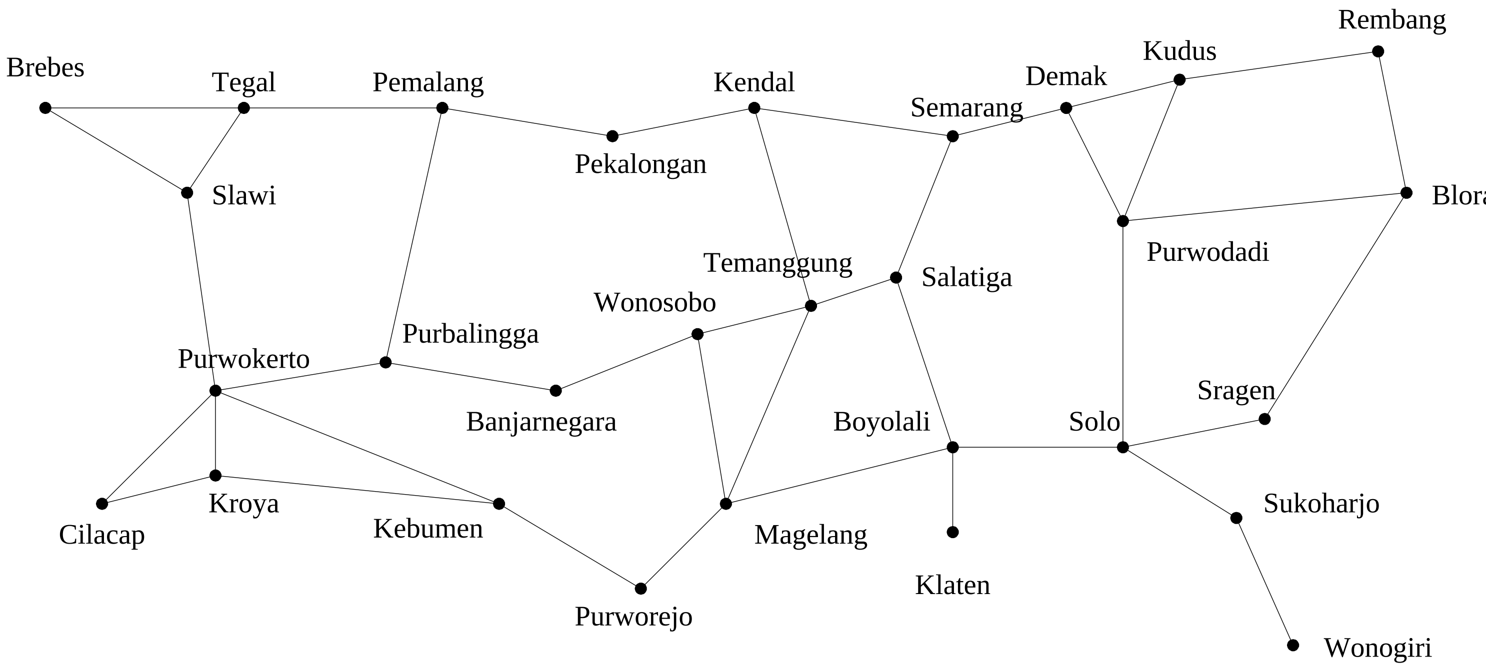
GRAPH

Graph Graph

- Graph digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut.
- Gambar berikut ini sebuah graph yang menyatakan peta jaringan jalan raya yang menghubungkan sejumlah kota di Provinsi Jawa Tengah.

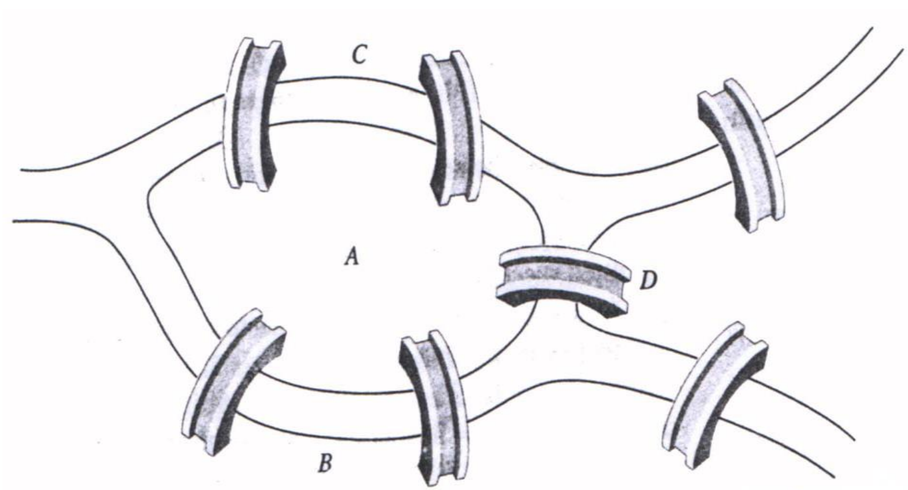
Graph

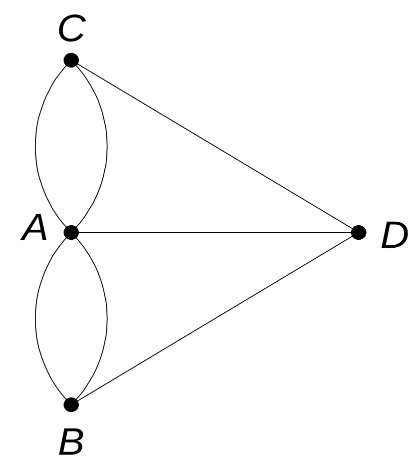


Rembang
Kudus
Brebes
Demak
Tegal
Pemalang
Kendal
Semarang
Pekalongan
Slawi
Blora
Purwodadi
Temanggung
Salatiga
Wonosobo
Purbalingga
Purwokerto
Sragen
Banjarnegara
Boyolali
Solo
Kroya
Sukoharjo
Kebumen
Cilacap
Magelang
Klaten
Purworejo
Wonogiri

Graph

- Sejarah Graph: masalah jembatan Königsberg (tahun 1736)





Graph yang merepresentasikan jembatan Königsberg:

Simpul (*vertex*) → menyatakan daratan

Sisi (*edge*) → menyatakan jembatan

Bisakah melalui setiap jembatan tepat sekali dan kembali lagi ke tempat semula?

Definisi Graph

Graph $G = (V, E)$, yang dalam hal ini:

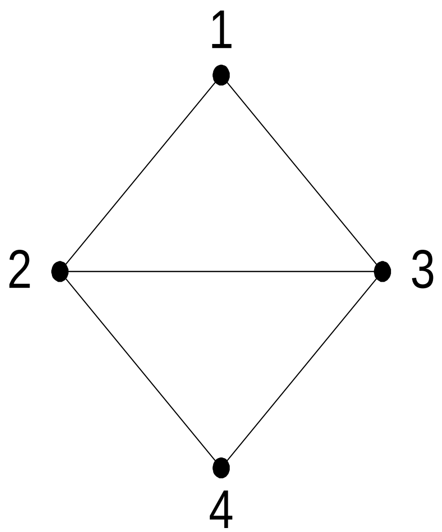
V = himpunan tidak-kosong dari simpul-simpul (*vertices*)

$= \{ v1 , v2 , ... , vn \}$

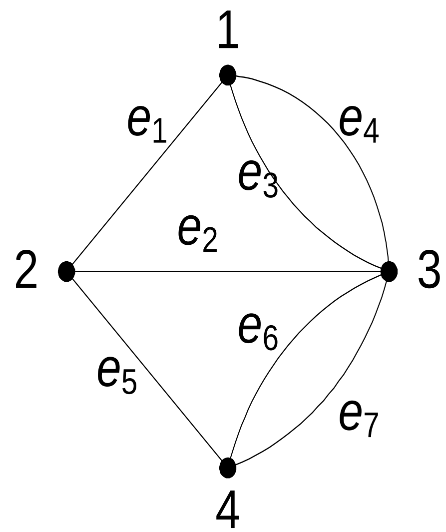
E = himpunan sisi (*edges*) yang menghubungkan sepasang simpul

