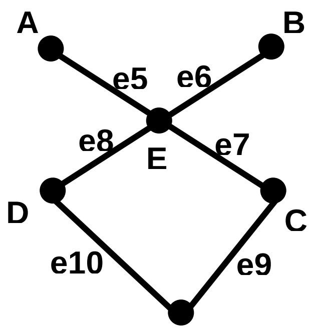
$G_1 \cup G_2$



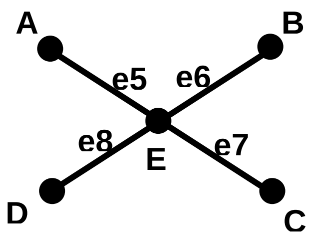
$G_1 \cap G_2$



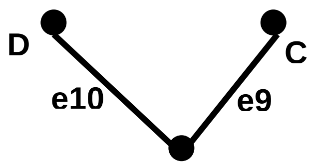
$G_1 \oplus G_2$



$G_1 - G_2$



$G_2 - G_1$

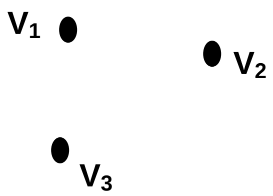


Graf Null / Hampa

Ada beberapa pengertian tentang graf null/hampa. Di sini akan dipakai pengertian bahwa suatu graf dikatakan graf null/hampa bila graf tersebut tidak mengandung ruas.

Contoh :

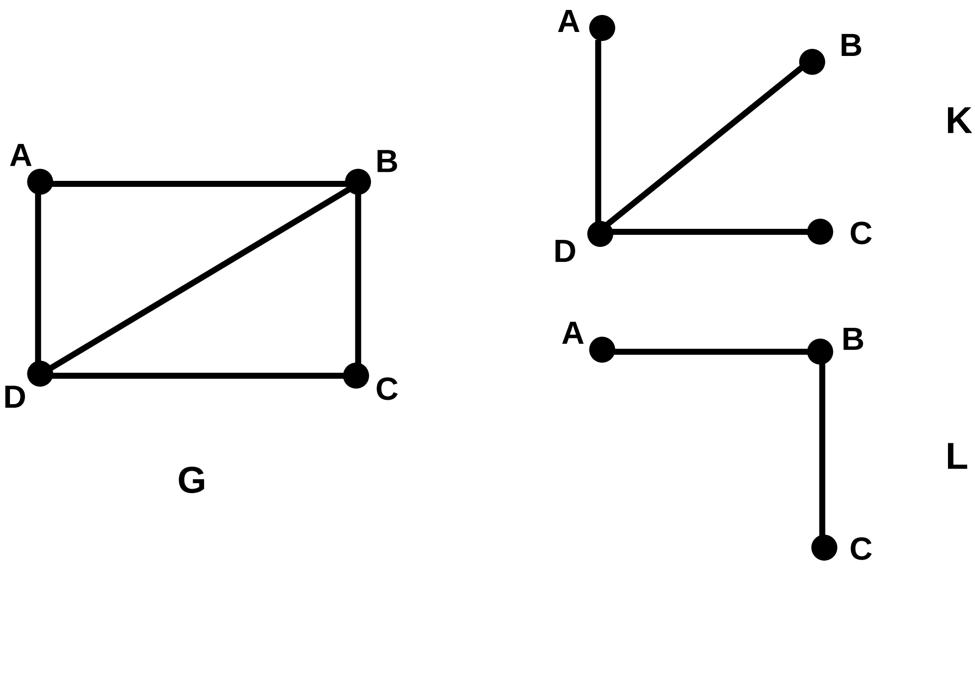
G :



$V \neq \varnothing$ dan
$E = \varnothing$

Suatu graf G dikatakan dikomposisikan menjadi K dan L bila G = $K \cup L$ dan $K \cap L = \varnothing$

Contoh :



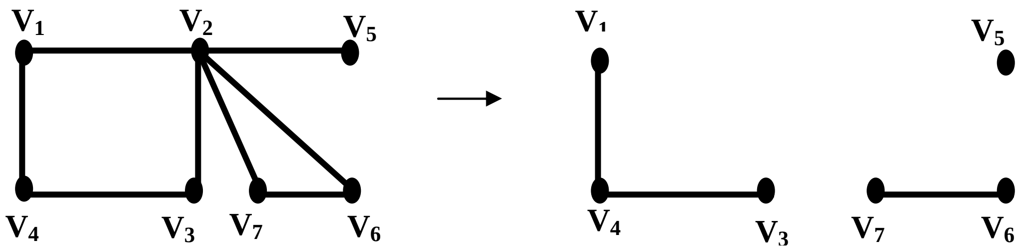
Penghapusan / Deletion

Penghapusan dapat dilakukan pada simpul ataupun ruas.

1) Penghapusan Simpul .

Notasinya : G – {V}

Contoh :

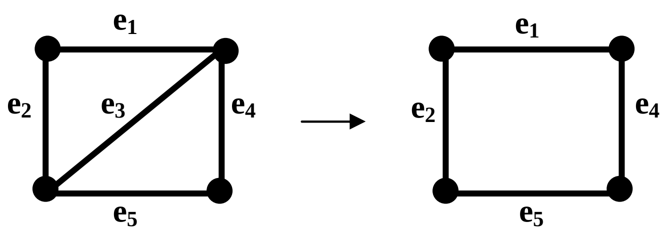


Penghapusan Simpul V_2

2) Penghapusan Ruas .

Notasinya : G – {e}

Contoh :

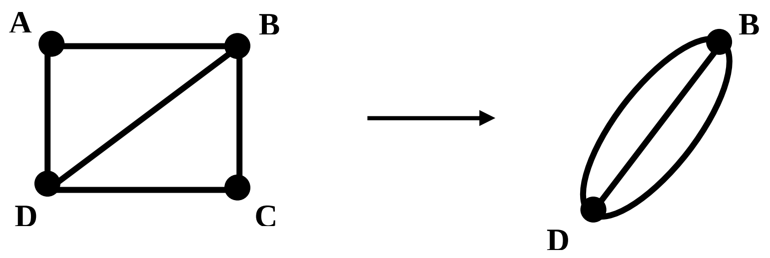


Penghapusan Ruas e_3

Pemendekan / Shorting

Pemendekan/Shorting adalah menghapus simpul yang dihubungkan oleh 2 ruas (simpul berderajat 2), lalu menghubungkan titik-titik ujung yang lain dari kedua ruas tersebut.

Contoh :

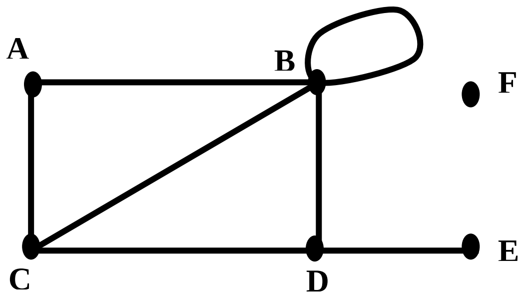


pemendekan terhadap simpul A dan C

Derajat Graf

Derajat graf adalah jumlah dari derajat simpul-simpulnya. Sedangkan derajat simpul adalah banyaknya ruas yang incidence (terhubung) ke simpul tersebut.

Contoh :



d (A) = 2

d (B) = 5

d (C) = 3

d (D) = 3

d (E) = 1

d (F) = 0

———————— +

Σ = 14 = 2 x Size

Berdasarkan derajat simpul, sebuah simpul dapat disebut :

- Simpul Ganjil, bila derajat simpulnya merupakan bilangan ganjil

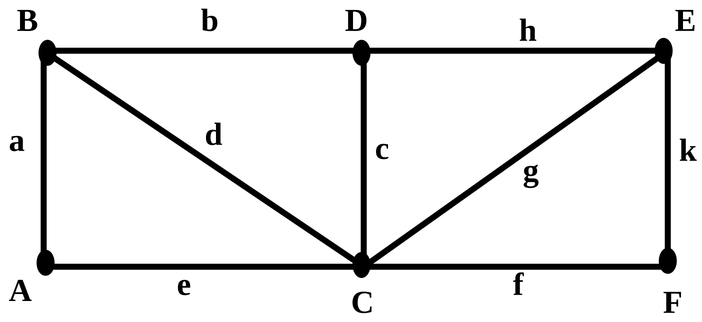
- Simpul Genap, bila derajat simpulnya merupakan bilangan genap
- Simpul Bergantung / Akhir, bila derajat simpulnya adalah 1
- Simpul Terpencil, bila derajat simpulnya adalah 0

Keterhubungan

Dalam keterhubungan sebuah graf, akan dikenal beberapa istilah-istilah berikut :

1. Walk : barisan simpul dan ruas
2. Trail : Walk dengan ruas yang berbeda
3. Path / Jalur : Walk dengan simpul yang berbeda
4. Cycle / Sirkuit : Trail tertutup dengan derajat setiap simpul = 2

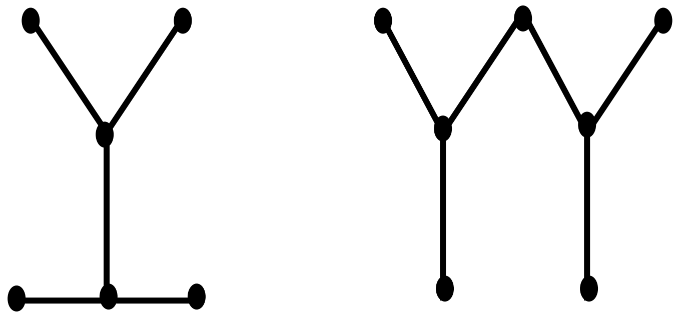
Contoh :



1) A, B, C, D, E, F, C, A, B, D, C → Walk
2) A, B, C, D, E, F, C, A → Trail
3) A, B, C, A → Cycle
4) A, B, D, C, B, D, E → Walk
5) A, B, C, D, E, C, F → Trail
6) A, B, D, C, E, D → Trail
7) A, B, D, E, F, C, A → Cycle
8) C, E, F → Path
9) B, D, C, B → Cycle
10) C, A, B, C, D, E, C, F, E → Trail
11) A, B, C, E, F, C, A → Trail

Graf yang tidak mengandung cycle disebut dengan Acyclic

Contoh :



Suatu graf G disebut terhubung jika untuk setiap 2 simpul dari graf terdapat jalur yang menghubungkan kedua simpul tersebut.

Subgraf terhubung suatu graf disebut komponen dari G bila subgraf tersebut tidak terkandung dalam subgraf terhubung lain yang lebih besar.

Jarak antara 2 simpul dalam graf G adalah panjang jalur terpendek antara ke-2 simpul tersebut.

Diameter suatu graf terhubung G adalah maksimum jarak antara simpul-simpul G.

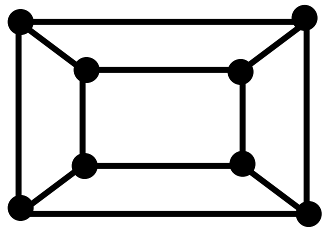
Ada **Subgraf S** dari graf terhubung G, yang bila kita ambil / pindahkan dari G, akan menyebabkan **G tidak terhubung** .

Kalau tidak ada **Subgraf sejati R** dari S, yang pemindahannya juga menyebabkan G tidak terhubung, maka **S** disebut **Cut-Set** dari G.

Graf Regular

Sebuah graf dikatakan graf regular bila derajat setiap simpulnya sama.