$= \{e1 , e2 , ... , en \}$

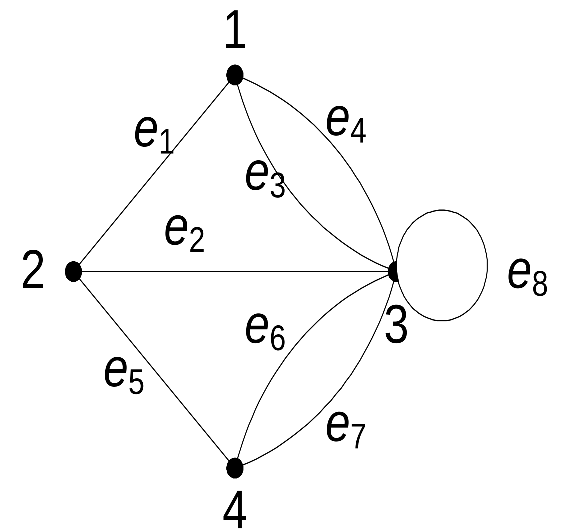
Graph



*G*1



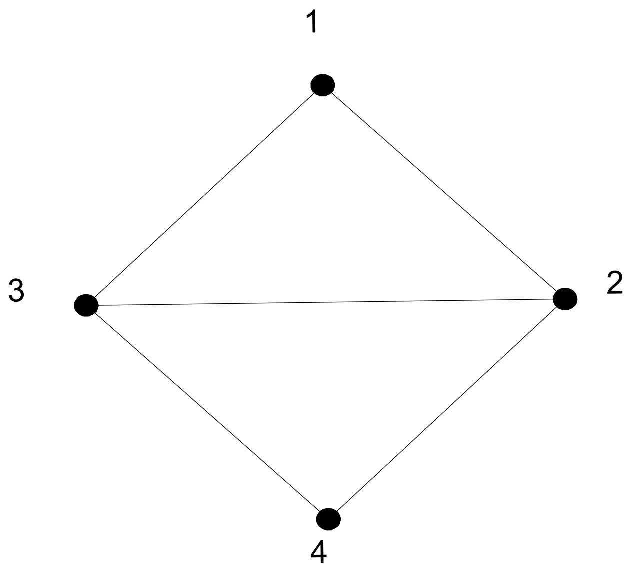
*G*2



*G*3

Graph

Graph G_1



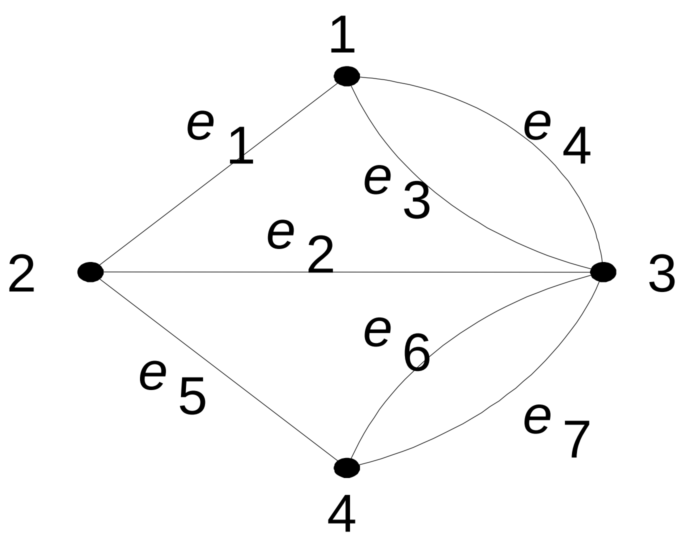
G_1 adalah graph dengan

V = { 1, 2, 3, 4 }

E = { (1, 2), (1, 3), (2, 3), (2, 4), (3, 4) }

Graph

- Graph G_2



G_2 adalah graph dengan

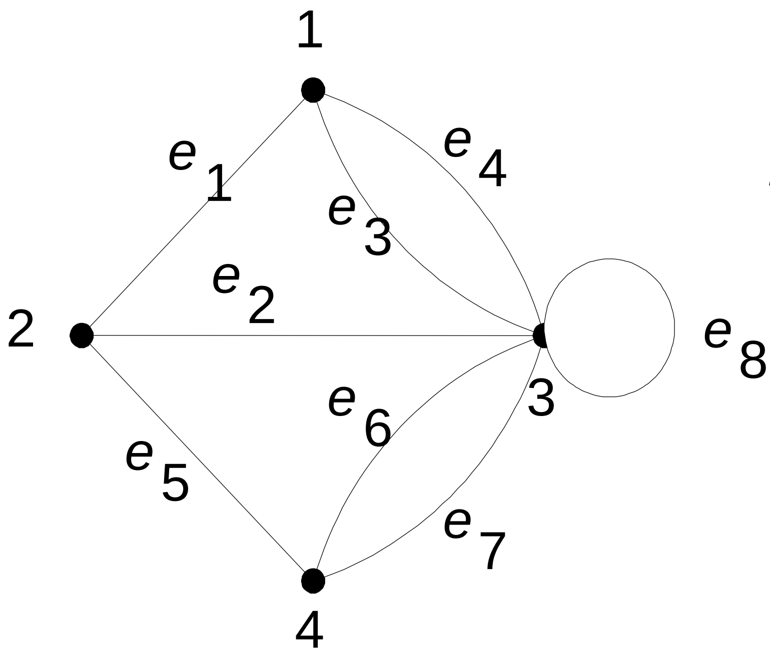
$V = \{ 1, 2, 3, 4 \}$

$E = \{ (1, 2), (2, 3), (1, 3), (1, 3), (2, 4), (3, 4), (3, 4) \}$

$= \{ e1, e2, e3, e4, e5, e6, e7\}$

Graph

- Graph G_3



G_3 adalah graph dengan

V = { 1, 2, 3, 4 }

E = { (1, 2), (2, 3), (1, 3),

(1, 3), (2, 4), (3, 4),

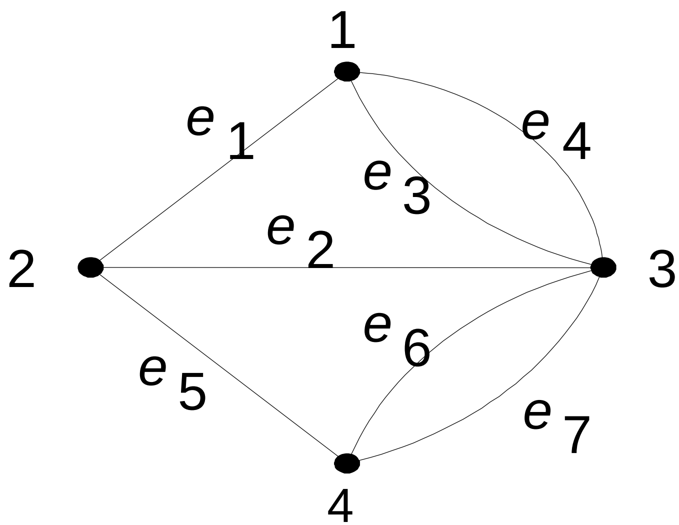
(3, 4), (3, 3) }

= { $e1$, $e2$, $e3$, $e4$, $e5$, $e6$,

$e7$, $e8$}

Graph

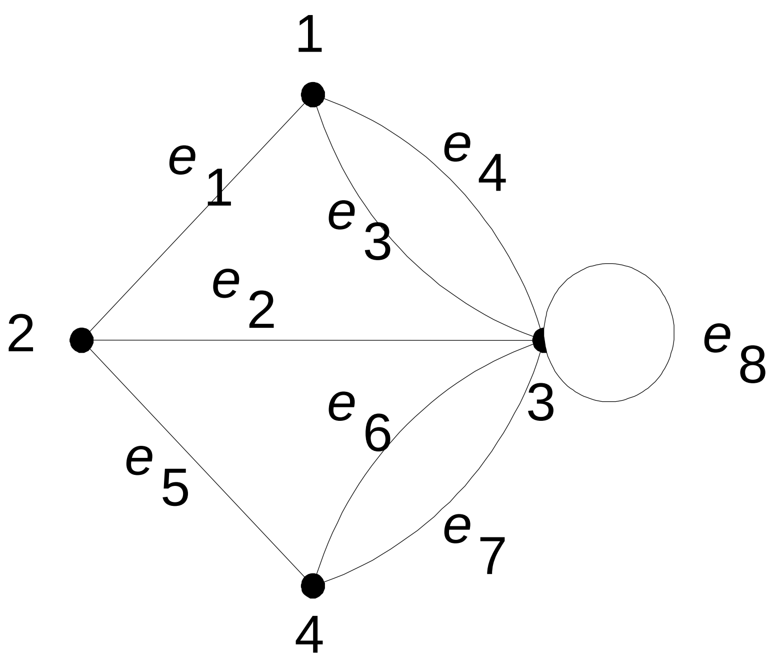
- Graph G_2



Pada G_2, sisi e_3 = (1, 3) dan sisi e_4 = (1, 3) dinamakan **sisi-ganda** (*multiple edges* atau *paralel edges*) karena kedua sisi ini menghubungi dua buah simpul yang sama, yaitu simpul 1 dan simpul 3.

Graph

- Graph G_3



Pada G_3, sisi e_8 = (3, 3) dinamakan **gelang** atau **kalang** (*loop*) karena ia berawal dan berakhir pada simpul yang sama.

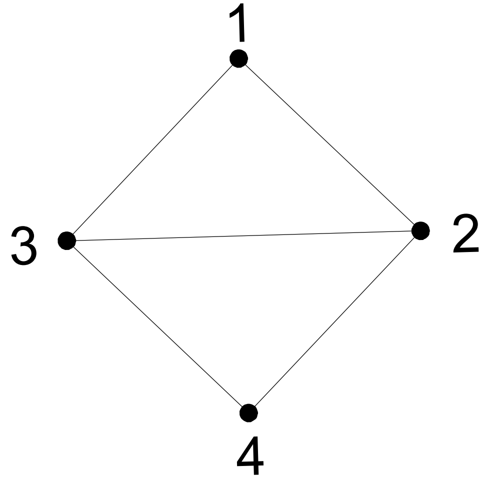
Jenis-Jenis Graph

Berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graph, maka graph digolongkan menjadi dua jenis:

1. **Graph sederhana** (*simple graph*).
2. **Graph tak-sederhana** (*unsimple-graph*).

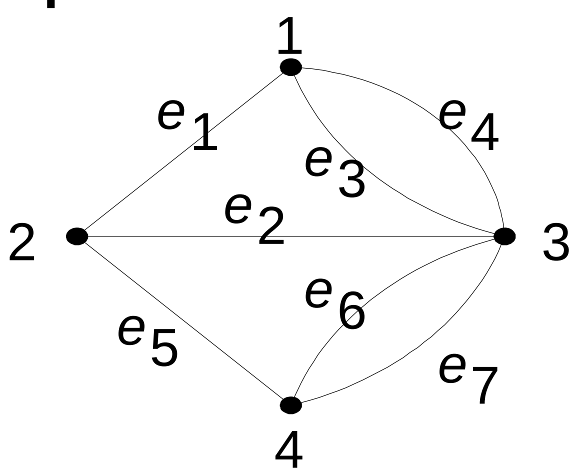
Graph sederhana (*simple graph*)

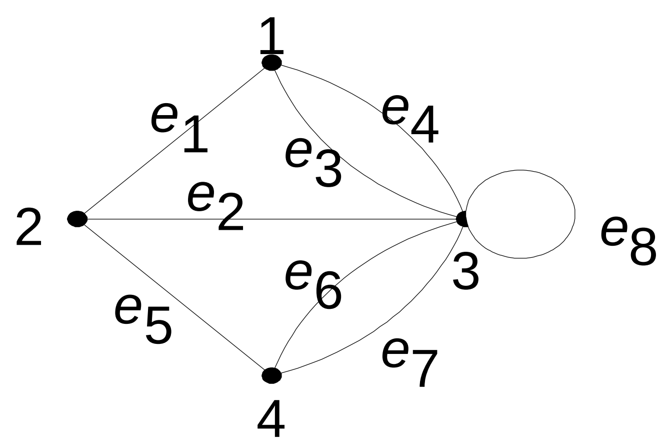
Graph yang tidak mengandung gelang maupun sisi-ganda dinamakan graph sederhana. *G*1 adalah contoh graph sederhana



Graph tak-sederhana (*unsimple-graph)*

Graph yang mengandung sisi ganda atau gelang dinamakan graph tak-sederhana (*unsimple graph*). *G*2 dan *G*3 adalah contoh graph tak-sederhana





Jenis-Jenis Graph

Berdasarkan jumlah simpul pada suatu graph, maka secara umum graph dapat digolongkan menjadi dua jenis:

1. **Graph berhingga** (*limited graph*)
2. **Graph tak-berhingga** (*unlimited graph*)

Graph berhingga (*limited graph*)

Graph berhingga adalah graph yang jumlah simpulnya, *n*, berhingga.

Graph tak-berhingga (*unlimited graph*)

Graph yang jumlah simpulnya, *n*, tidak berhingga banyaknya disebut **graph tak-berhingga**.

- Berdasarkan orientasi arah pada sisi, maka secara umum graph dibedakan atas 2 jenis:
- 1. **Graph tak-berarah** (*undirected graph*)
- Graph yang sisinya tidak mempunyai orientasi arah disebut graph tak-berarah. Tiga buah graph pada Gambar 2 adalah graph tak-berarah.
- 2. **Graph berarah** (*directed graph* atau *digraph*)
- Graph yang setiap sisinya diberikan orientasi arah disebut sebagai graph berarah. Dua buah graph pada Gambar 3 adalah graph berarah.

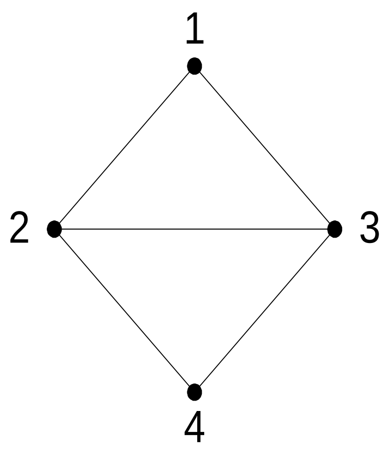
Jenis-Jenis Graph

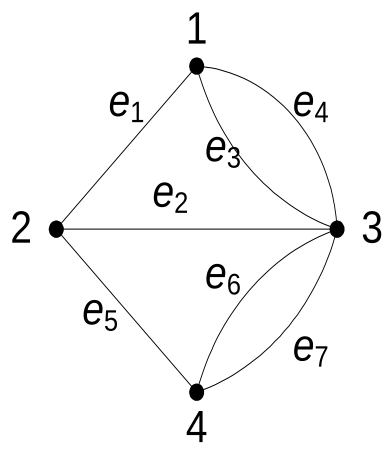
Berdasarkan orientasi arah pada sisi, maka secara umum graph dibedakan atas 2 jenis:

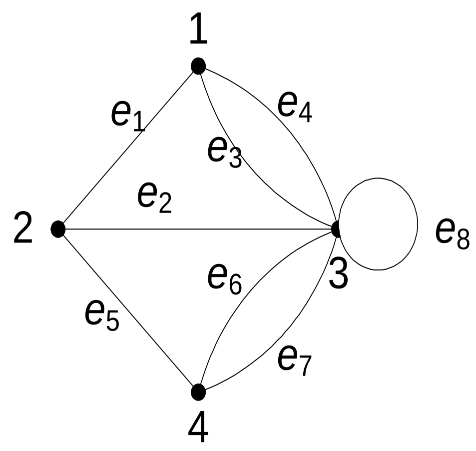
1. **Graph tak-berarah** (*undirected graph*)
2. **Graph berarah** (*directed graph* atau *digraph*)

Graph tak-berarah (*undirected graph*)

Graph yang sisinya tidak mempunyai orientasi arah disebut graph tak-berarah. Graph G_1, G_2, dan G_3 adalah graph tak-berarah.

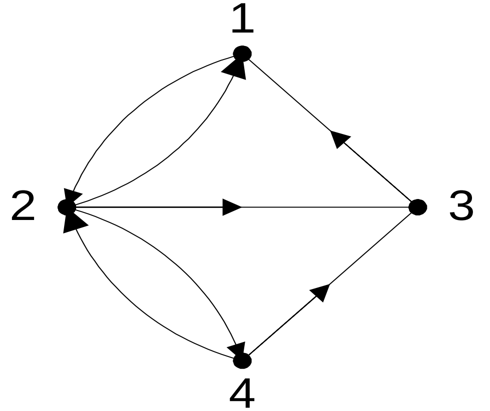




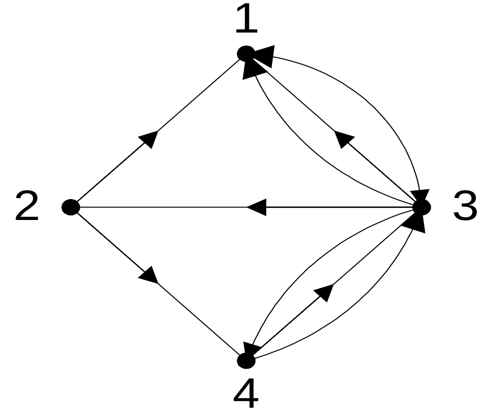


Graph berarah (*directed graph* atau *digraph*)

Graph yang setiap sisinya diberikan orientasi arah disebut sebagai graph berarah.