Contoh :



GRAF TIDAK BERARAH

Obyektif :

4. Mengerti apa yang dimaksud dengan Graf tidak berarah

5. Mengerti mengenai Graf Berlabel

Graf Secara Formal

Sebuah Graf G mengandung 2 himpunan :

(1). Himp. V, yang elemennya disebut simpul

→ **V**ertex / point / titik / node

(2). Himp. E, yang merupakan pasangan tak terurut dari simpul-simpul, disebut ruas

→ **E**dge / rusuk / sisi

Sehingga sebuah graf dinotasikan sebagai G (V, E)

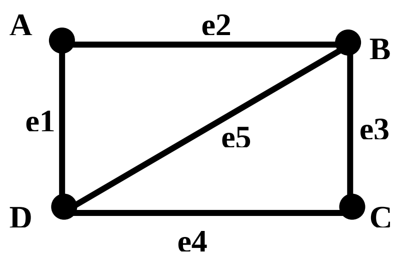
Contoh :

G (V, E)

V = { A, B, C, D }

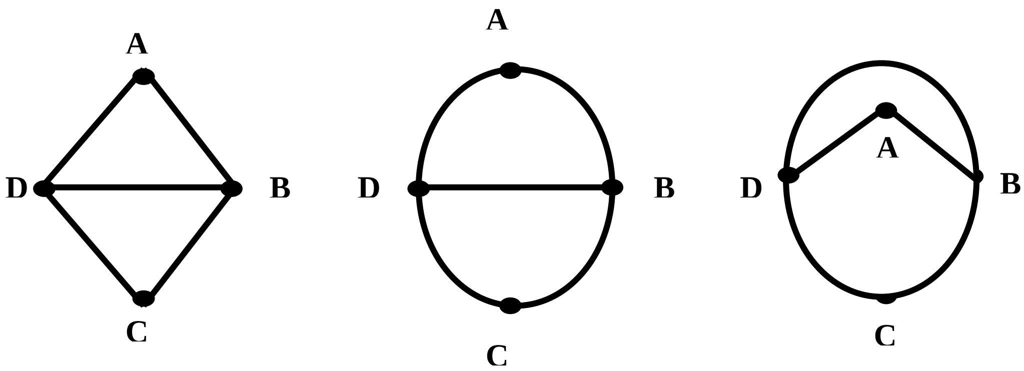
E = { (A, B), (B, C), (C, D), (D, A), (B, D) }

Secara Geometri :



→ terdiri dari 4 simpul dan 5 ruas

Tidak ada ketentuan khusus dalam penyajian graf secara geometri, seperti dimana dan bagaimana menyajikan simpul dan ruas. Berikut contoh penyajian Graf yang sama, tetapi disajikan berbeda.



Beberapa istilah lain dalam graf :

- **Berdampingan**

  simpul U dan V disebut berdampingan bila terdapat ruas (U,V)
- **Order**

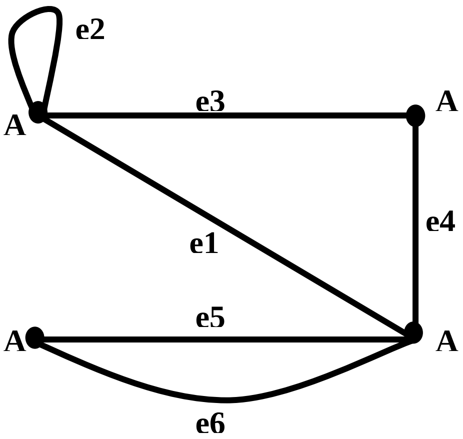
  banyaknya simpul
- **Size**

  banyaknya ruas
- **Self-loop (loop) / Gelung**

  ruas yang menghubungkan simpul yang sama (sebuah simpul)
- **Ruas sejajar / berganda**

  ruas-ruas yang menghubungkan 2 simpul yang sama

Sebuah graf dikatakan multigraf bila graf tersebut mengandung ruas sejajar atau gelung. Sedangkan graf yang tidak mengandung ruas sejajar atau gelung dikenal sebagai graf sederhana, atau yang disebut graf. Adapun contoh multigraf adalah sebagai berikut.



Multigraf

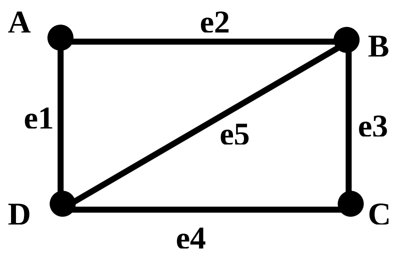
Subgraf

G‘(V‘, E‘) adalah Subgraf dari G (V, E) bila : V‘ $\subset$ V dan E‘ $\subset$ E

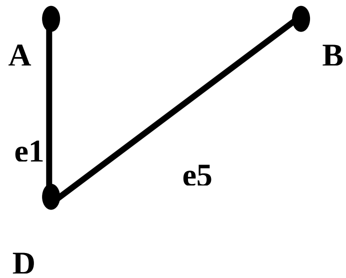
Apabila <u>E</u>‘ mengandung semua ruas di <u>E</u> yang kedua ujungnya di <u>V</u>‘ , maka <u>G</u>‘ adalah Subgraf yang dibentuk oleh <u>V</u>‘ (**Spanning Subgraph)**

Contoh :

G :

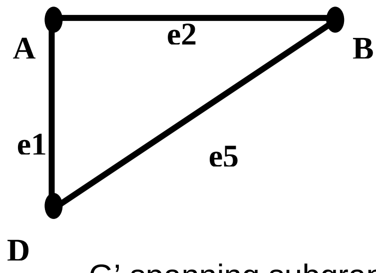


G' :



G' subgraf dari G

G' :

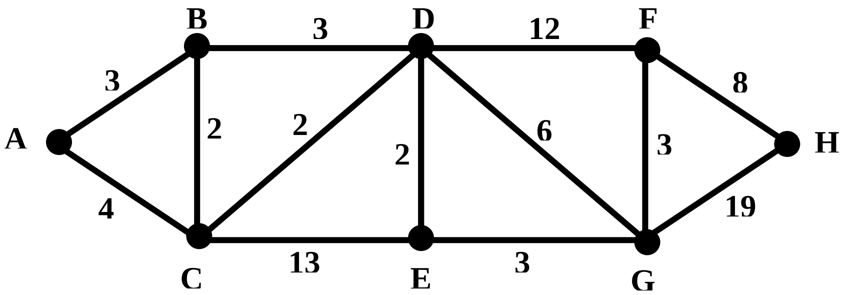


G' spanning subgrapf dari G

Graf berlabel

Graf berlabel/ berbobot adalah graf yang setiap ruasnya mempunyai nilai/bobot berupa bilangan non negatif.