(a) *G*4



(b) *G*5

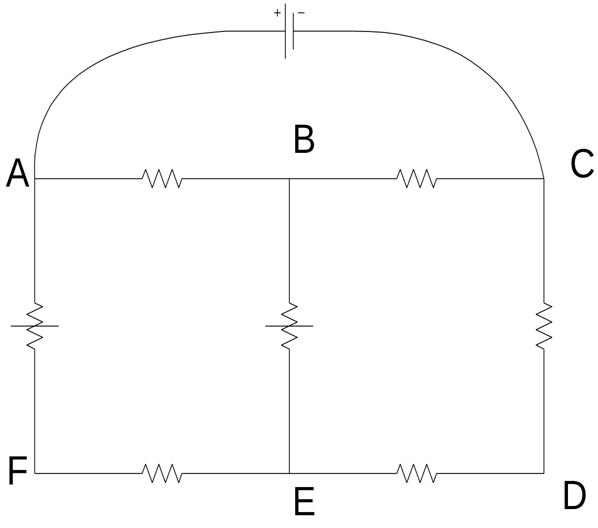
(a) graph berarah, (b) graph-ganda berarah

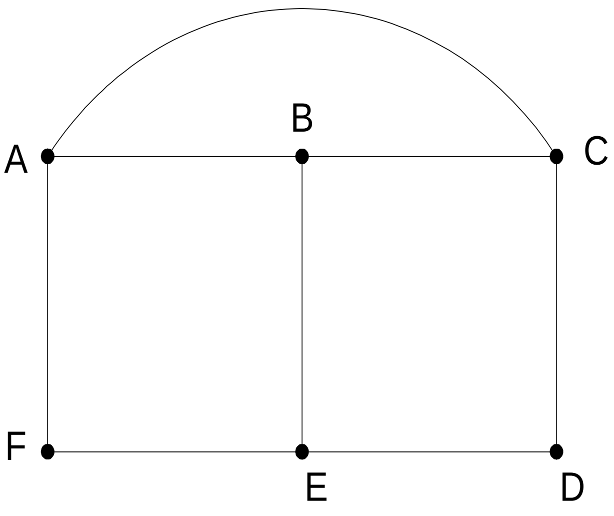
Jenis-jenis graph [ROS99]

Jenis	Sisi	Sisi ganda dibolehkan?	Sisi gelang dibolehkan?
Graph sederhana	Tak-berarah	Tidak	Tidak
Graph ganda	Tak-berarah	Ya	Tidak
Graph semu	Tak-berarah	Ya	Ya
Graph berarah	Bearah	Tidak	Ya
Graph-ganda berarah	Bearah	Ya	Ya

Contoh Terapan Graph

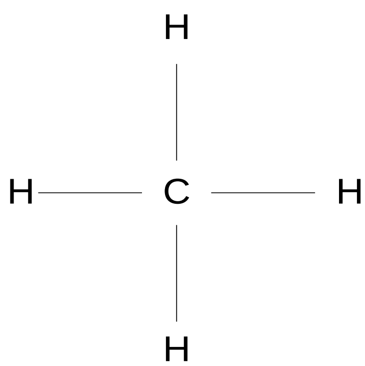
- *Rangkaian listrik*.



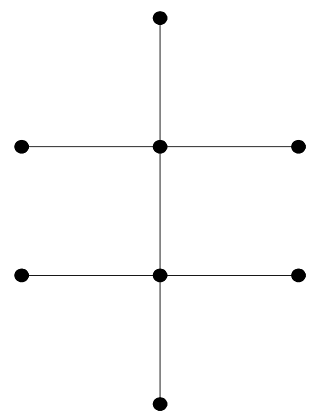


Contoh Terapan Graph

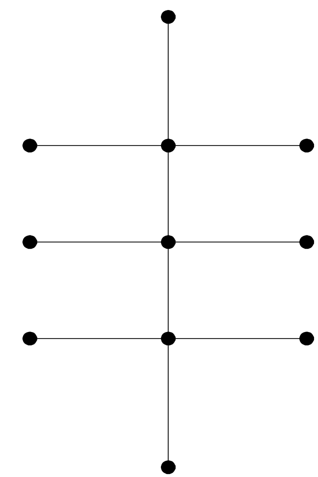
- *Isomer senyawa kimia karbon*



metana (CH4)



etana (C2H6)

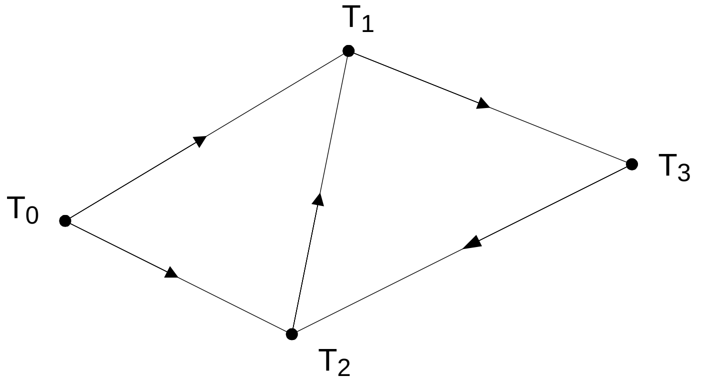


propana (C3H8)

Contoh Terapan Graph

Transaksi konkuren pada basis data terpusat

Transaksi T_0 menunggu transaksi T_1 dan T_2
Transaksi T_2 menunggu transaksi T_1
Transaksi T_1 menunggu transaksi T_3
Transaksi T_3 menunggu transaksi T_2

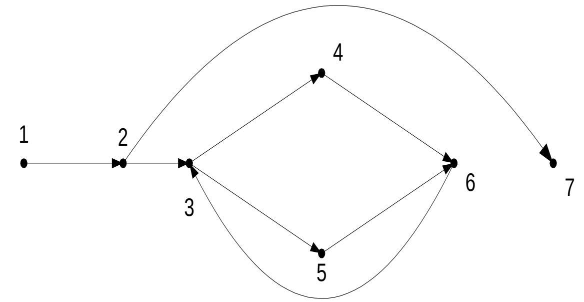


Contoh Terapan Graph

. *Pengujian program*

read(x);
while x <> 9999 **do**
 begin
   if x < 0 **then**
      writeln(‘Masukan tidak boleh negatif’)
   else
      x:=x+10;
   read(x);
 end;
writeln(x);

keterangan



Keterangan:
1 : read(x)
2 : x <> 9999
3 : x < 0
4 : writeln(‘Masukan tidak boleh negatif’);
5 : x := x + 10
6 : read(x)
7 : writeln(x)

Contoh Terapan Graph

Terapan graph pada teori otomata [LIU85].

Mesin jaja (*vending machine*)

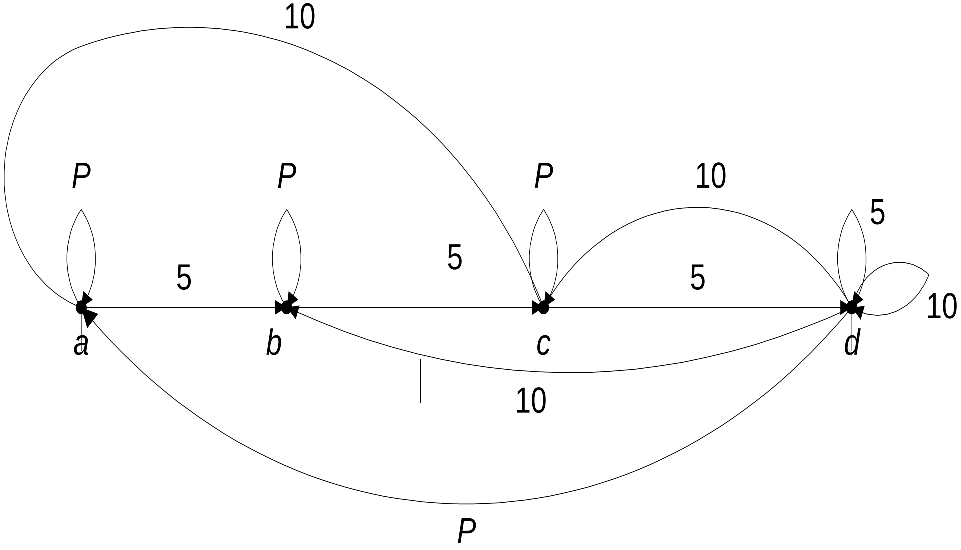
Keterangan:

a : 0 sen dimasukkan

b : 5 sen dimasukkan

c : 10 sen dimasukkan

d : 15 sen atau lebih dimasukkan



Ketetanggaan (*Adjacent*)

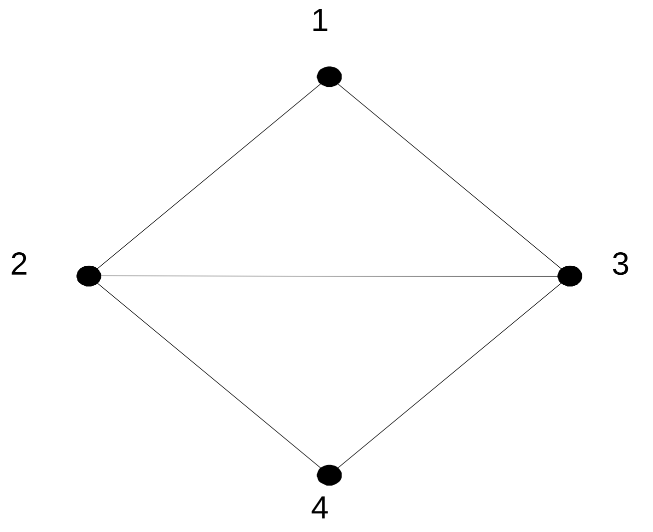
Dua buah simpul dikatakan *bertetangga* bila keduanya terhubung langsung.

Tinjau graph :

simpul 1 bertetangga dengan simpul 2 dan 3,

simpul 1 tidak bertetangga dengan simpul 4.

- Graph



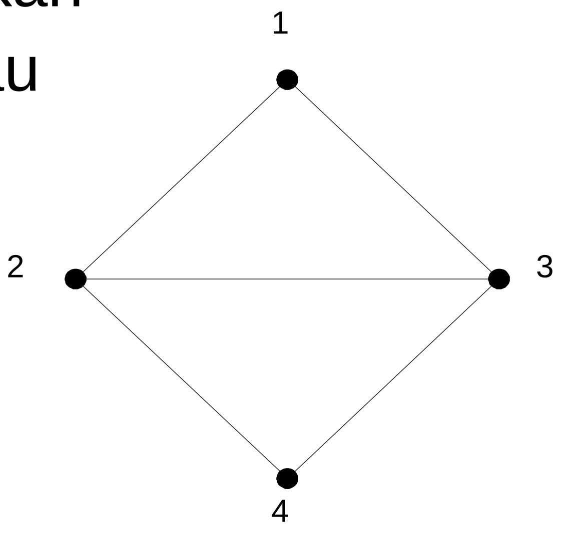
Bersisian (*Incidency*)

Untuk sembarang sisi *e* = (*vj*, *vk*) dikatakan

e bersisian dengan simpul *vj* , atau

e bersisian dengan simpul *vk*

Tinjau graph :



sisi (2, 3) bersisian dengan simpul 2 dan simpul 3,

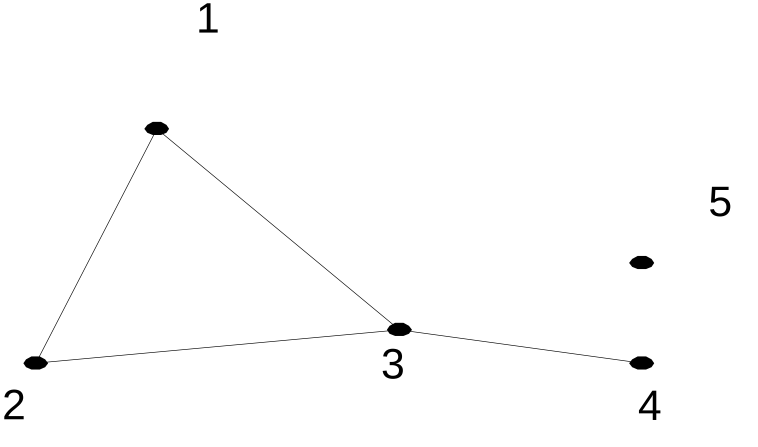
sisi (2, 4) bersisian dengan simpul 2 dan simpul 4,

tetapi sisi (1, 2) tidak bersisian dengan simpul 4.

Simpul Terpencil (*Isolated Vertex*)

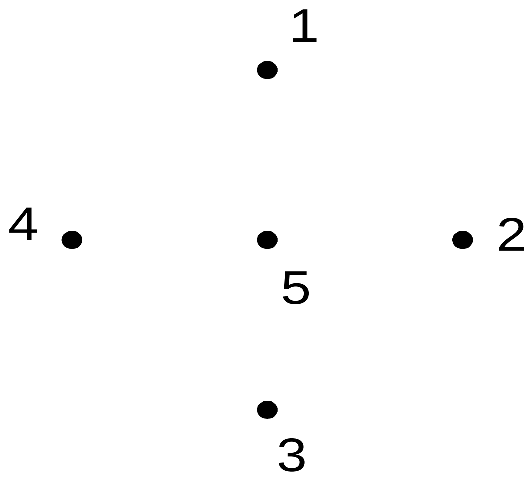
Simpul terpencil ialah simpul yang tidak mempunyai sisi yang bersisian dengannya.

Tinjau graph : simpul 5 adalah simpul terpencil



Graph Kosong (*null graph* atau *empty graph*)

Graph yang himpunan sisinya merupakan himpunan kosong (N_n).



Derajat (*Degree*)

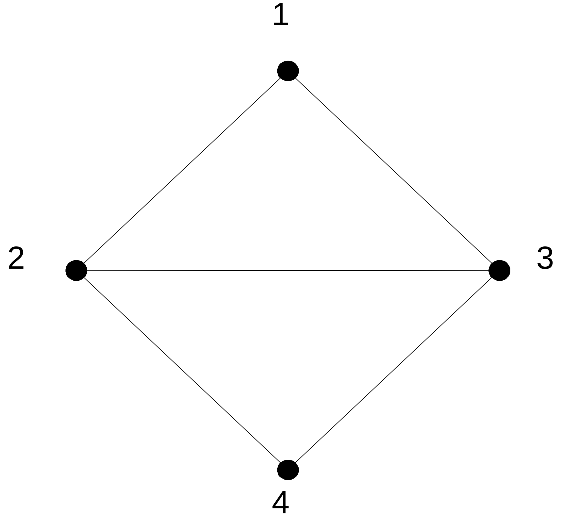
Derajat suatu simpul adalah jumlah sisi yang bersisian dengan simpul tersebut.

Notasi: $d(v)$

Tinjau graph $G1$:

$d(1) = d(4) = 2$

$d(2) = d(3) = 3$



Derajat (*Degree*)

Tinjau graph *G*3:

d(5) = 0 → simpul terpencil

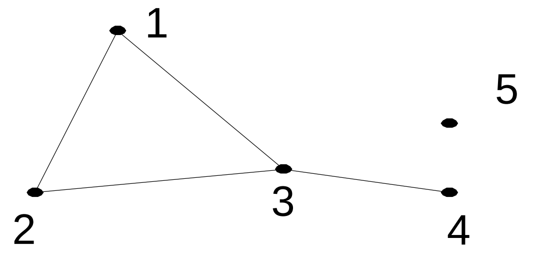
d(4) = 1 → simpul anting-anting (*pendant vertex*)

Tinjau graph *G*2:

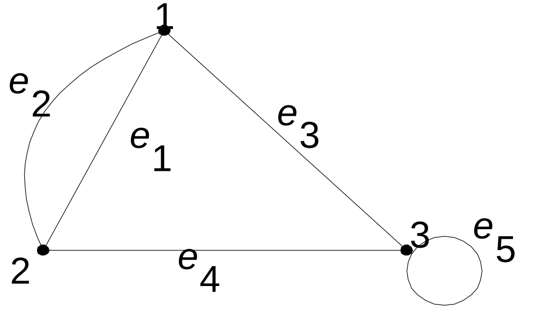
d(1) = 3 → bersisian dengan sisi ganda

d(2) = 4 → bersisian dengan sisi gelang (*loop*)

- Graph G3



- Graph G2



Derajat (*Degree*)

Pada graph berarah,

$d_{\text{in}}(v)$ = derajat-masuk (*in-degree*)

= jumlah busur yang masuk ke simpul v

$d_{\text{out}}(v)$ = derajat-keluar (*out-degree*)

= jumlah busur yang keluar dari simpul v

$$d(v) = d_{in}(v) + d_{out}(v)$$

Derajat (*Degree*)

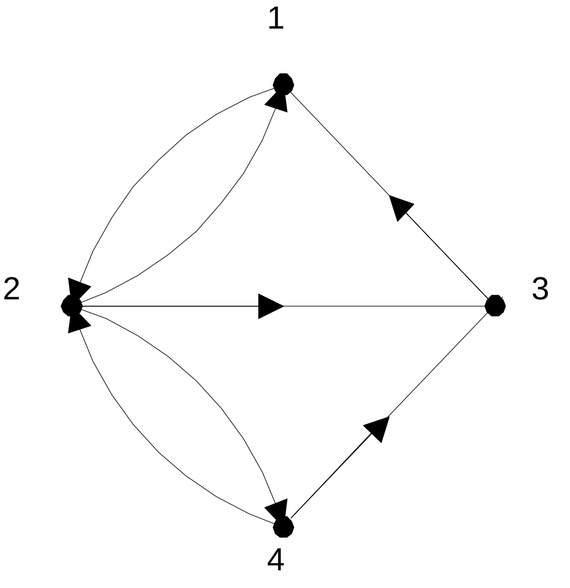
Tinjau graph :

$d_{in}(1) = 2$; $d_{out}(1) = 1$

$d_{in}(2) = 2$; $d_{out}(2) = 3$

$d_{in}(3) = 2$; $d_{out}(3) = 1$

$d_{in}(4) = 1$; $d_{out}(4) = 2$



Lemma Jabat Tangan

Jumlah derajat semua simpul pada suatu graph adalah genap, yaitu dua kali jumlah sisi pada graph tersebut.

Dengan kata lain, jika $G = (V, E)$, maka

$$\sum_{v \in V} d(v) = 2|E|$$

Lemma Jabat Tangan

Tinjau graph $G1$:

$d(1) + d(2) + d(3) + d(4) =$

$2 + 3 + 3 + 2 = 10 =$

$2 \times$ jumlah sisi $= 2 \times 5$

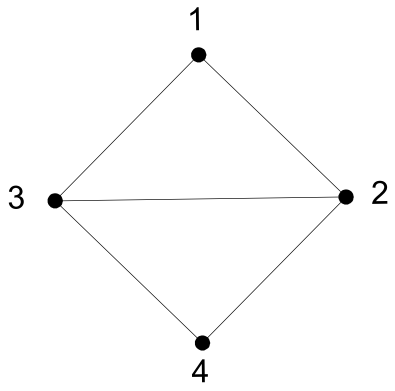
Tinjau graph $G2$:

$d(1) + d(2) + d(3)$

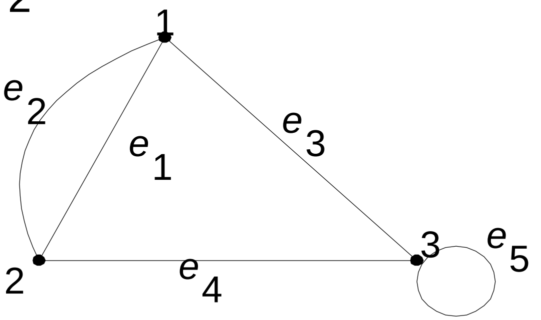
$= 3 + 3 + 4 = 10$

$= 2 \times$ jumlah sisi $= 2 \times 5$

- Graph G_1



- Graph G_2



Lemma Jabat Tangan

Tinjau graph G_3:

$d(1) + d(2) + d(3) + d(4) + d(5)$

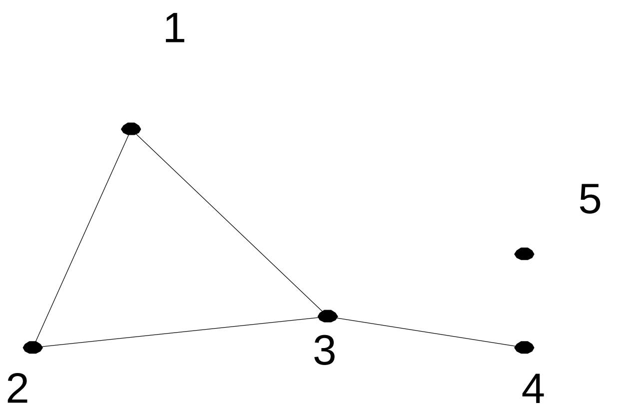
$= 2 + 2 + 3 + 1 + 0$

$= 8$

$= 2 \times$ jumlah sisi

$= 2 \times 4$

- Graph G_3



Lemma Jabat Tangan

Contoh.

Diketahui graph dengan lima buah simpul. Dapatkah kita menggambar graph tersebut jika derajat masing-masing simpul adalah:

(a) 2, 3, 1, 1, 2

(b) 2, 3, 3, 4, 4

<u>Penyelesaian</u>:

(a) tidak dapat, karena jumlah derajat semua simpulnya ganjil
$(2 + 3 + 1 + 1 + 2 = 9)$.

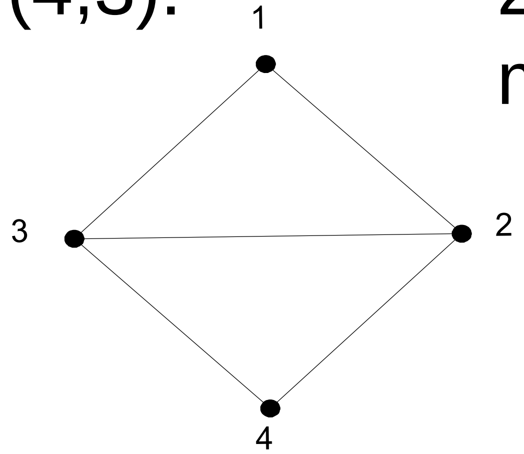
(b) dapat, karena jumlah derajat semua simpulnya genap
$(2 + 3 + 3 + 4 + 4 = 16)$.

Lintasan (*Path*)

Lintasan yang panjangnya n dari simpul awal v_0 ke simpul tujuan v_n di dalam graph G ialah barisan berselang-seling simpul-simpul dan sisi-sisi yang berbentuk v_0, e_1, v_1, e_2, v_2,... , v_{n-1}, e_n, v_n sedemikian sehingga $e_1 = (v_0, v_1)$, $e_2 = (v_1, v_2)$, ... , $e_n = (v_{n-1}, v_n)$ adalah sisi-sisi dari graph G.

Lintasan (*Path*)

- Tinjau graph *G*1: lintasan 1, 2, 4, 3 adalah lintasan dengan barisan sisi (1,2), (2,4), (4,3).



- **Panjang lintasan** adalah jumlah sisi dalam lintasan tersebut. Lintasan 1, 2, 4, 3 pada *G*1 memiliki panjang 3.

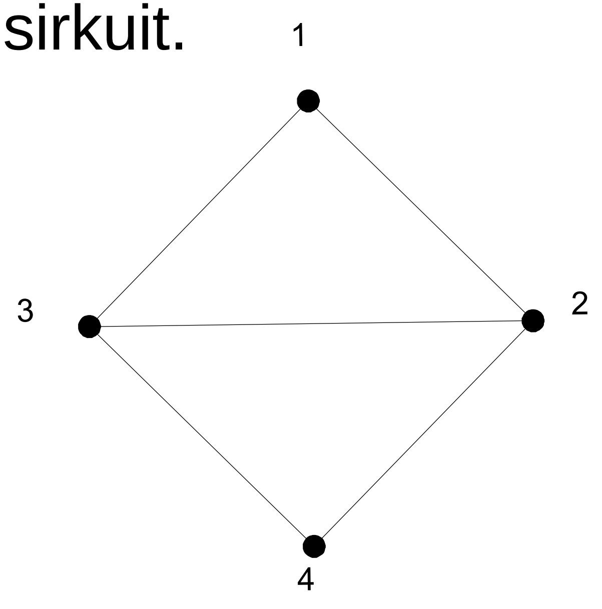
Siklus (*Cycle*) atau Sirkuit (*Circuit*)

- Lintasan yang berawal dan berakhir pada simpul yang sama disebut **sirkuit** atau **siklus**.

- **Panjang sirkuit** adalah jumlah sisi dalam sirkuit tersebut. Sirkuit 1, 2, 3, 1 pada *G*1 memiliki panjang 3.

- Tinjau graph *G*1:

  1, 2, 3, 1 adalah sebuah sirkuit.



Terhubung (*Connected*)

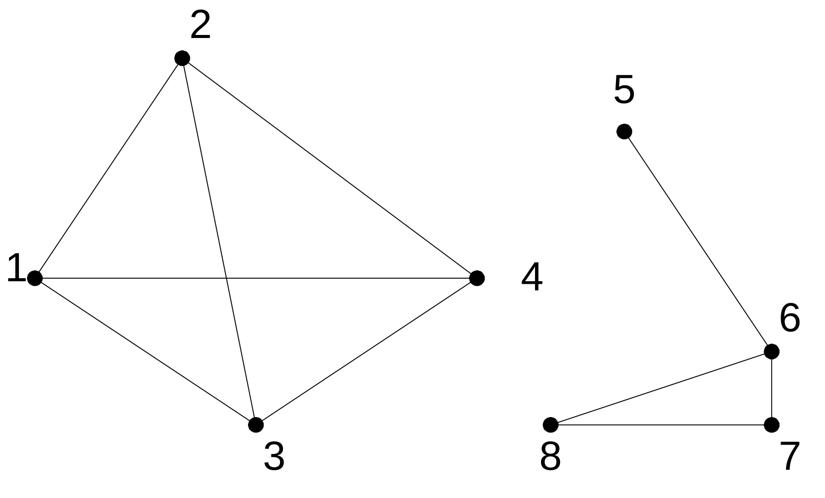
Dua buah simpul v_1 dan simpul v_2 disebut **terhubung** jika terdapat lintasan dari v_1 ke v_2.

G disebut **graph terhubung** (*connected graph*) jika untuk setiap pasang simpul v_i dan v_j dalam himpunan *V* terdapat lintasan dari v_i ke v_j

Jika tidak, maka *G* disebut **graph tak-terhubung** (*disconnected graph*).

Terhubung (*Connected*)

- Contoh graph tak-terhubung:



Terhubung (*Connected*) Graph berarah

- Graph berarah *G* dikatakan terhubung jika graph tidak berarahnya terhubung (graph tidak berarah dari *G* diperoleh dengan menghilangkan arahnya).