Contoh :



Isomorfisma

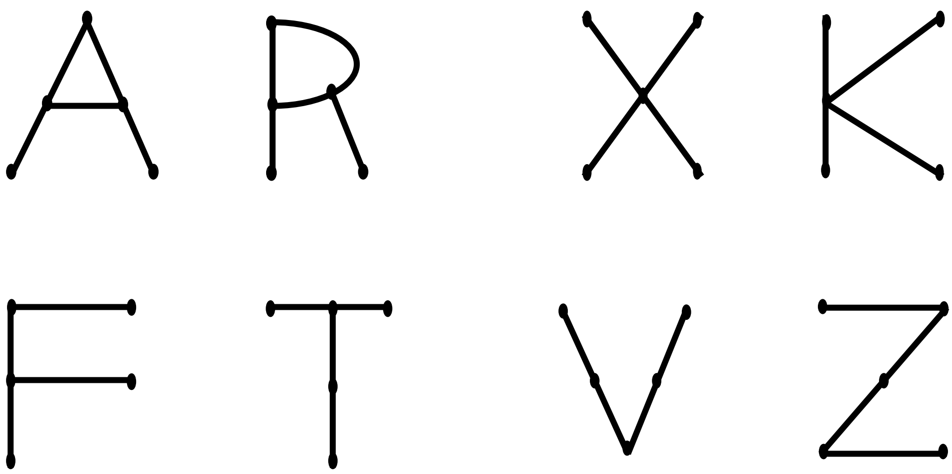
G (V,E) dan G* (V*,E*) adalah 2 buah Graf.

$f : V \rightarrow V^*$ suatu fungsi satu-satu dan pada, sedemikian sehingga (u,v) adalah ruas dari G jika dan hanya jika (f (u),f(v)) adalah ruas dari G *

Maka f disebut fungsi yang *isomorfisma* dan G & G * adalah graf-graf yang *isomorfis*

Contoh :

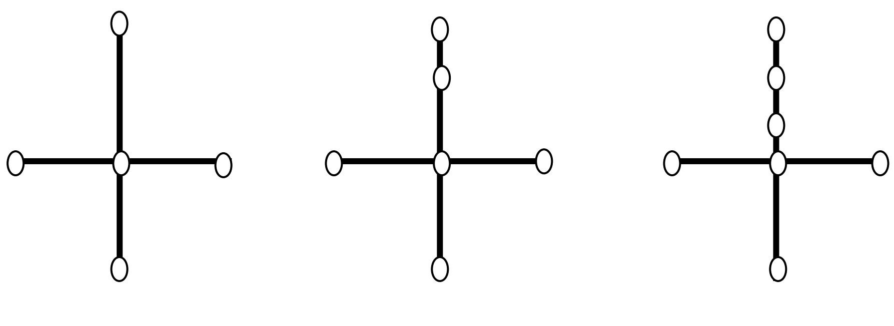
Graf yang berbentuk huruf A & R, X & K, F & T, dan V & Z, di bawah ini adalah isomorfis.



Homomorfis

Jika G* dan G** diperoleh dari G dengan membagi beberapa ruas dari G oleh penambahan beberapa simpul pada ruas tersebut, maka kedua graf G* dan G** disebut *homomorfis*

Contoh :



G **G*** **G****

Operasi pada Graf

Berdasarkan definisi graf (yang terdiri dari 2 himpunan) dan operasi pada himpunan, maka pada graf juga dapat dilakukan operasi-operasi. Bila diketahui 2 buah graf : $G_1(V_1,E_1)$ dan $G_2(V_2,E_2)$, maka :

1. Gabungan $G_1 \cup G_2$ adalah graf dengan himpunan V nya = $V_1 \cup V_2$ dan himpunan E nya = $E_1 \cup E_2$
2. Irisan $G_1 \cap G_2$ adalah graf dengan himpunan V nya = $V_1 \cap V_2$ dan himpunan E nya = $E_1 \cap E_2$
3. Selisih $G_1 - G_2$ adalah graf dengan himpunan V nya = V_1 dan himpunan E nya = $E_1 - E_2$

   Sedangkan Selisih $G_2 – G_1$ adalah graf dengan himpunan V nya = V_2 dan himpunan E nya = $E_2 – E_1$
4. Penjumlahan Ring $G_1 \oplus G_2$ adalah graf yang dihasilkan dari

   $(G_1 \cup G_2) – (G_1 \cap G_2)$ atau $(G_1 - G_2) \cup (G_2 - G_1)$

Graf Planar

Obyektif :

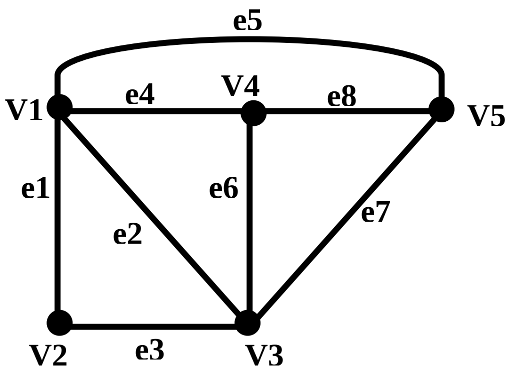
6. **Mengerti penyajian Graf dalam bentuk Matriks**
7. **Mengerti apa yang dimaksud dengan Graf Planar**
8. **Mengerti apa yang dimaksud dengan Graf Non Planar**
9. **Memahami Teorema Kuratowski**

Matriks dan Graf

Graf dapat disajikan dalam bentuk matriks. Matriks-matriks yang dapat menyajikan model graf tersebut antara lain :

- Matriks Ruas
- Matriks Adjacency
- Matriks Incidence

Sebagai contoh, untuk graf seperti di bawah ini :