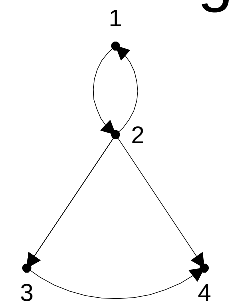
Terhubung (*Connected*) Graph berarah

- Dua simpul, *u* dan *v*, pada graph berarah *G* disebut **terhubung kuat** (*strongly connected*) jika terdapat lintasan berarah dari *u* ke *v* dan juga lintasan berarah dari *v* ke *u*.
- Jika *u* dan *v* tidak terhubung kuat tetapi terhubung pada graph tidak berarahnya, maka *u* dan *v* dikatakan **terhubung lemah** (*weakly connected*).

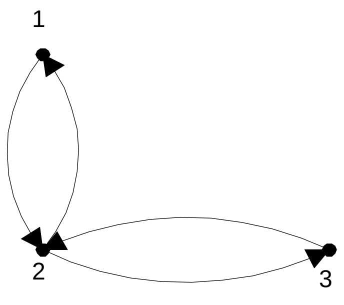
Terhubung (*Connected*) Graph berarah

Graph berarah *G* disebut **graph terhubung kuat** (*strongly connected graph*) apabila untuk setiap pasang simpul sembarang *u* dan *v* di *G*, terhubung kuat. Kalau tidak, *G* disebut **graph terhubung lemah**.

- Graph berarah terhubung lemah



- Graph berarah terhubung kuat



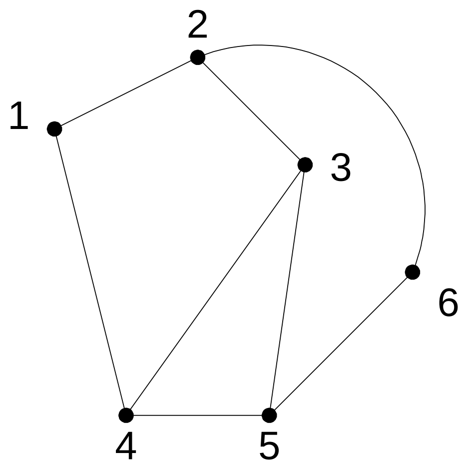
Upagraph (*Subgraph*) dan Komplemen Upagraph

- Misalkan $G = (V, E)$ adalah sebuah graph.

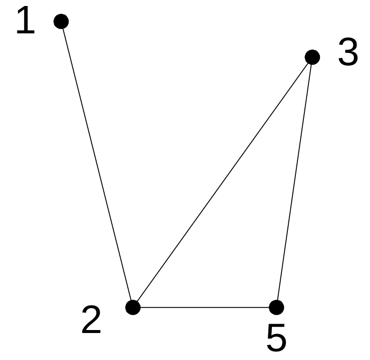
  $G1 = (V_1, E_1)$ adalah **upagraph** (*subgraph*) dari G jika $V_1 \subseteq V$ dan $E_1 \subseteq E$.

- **Komplemen** dari upagraph G_1 terhadap graph G adalah graph $G_2 = (V_2, E_2)$ sedemikian sehingga $E_2 = E - E_1$ dan V_2 adalah himpunan simpul yang anggota-anggota E_2 bersisian dengannya.

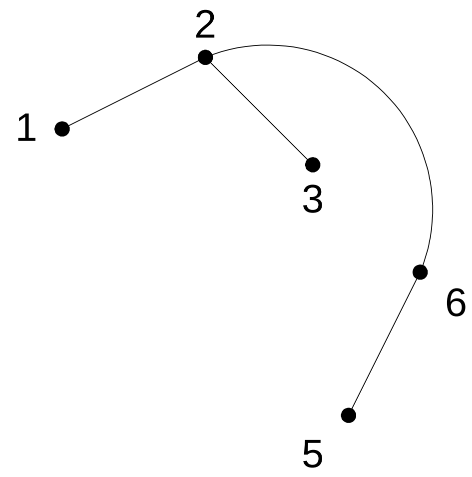
Upagraph (*Subgraph*) dan Komplemen Upagraph



(a) Graph *G*1



(b) Sebuah upagraph

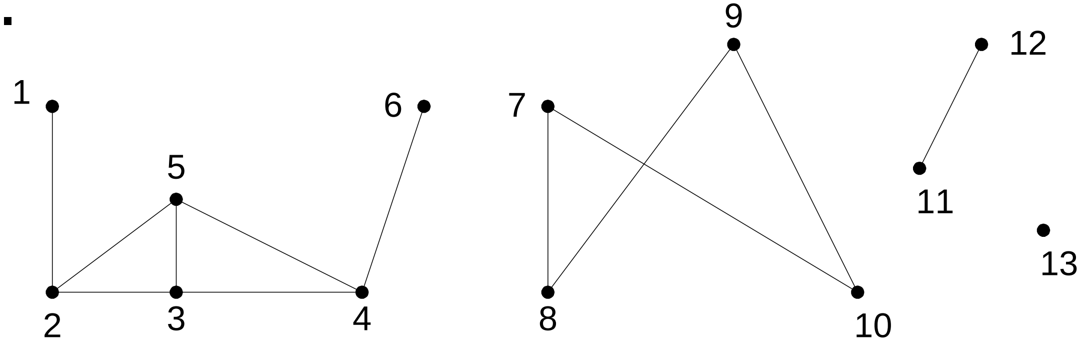


(c) komplemen
dari upagraph

Komponen graph (*connected component*)

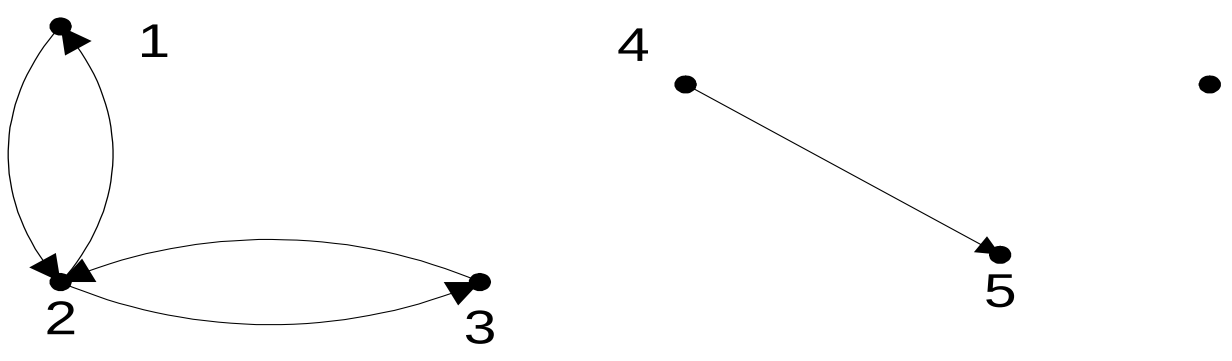
adalah jumlah maksimum upagraph terhubung dalam graph *G*.

Graph *G* di bawah ini mempunyai 4 buah komponen.



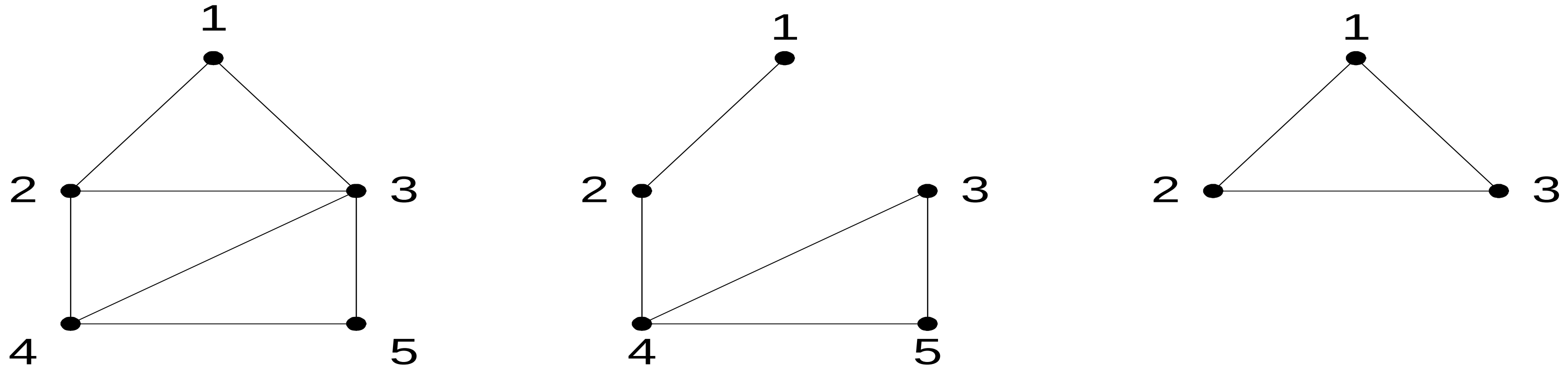
Komponen graph (*connected component*)

- Pada graph berarah, komponen terhubung kuat (*strongly connected component*) adalah jumlah maksimum upagraph yang terhubung kuat.
- Graph di bawah ini mempunyai 2 buah komponen terhubung kuat:



Upagraph Rentang (*Spanning Subgraph*)

- Upagraph $G_1 = (V_1, E_1)$ dari $G = (V, E)$ dikatakan **upagraph rentang** jika $V_1 = V$ (yaitu $G1$ mengandung semua simpul dari G).



(a) graph G, (b) upagraph rentang dari G (c)bukan upagraph rentang dari G,

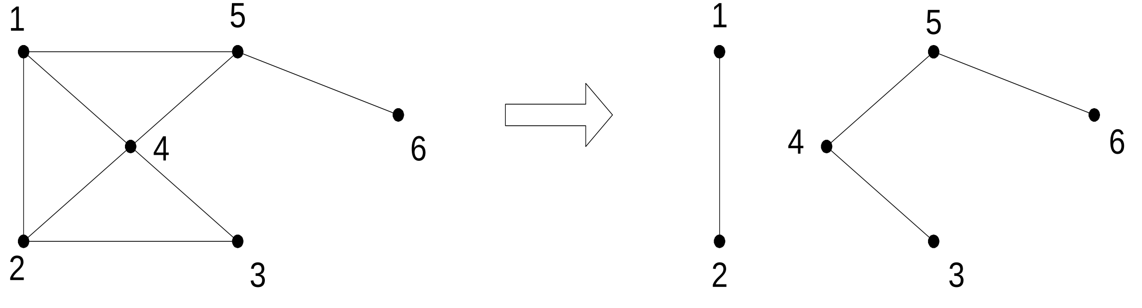
Cut-Set

Cut-set dari graph terhubung *G* adalah himpunan sisi yang bila dibuang dari *G* menyebabkan *G* tidak terhubung.

Jadi, *cut-set* selalu menghasilkan dua buah komponen.

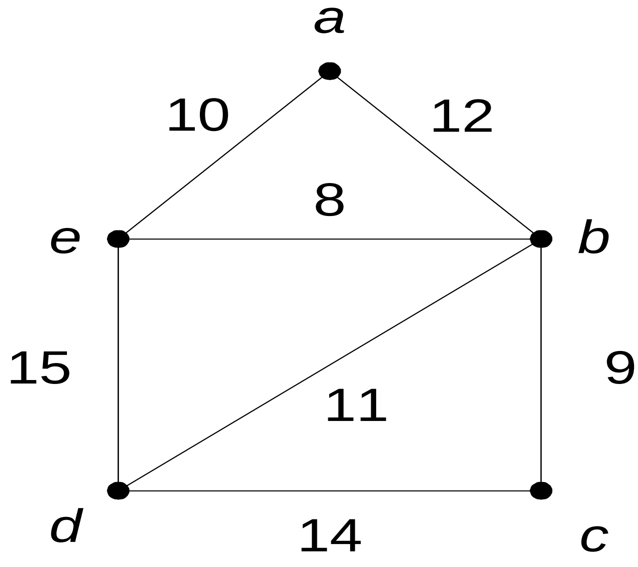
Cut-Set

- Pada graph di bawah, {(1,5), (1,4), (2,4), (2,3)} adalah *cut-set*. Terdapat banyak *cut-set* pada sebuah graph terhubung.
- Himpunan {(1,5), (4,5)} juga adalah *cut-set*, {(1,2), (1,4), (1,5)} adalah *cut-set*, {(5,6)} juga *cut-set*,
- tetapi {(1,5), (4,5), (3,4)} bukan *cut-set* sebab himpunan bagiannya, {(1,5), (4,5)} adalah *cut-set*.



Graph Berbobot (*Weighted Graph*)

Graph berbobot adalah graph yang setiap sisinya diberi sebuah harga (bobot).