Maka,

Matriks Ruas :

$$\underset{n \times 2}{\begin{bmatrix} 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \\ 2 & 3 \\ 3 & 4 \\ 3 & 5 \\ 4 & 5 \end{bmatrix}}$$

Atau :

$$
\begin{array}{c}
2 \times n \\
\begin{bmatrix}
1 & 1 & 1 & 1 & 2 & 3 & 3 & 4 \\
2 & 3 & 4 & 5 & 3 & 4 & 5 & 5
\end{bmatrix}
\end{array}
$$

Matriks Adjacency :

$$
\begin{array}{c c}
 & \begin{array}{ccccc} V1 & V2 & V3 & V4 & V5 \end{array} \\
\begin{array}{c} V1 \\ V2 \\ V3 \\ V4 \\ V5 \end{array} &
\begin{bmatrix}
0 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 1 \\
1 & 0 & 1 & 1 & 0
\end{bmatrix}
\end{array}
$$

Matriks Incidence :

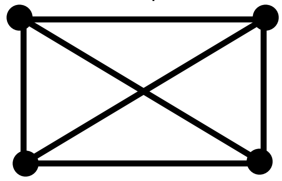
$$
\begin{array}{c c}
 & \begin{array}{cccccccc} e1 & e2 & e3 & e4 & e5 & e6 & e7 & e8 \end{array} \\
\begin{array}{c} V1 \\ V2 \\ V3 \\ V4 \\ V5 \end{array} &
\begin{bmatrix}
1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 1
\end{bmatrix}
\end{array}
$$

Graf Planar

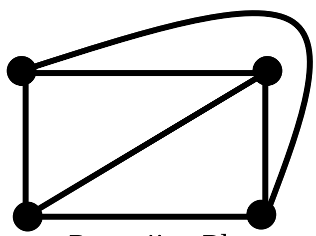
Sebuah graf dikatakan graf planar bila graf tersebut dapat disajikan (secara geometri) tanpa adanya ruas yang berpotongan. Sebuah graf yang disajikan tanpa adanya ruas yang berpotongan disebut dengan penyajian planar/map/peta.

Contoh :

K_4



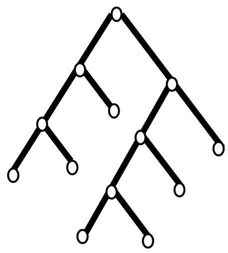
Graf Planar



Penyajian Planar

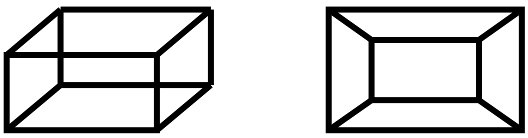
Graf yang termasuk planar antara lain :

- Tree / Pohon
- Kubus
- Bidang Empat
- Bidang Delapan Beraturan

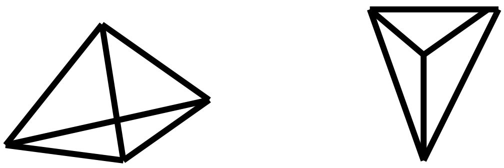


Tree / Pohon

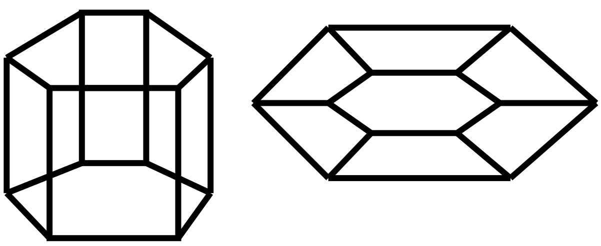
Kubus



Bidang Empat

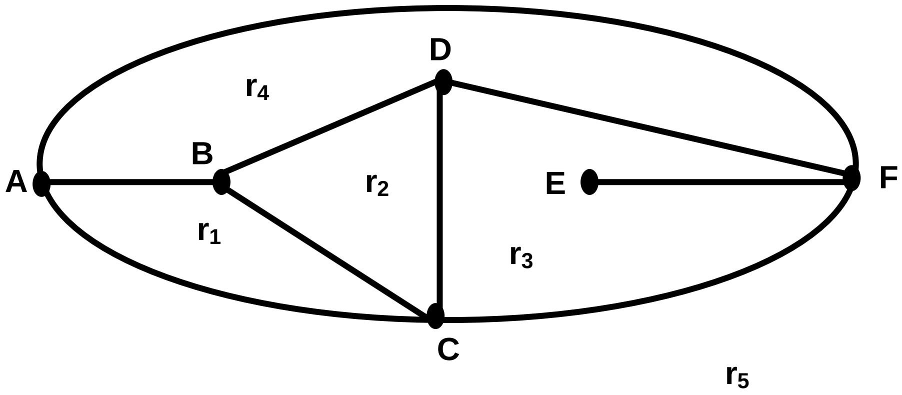


Bidang Delapan Beraturan



Pada penyajian planar/map, dikenal istilah region. Derajat dari suatu region adalah panjang walk batas region tersebut

Contoh :



d (r1) = 3

d (r2) = 3

d (r3) = 5

d (r4) = 4

d (r5) = 3

+

Σ = 18 = 2 x SIZE

Region dengan batasnya gelung, maka d (r) = 1

Region dengan batasnya ruas sejajar, maka d (r) = 2

Formula Euler untuk Graf Planar

Untuk Graf Planar berlaku Formula Euler berikut :

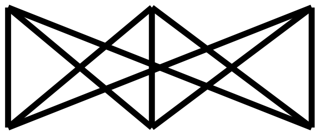
V – E + R = 2

Dimana p = jumlah simpul dan q = jumlah ruas

Graf Non-Planar

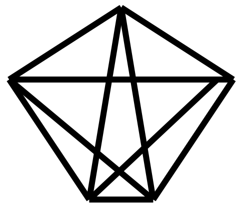
Sebuah graf yang tidak dapat disajikan (secara geometri) tanpa adanya ruas yang berpotongan dikenal sebagai graf non planar.