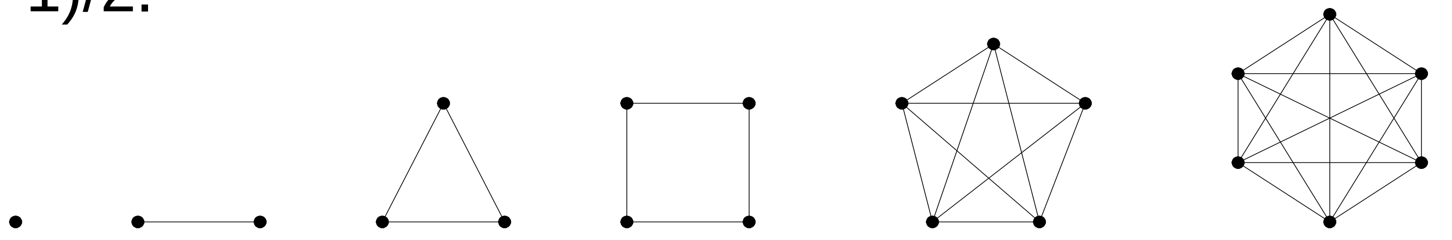
Beberapa Graph Sederhana Khusus

a. Graph Lengkap (*Complete Graph*)
b. Graph Lingkaran
c. Graph Teratur (*Regular Graphs*)
d. Graph *Bipartite* (*Bipartite Graph*)

Graph lengkap

ialah graph sederhana yang setiap simpulnya mempunyai sisi ke semua simpul lainnya. Graph lengkap dengan *n* buah simpul dilambangkan dengan K_n. Jumlah sisi pada graph lengkap yang terdiri dari *n* buah simpul adalah

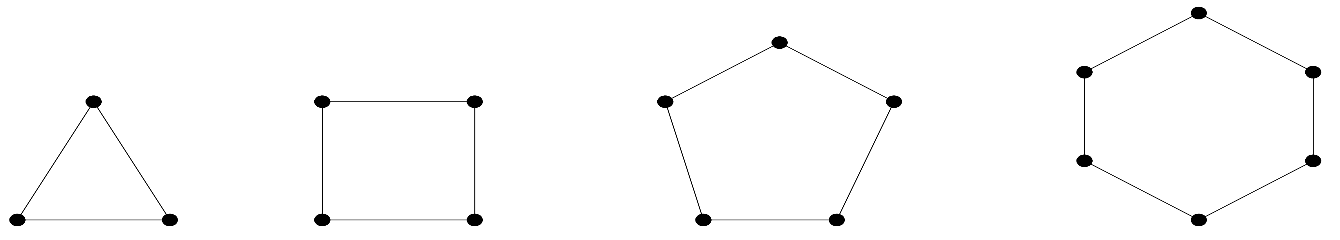
$n(n - 1)/2$.



K_1 K_2 K_3 K_4 K_5 K_6

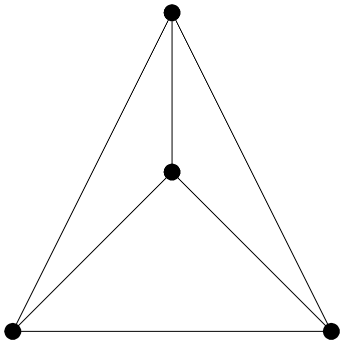
Graph lingkaran

adalah graph sederhana yang setiap simpulnya berderajat dua. Graph lingkaran dengan n simpul dilambangkan dengan C_n.



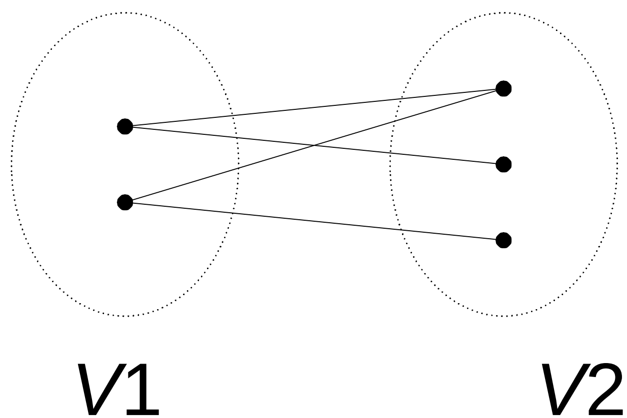
Graph Teratur (*Regular Graphs*)

Graph yang setiap simpulnya mempunyai derajat yang sama disebut **graph teratur**. Apabila derajat setiap simpul adalah *r*, maka graph tersebut disebut sebagai graph teratur derajat *r*. Jumlah sisi pada graph teratur adalah *nr*/2.



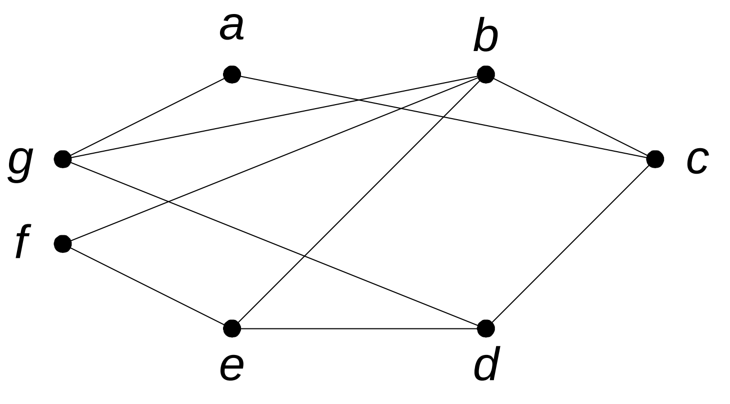
Graph *Bipartite* (*Bipartite Graph*)

Graph G yang himpunan simpulnya dapat dipisah menjadi dua himpunan bagian V_1 dan V_2, sedemikian sehingga setiap sisi pada G menghubungkan sebuah simpul di V_1 ke sebuah simpul di V_2 disebut **graph bipartit** dan dinyatakan sebagai $G(V_1, V_2)$.

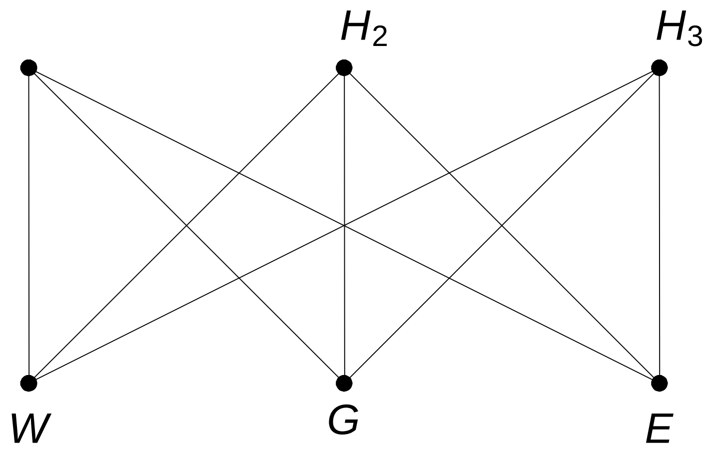


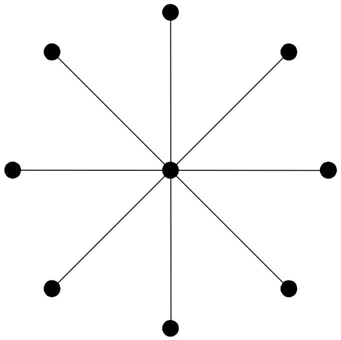
Graph *Bipartite* (*Bipartite Graph*)

Graph G di bawah ini adalah graph bipartit, karena simpul-simpunya dapat dibagi menjadi $V1 = \{a, b, d\}$ dan $V2 = \{c, e, f, g\}$



Graph *Bipartite* (*Bipartite Graph*)





Representasi Graph

1. Matriks Ketetanggaan

(*adjacency matrix*)

2. Matriks Bersisian

(*incidency matrix*)

3. Senarai Ketetanggaan

(*adjacency list*)

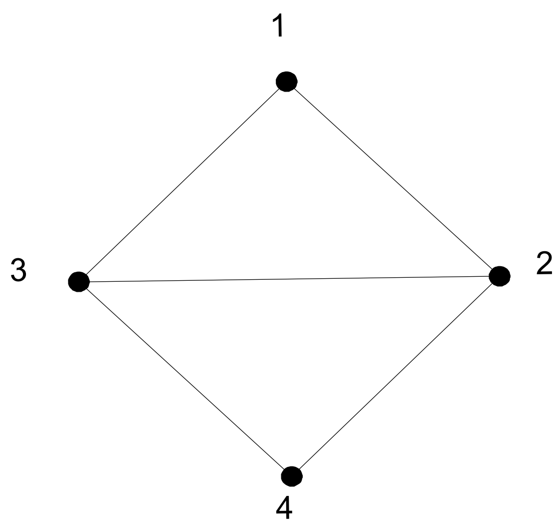
Matriks Ketetanggaan (*adjacency matrix*)

$A = [a_{ij}]$,

$$a_{ij} = \begin{cases} 1, & \text{jika simpul } i \text{ dan } j \text{ bertetangga} \\ 0, & \text{jika simpul } i \text{ dan } j \text{ tidak bertetangga} \end{cases}$$

Matriks Ketetanggaan (*adjacency matrix*)

- Graph

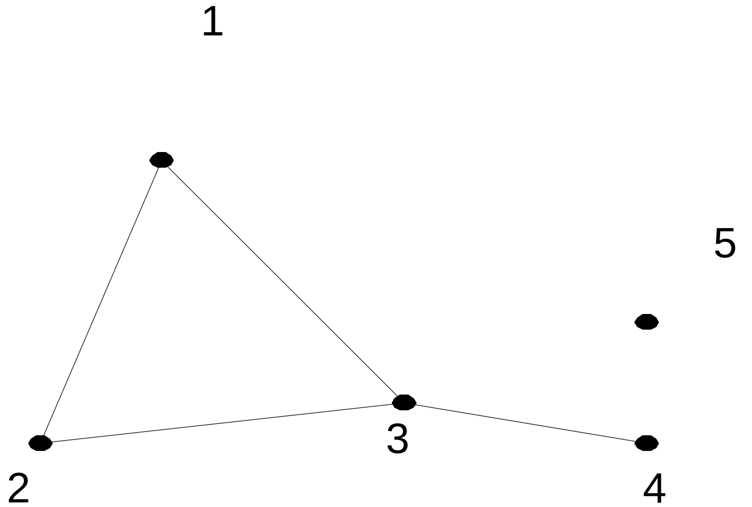


- **Matriks Ketetanggaan**

$$\begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Matriks Ketetanggaan (*adjacency matrix*)

- Graph

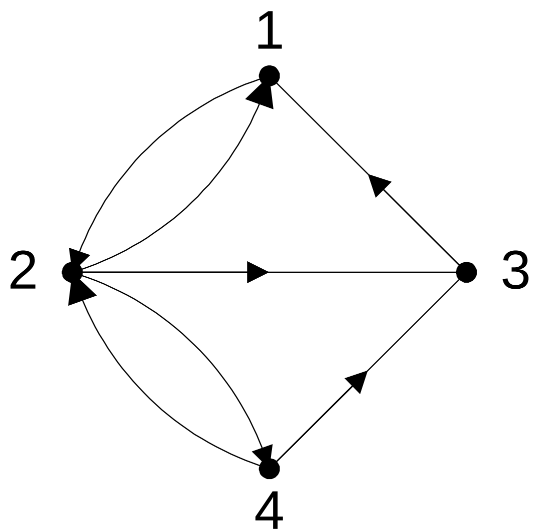


- **Matriks Ketetanggaan**

$$
\begin{array}{c c}
 & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}
\end{array}
$$

Matriks Ketetanggaan (*adjacency matrix*)

- Graph

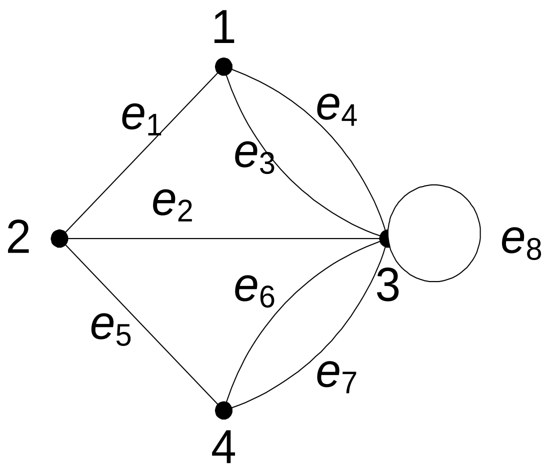


- **Matriks Ketetanggaan**

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Matriks Ketetanggaan (*adjacency matrix*)

- Graph



- **Matriks Ketetanggaan**

$$
\begin{array}{c}
\\ 1 \\ 2 \\ 3 \\ 4
\end{array}
\begin{array}{c}
\begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\
\begin{bmatrix}
0 & 1 & 2 & 0 \\
1 & 0 & 1 & 1 \\
2 & 1 & 1 & 2 \\
0 & 1 & 2 & 0
\end{bmatrix}
\end{array}
$$

Derajat tiap simpul *i*:

(a) Untuk graph tak-berarah,

$$d(v_i) = \sum_{j=1}^{n} a_{ij}$$

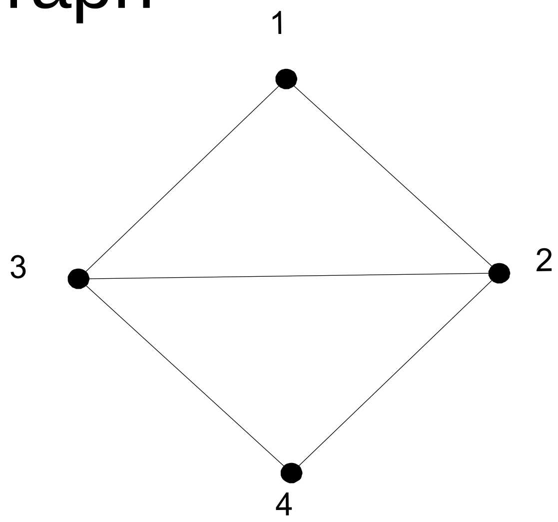
(b) Untuk graph berarah,

$d_{in}(v_j)$ = jumlah nilai pada kolom j = $\sum_{i=1}^{n} a_{ij}$

$d_{out}(v_i)$ = jumlah nilai pada baris i = $\sum_{j=1}^{n} a_{ij}$

Derajat tiap simpul

- Graph



Derajat simpul 2 = 1+0+1+1 = 3

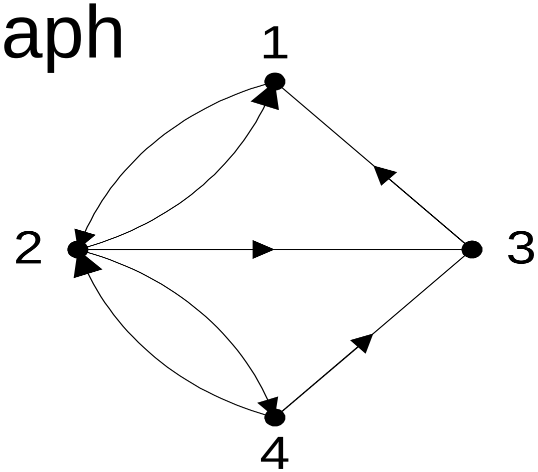
Derajat simpul 4 = 0+1+1+0 = 2

- **Matriks Ketetanggaan**

$$\begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Derajat tiap simpul

- Graph



Derajat masuk simpul 2 =
1+0+0+1 = 2

Derajat keluar simpul 2 =
1+0+1+1 = 3

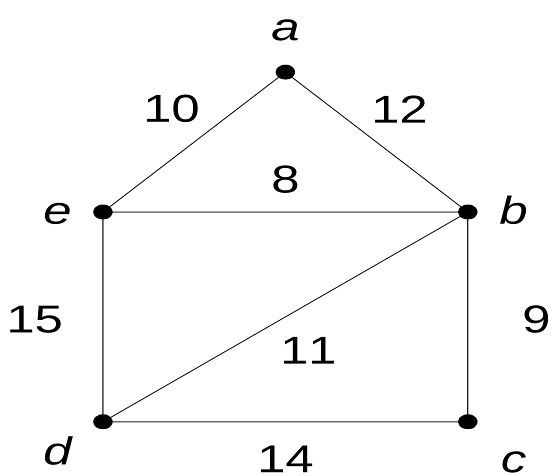
- **Matriks Ketetanggaan**

$$
\begin{array}{c c}
 & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}
\end{array}
$$

Matriks Ketetanggaan Graph Berbobot

- Graph

Tanda $\propto$ bila tdk ada sisi dari simpul I ke j



- **Matriks Ketetanggaan**

$$
\begin{array}{c c}
 & \begin{array}{ccccc} a & b & c & d & e \end{array} \\
\begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} &
\begin{bmatrix}
\infty & 12 & \infty & \infty & 10 \\
12 & \infty & 9 & 11 & 8 \\
\infty & 9 & \infty & 14 & \infty \\
\infty & 11 & 14 & \infty & 15 \\
10 & 8 & \infty & 15 & \infty
\end{bmatrix}
\end{array}
$$

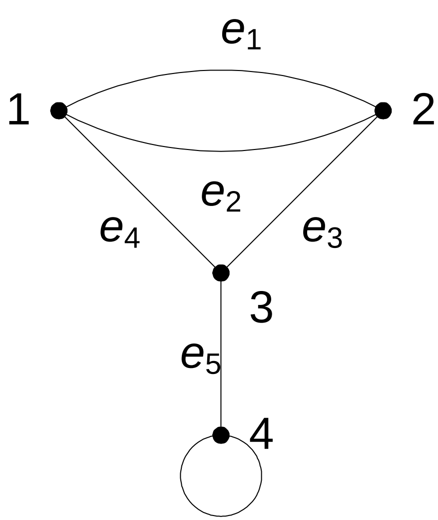
Matriks Bersisian (*incidency matrix*)

$A = [a_{ij}]$,

$$a_{ij} = \begin{cases} 1, & \text{jika simpul } i \text{ bersisian dengan sisi } j \\ 0, & \text{jika simpul } i \text{ tidak bersisian dengan sisi } j \end{cases}$$

Matriks Bersisian (*incidency matrix*)

- Graph

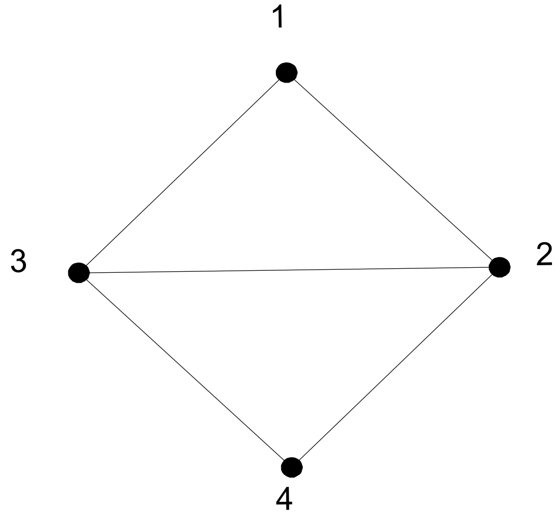


- **Matriks Bersisian**

$$
\begin{array}{cc}
 & \begin{array}{ccccc} e1 & e2 & e3 & e4 & e5 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}
\end{array}
$$

Senarai Ketetanggaan (*adjacency list*)

- Graph

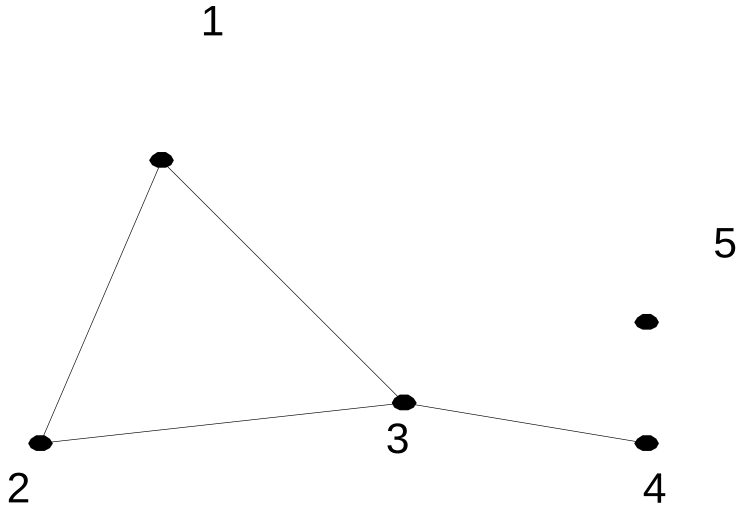


- **Senarai Ketetanggaan**

Simpul	Simpul Tetangga
1	2, 3
2	1, 3, 4
3	1, 2, 4
4	2, 3

Matriks Ketetanggaan (*adjacency matrix*)

- Graph

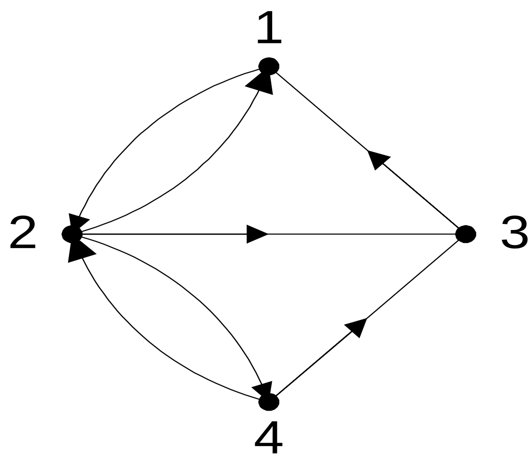


- **Senarai Ketetanggaan**

Simpul	Simpul Tetangga
1	2, 3
2	1, 3
3	1, 2, 4
4	3
5	-

Senarai Ketetanggaan (*adjacency list*)

- Graph



- **Senarai Ketetanggaan**

Simpul	Simpul Terminal
1	2
2	1, 3, 4
3	1
4	2, 3

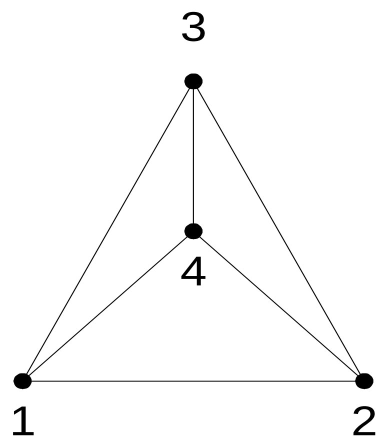
Graph Isomorfik (*Isomorphic Graph*)

- Dua buah graph yang sama tetapi secara geometri berbeda disebut graph yang saling **isomorfik**.
- Dua buah graph, *G*1 dan *G*2 dikatakan isomorfik jika terdapat korespondensi satu-satu antara simpul-simpul keduanya dan antara sisi-sisi keduaya sedemikian sehingga hubungan kebersisian tetap terjaga.

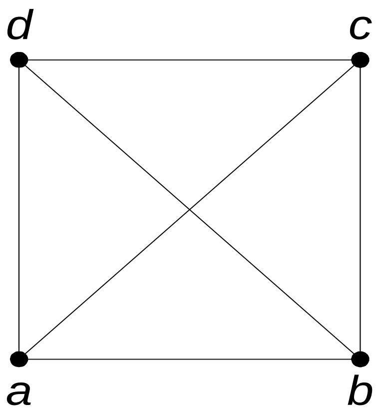
Graph Isomorfik (*Isomorphic Graph*)

- Dengan kata lain, misalkan sisi *e* bersisian dengan simpul *u* dan *v* di *G*1, maka sisi *e*' yang berkoresponden di *G*2 harus bersisian dengan simpul *u*' dan *v*' yang di *G*2.
- Dua buah graph yang isomorfik adalah graph yang sama, kecuali penamaan simpul dan sisinya saja yang berbeda. Ini benar karena sebuah graph dapat digambarkan dalam banyak cara.

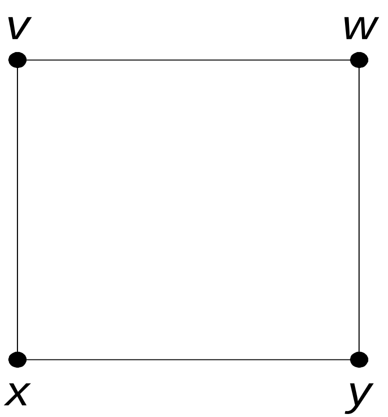
Graph Isomorfik (*Isomorphic Graph*)



(a) *G*1



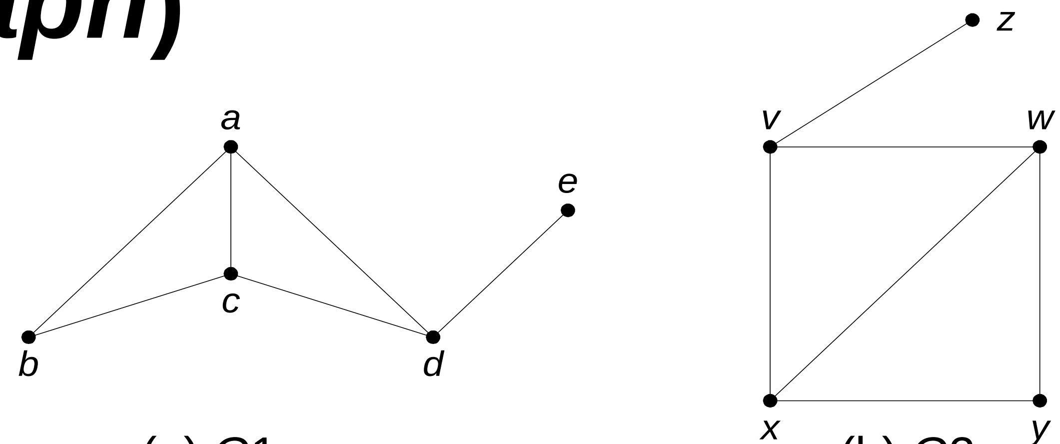
(b) *G*2



(c) *G*3

***G*1 isomorfik dengan *G*2, tetapi *G*1 tidak isomorfik dengan *G*3**

Graph Isomorfik (*Isomorphic Graph*)