Contoh :



$K_{3,3}$

Utility Graph



K_5 = Bintang

Teorema Kuratowski (1930)

Suatu graf adalah Non-Planar jika dan hanya jika mengandung subgraf yang Homomorfis ke $K_{3,3}$ atau ke K_5

Pewarnaan pada Graf

Obyektif :

10. Memahami pewarnaan simpul graf

11. Memecahkan permodelan masalah pewarnaan simpul graf

Pewarnaan Graf

Pewarnaan graf adalah pemberian warna terhadap simpul-simpul graf dimana 2 buah simpul yang berdampingan tidak boleh mempunyai warna yang sama.

G berwarna n artinya graf tersebut menggunakan n warna.

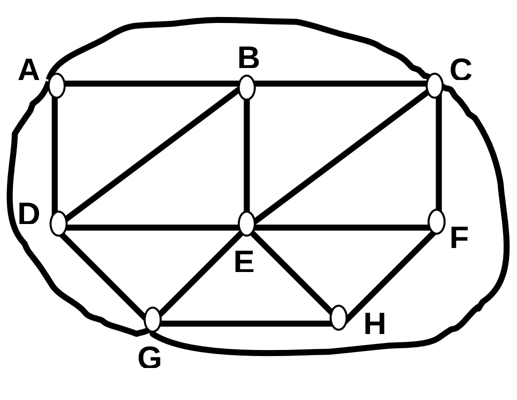
Bilangan kromatis dari G = K(G) adalah jumlah minimum warna yang dibutuhkan.

Algoritma yang dapat digunakan untuk mendapatkan bilangan kromatis dari sebuah graf adalah Algoritma Welch-Powell.

Adapun langkah-langkahnya adalah :

1. Urutkan simpul-simpul berdasarkan derajatnya.
   Dari besar ke kecil.
2. Warnai.

Contoh :



Langkah 1 :

urutan simpulnya dari besar ke kecil adalah : E, C, G, A, B, D, F, H

Langkah 2 :

mewarnai :

warna Merah : E, A

warna Putih : C, D, H

warna Biru : G, B, F

Sehingga bilangan kromatis graf di atas adalah 3.

Teorema :

Pernyataan berikut adalah ekivalen :

(1) G berwarna 2

(2) G adalah bipartisi

(3) Setiap sirkuit dalam G mempunyai panjang genap

Graf Lengkap k dengan n simpul membutuhkan n warna

Teorema :

Suatu graf planar G adalah berwarna 5

Pewarnaan Region

Pewarnaan region dapat dilakukan (seperti pemberian warna pada wilayah-wilayah di peta) dengan cara membuat dual dari map tersebut. Gambarkan sebuah simpul baru pada masing-masing region suatu map M, kemudian buat sebuah ruas yang menghubungkan simpul pada 2 buah region yang berdampingan bila terdapat ruas sebagai batas / persekutuan kedua region tersebut. Buatlah tanpa adanya ruas baru yang berpotongan, maka akan terbentuk suatu map M*, yang disebut dual dari map M.

Setelah Dualnya terbentuk, dapar dilakukan pewarnaan terhadap simpul-simpulnya. Simpul-simpul tersebut mewakili region sebelumnya, sehingga warna yang digunakan untuk suatu simpul berarti warna yang dapat digunakan untuk pewarnaan region yang diwakilinya.

Teorema : suatu map M adalah berwarna 5

Setiap graf planar adalah berwarna (simpul) 4

Dibuktikan oleh Apple & Haken (1976) – 2000 Graf, jutaan kasus.

POHON

Obyektif :

12. Mengerti apa yang dimaksud dengan Tree (Pohon)

13. Mengerti Pohon Rentangan

14. Mengerti Pohon Rentangan Minimal

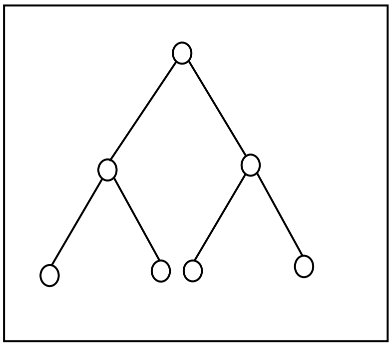
Pohon

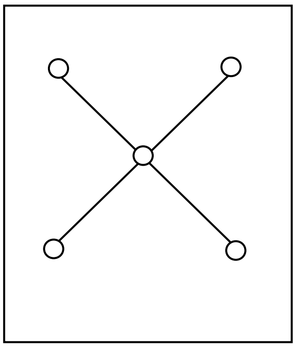
Tree atau pohon adalah graf terhubung yang tidak mengandung sirkuit.

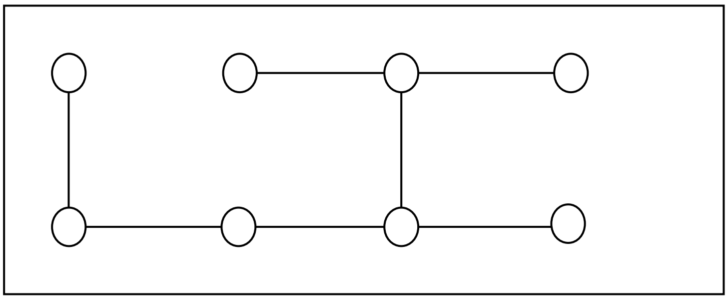
Untuk itu perlu diingat kembali bahwa :

- Suatu Graf G disebut terhubung apabila untuk setiap dua simpul dari graf G selalu terdapat jalur yang menghubungkan kedua simpul tersebut.
- Sirkuit atau cycle adalah suatu lintasan tertutup dengan derajat setiap simpul dua.

Contoh :







Sifat :

Suatu Graf G adalah Pohon jika dan hanya jika terdapat satu dan hanya satu jalur diantara setiap pasang simpul dari Graf G.

Teorema :

Suatu Graf G dengan n buah simpul adalah Pohon jika :

(1) G terhubung dan tak mengandung sirkuit, atau

(2) G tidak mengandung sirkuit dan mempunyai n-1 buah ruas, atau

(3) G mempunyai n-1 buah ruas dan terhubung

Pohon Rentangan

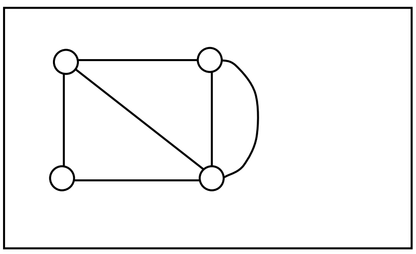
Suatu spanning tree atau pohon rentangan adalah suatu subgraf dari graf G yang mengandung semua simpul dari G, dan merupakan suatu pohon.

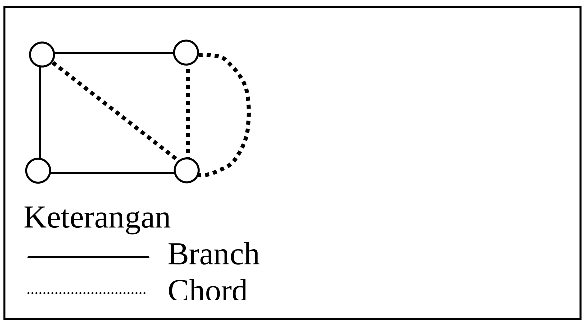
GRAF G	SPANNING TREE
n simpul	n simpul
m ruas	n – 1 ruas

n – 1 ruas → BRANCH (CABANG)

$m - (n - 1)$ → CHORD

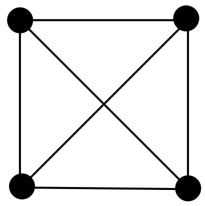
Contoh :



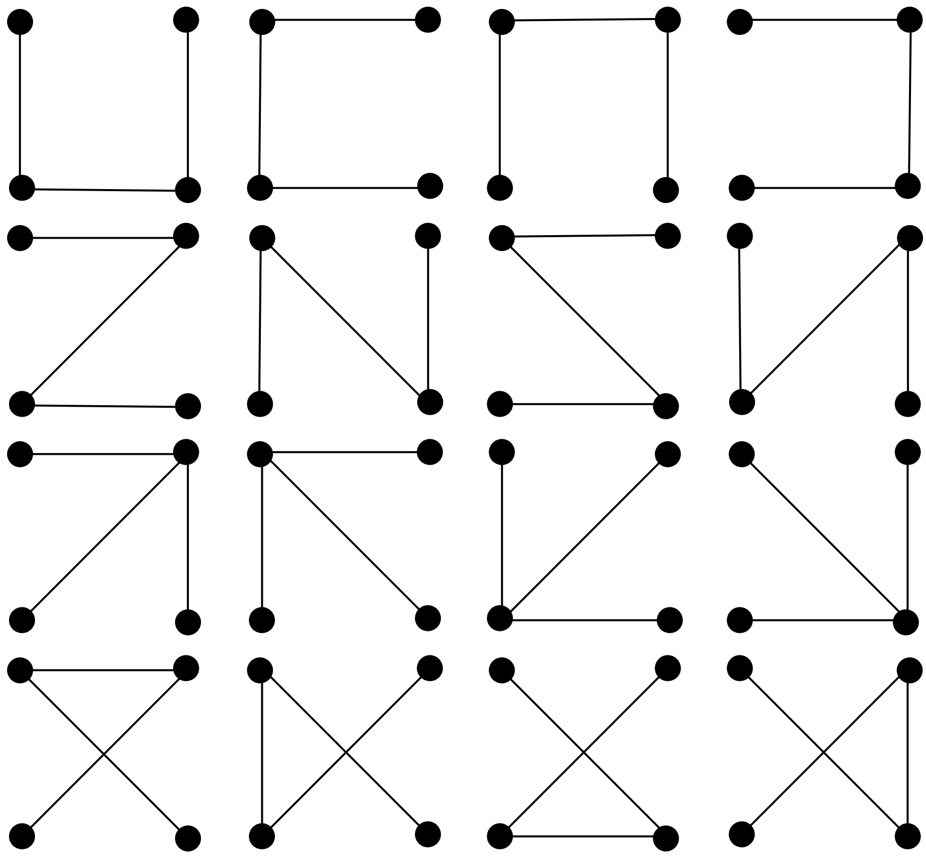


Contoh :

Graf G :



Pohon Rentangan dari Graf G :

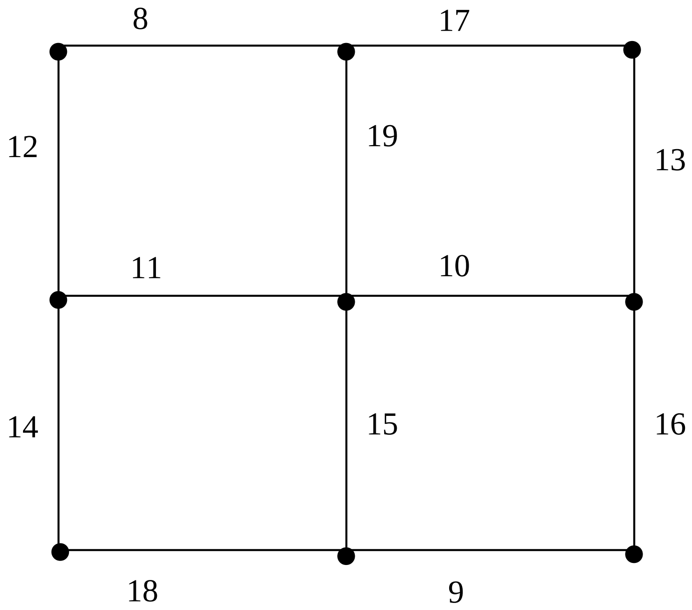


Pohon Rentangan Minimal

Apabila G suatu Graf berbobot (Suatu Network); maka pohon rentangan minimal dari graf adalah pohon rentangan dengan jumlah bobot terkecil.

$\Rightarrow$ Minimal spanning tree

Contoh :



Untuk mendapatkan pohon rentangan minimal dapat digunakan Algoritma berikut :

- Solin
- Kruskal
- Prim’s

SOLIN

1. Urutkan ruas dari G menurut bobotnya; dari besar ke kecil.
2. Lakukan penghapusan ruas berdasarkan urutan yang sudah dilakukan; dengan ketentuan bahwa penghapusan ruas tersebut tidak menyebabkan graf menjadi tidak terhubung.

KRUSKAL

1. Urutkan ruas dari G menurut bobotnya; dari kecil ke besar.
2. Lakukan penambahan ruas berdasarkan urutan yang sudah dilakukan; dengan ketentuan bahwa penambahan ruas tersebut tidak menyebabkan adanya sirkuit.

PRIM'S

= Kruskal + menjaga graf tetap terhubung

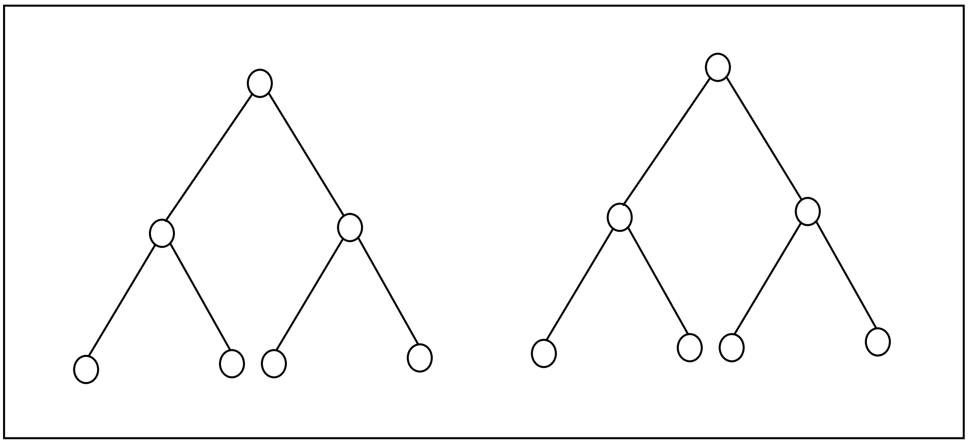
Untuk mencari pohon rentangan maksimal, dapat dilakukan dengan dengan cara merubah bobot tiap ruas menjadi – (bobot yang lama)

Definisi :

Hutan atau foresi adalah graf yang tidak mengandung sirkuit.

∴ Pohon adalah hutan yang terhubung

Contoh :



Graf Berarah

Obyektif :

15. Memahami konsep graf berarah

Suatu graf berarah (Direct graf disingkat Digraf) D terdiri atas 2 himpunan :
(1) Himpunan V, anggotanya disebut simpul.
(2) Himpunan A, merupakan himpunan pasangan terurut, yang disebut ruas berarah atau **arc**.
Graf berarah diatas, kita tulis sebagai D(V, A)
Apabila arc dan/atau simpul suatu graf berarah menyatakan suatu bobot, maka graf berarah tersebut dinamakan suatu jaringan atau Network

Beberapa definisi pada graf berarah :
Misalkan D suatu graf berarah.
Kita menyebut arc a = (u, v) adalah mulai pada titik awal u, dan berakhir pada titik terminal v.
Derajat keluar(out degree) suatu simpul adalah banyaknya arc yang mulai/keluar dari simpul tersebut.
Derajat kedalam (in degree) suatu simpul adalah banyaknya arc yang berakhir / masuk ke simpul tersebut.
Sumber (source) adalah simpul yang mempunyai derajat kedalam = 0.
Sink (muara) adalah simpul yang mempunyai derajat keluar = 0.

Mesin Stata Hingga dan Automata Hingga

Obyektif :

16. Memahami Mesin State Hingga

17. Memahami Automata Hingga

Mesin State Hingga

Mesin State Hingga merupakan suatu struktur abstrak yang didefinisikan terdiri atas :

(1) Himpunan hingga A, berisi simbol input

(2) Himpunan hingga S, berisi internal state

(3) Himpunan hingga Z, berisi simbol output

(4) Sebuah fungsi f : S x A $\rightarrow$ S, disebut fungsi next-state

(5) Seubuah fungsi g : S x A $\rightarrow$ Z disebut fungsi output

INPUT	:	Untai
OUTPUT	:	Untai

$\Rightarrow$ M (A, S, Z, f, g)

$\Rightarrow$ M (A, S, Z, q0, f, g)

Contoh : M (A, S, Z, f, g) dengan :

(1) A = (a,b)

(2) S = (q0, q1, q2)

(3) Z = (x, y, z)

(4) f : S x A $\rightarrow$ S, yang didefinisikan sebagai :

f (qo, a) = q1	f (q0, b) = q2
f (q1, a) = q2	f (q1, b) = q1
f (q2, a) = qo	f (q2, b) = q1

(5) g : S x A $\rightarrow$ Z, yang didefinisikan sebagai :

g (q0, a) = x	g (q0, b)= y
g (q1, a) = x	g (q1, b)= z
g (q2, a) = z	g (q2, b)= y

Automata Hingga

Automata Hingga merupakan suatu struktur abstrak yang didefinisikan terdiri atas :

(1) Himpunan hingga A, berisi simbul input

(2) Himpunan hingga S, berisi internal state

(3) Himpunan T (dimana T $\subset$ S), elemennya disebut state penerima

(4) State awal (biasanya q0), anggota S

(5) Fungsi next-state f : S x A $\rightarrow$ S

INPUT	:	Untai
OUTPUT	:	Diterima atau ditolak

$\Rightarrow$ M (A, S, T, qo, f)

Contoh : M (A, S, T, qo, f) dengan :

(1) A = { a, b }

(2) S = { q0, q1, q2 }

(3) T = { qo, q1 }

(4) State awal = q0

(5) Fungsi next-state f : S x A $\rightarrow$ S, yang didefinisikan sebagai tabel berikut :

f	**a**	**b**
q0	q0	q1
q1	q0	q2
q2	q2	q2