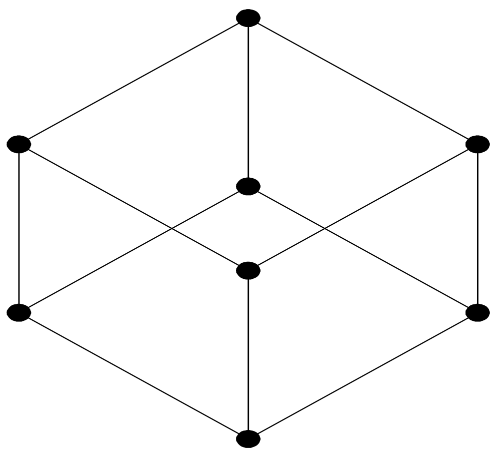


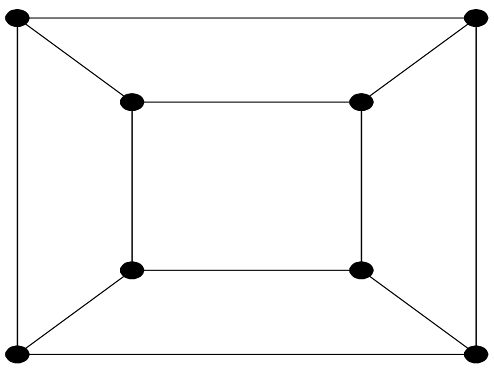
(a) *G*1 (b) *G*2

Graph (a) dan graph (b) isomorfik

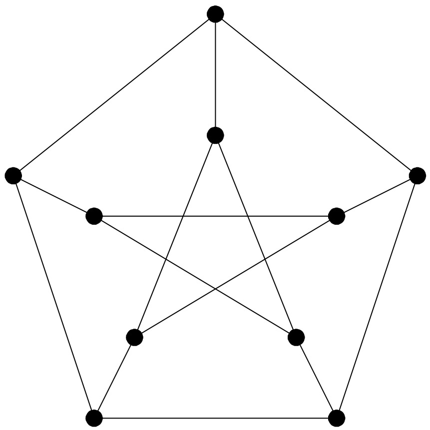
$$\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array} \qquad \begin{array}{c} \\ x \\ y \\ w \\ v \\ z \end{array} \begin{array}{c} \begin{array}{ccccc} x & y & w & v & z \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

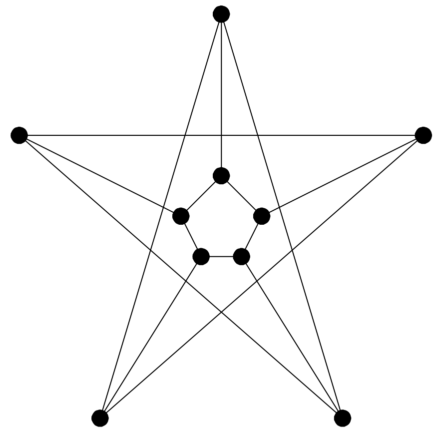
Dua buah graph isomorfik

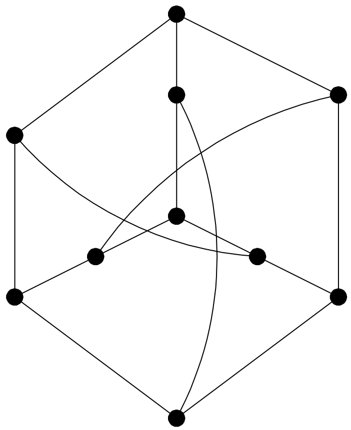




Tiga buah graph isomorfik







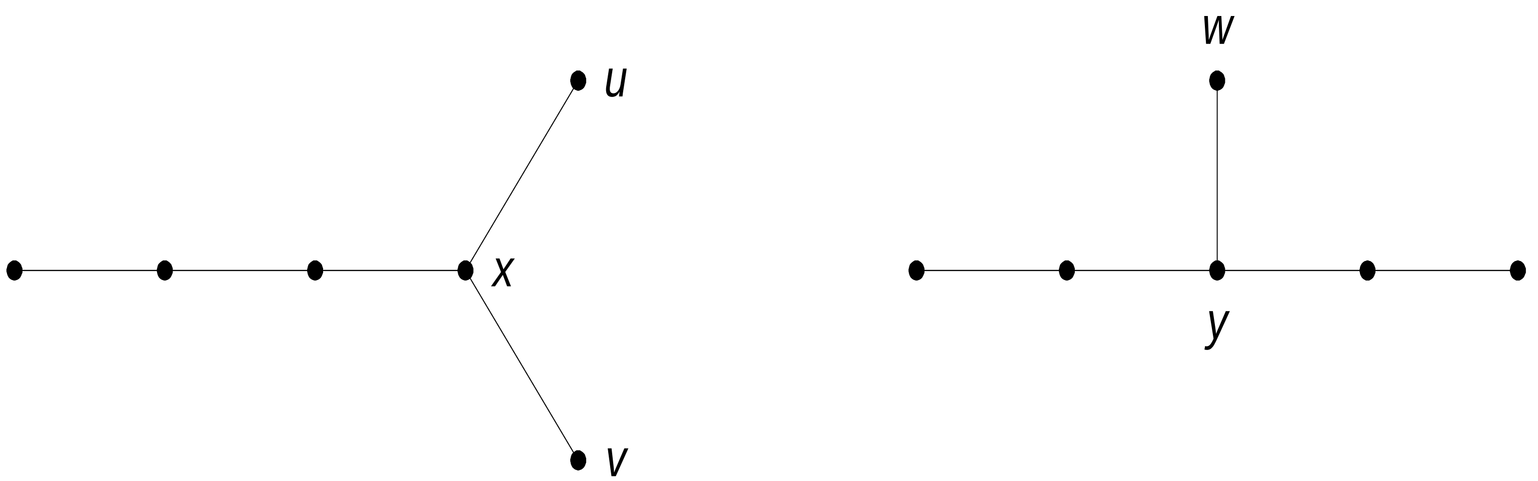
Graph Isomorfik (*Isomorphic Graph*)

Dari definisi graph isomorfik dapat dikemukakan bahwa dua buah graph isomorfik memenuhi ketiga syarat berikut [DEO74]:

1. Mempunyai jumlah simpul yang sama.
2. Mempunyai jumlah sisi yang sama
3. Mempunyai jumlah simpul yang sama berderajat tertentu

Graph Isomorfik (*Isomorphic Graph*)

Ketiga syarat ini ternyata belum cukup menjamin. Pemeriksaan secara visual perlu dilakukan.

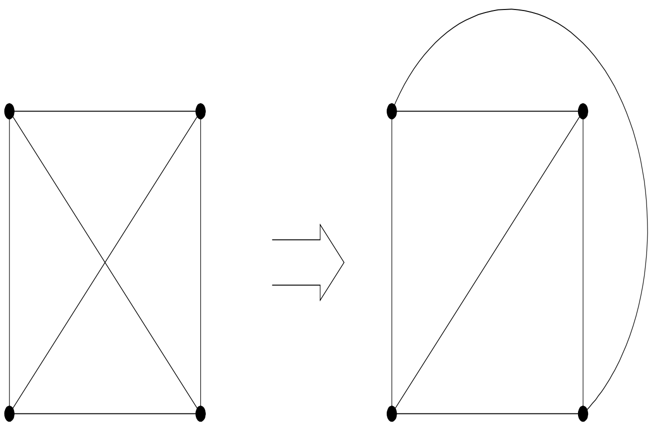


Graph Planar (*Planar Graph*) dan Graph Bidang (*Plane Graph*)

Graph yang dapat digambarkan pada bidang datar dengan sisi-sisi tidak saling memotong disebut sebagai **graph planar**, jika tidak, ia disebut **graph tak-planar**.

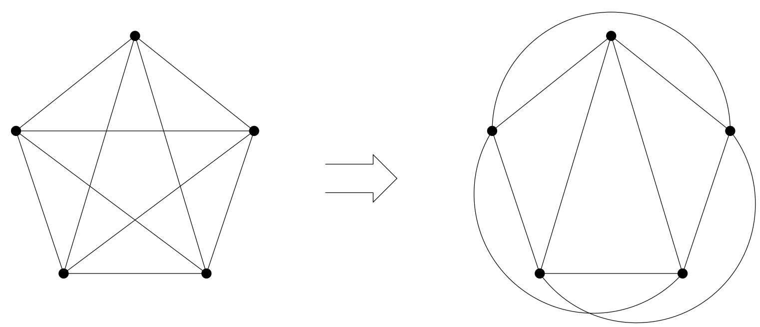
Graph Planar (*Planar Graph*)

- Graph Planar



Graph K_4

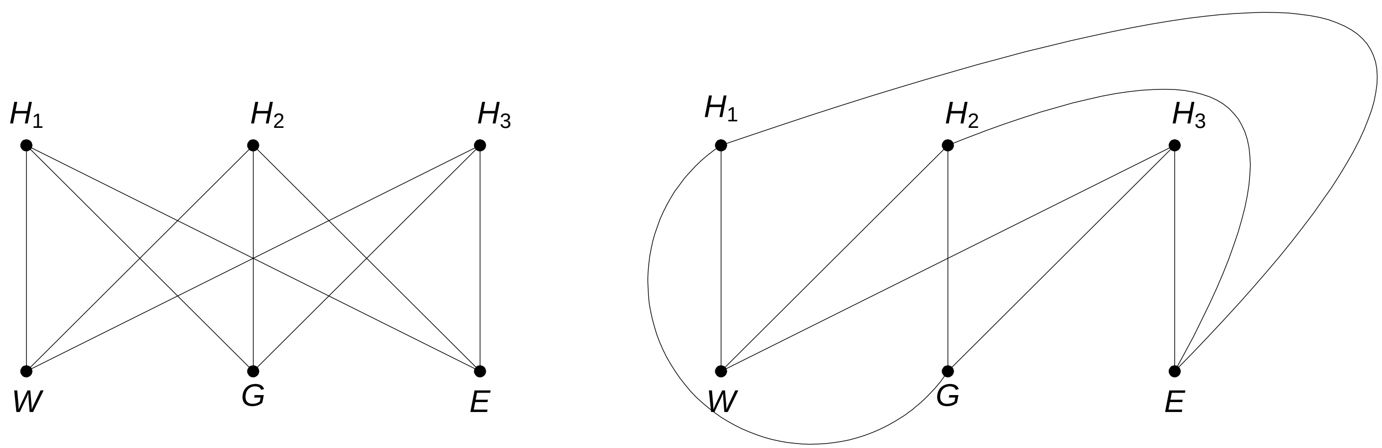
- Graph tidak planar



Graph K_5

Graph Planar (*Planar Graph*)

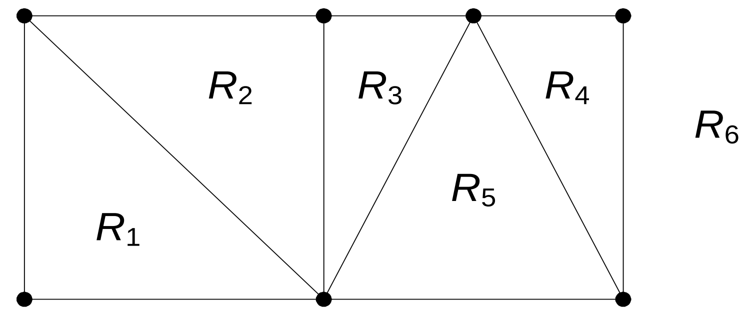
- Graph persoalan utilitas ($K_{3,3}$) bukan graph planar



Graph Planar (*Planar Graph*)

- Sisi-sisi pada graph planar membagi bidang menjadi beberapa wilayah (*region*) atau muka (*face*). Jumlah wilayah pada graph planar dapat dihitung dengan mudah.

- Graph planar yang terdiri atas 6 wilayah



Graph Planar (*Planar Graph*)

Rumus Euler

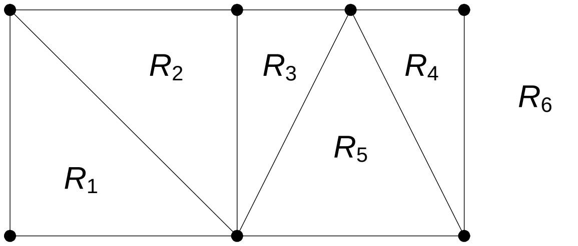
$$n - e + f = 2$$

yang dalam hal ini,

f = jumlah wilayah

e = jumlah sisi

n = jumlah simpul



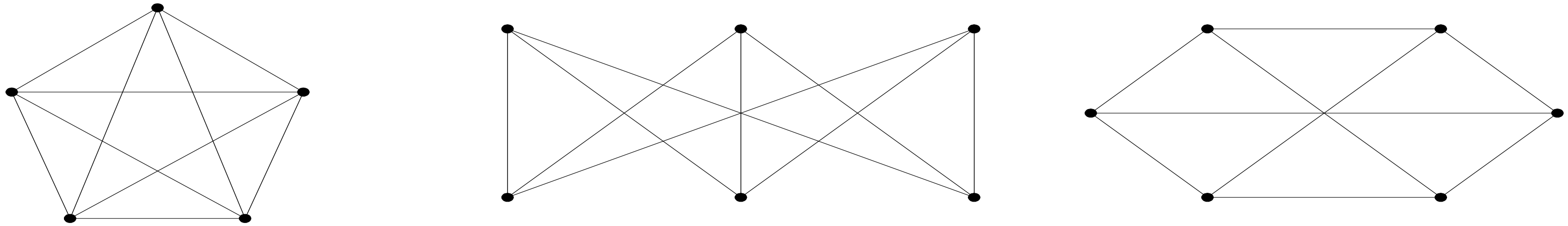
n = 11

e = 7

f = 11-7+2 = 6

Teorema Kuratoswki

- Berguna untuk menentukan dengan tegas keplanaran suatu graph.



(a) (b) (c)

(a) Graph Kuratowski pertama

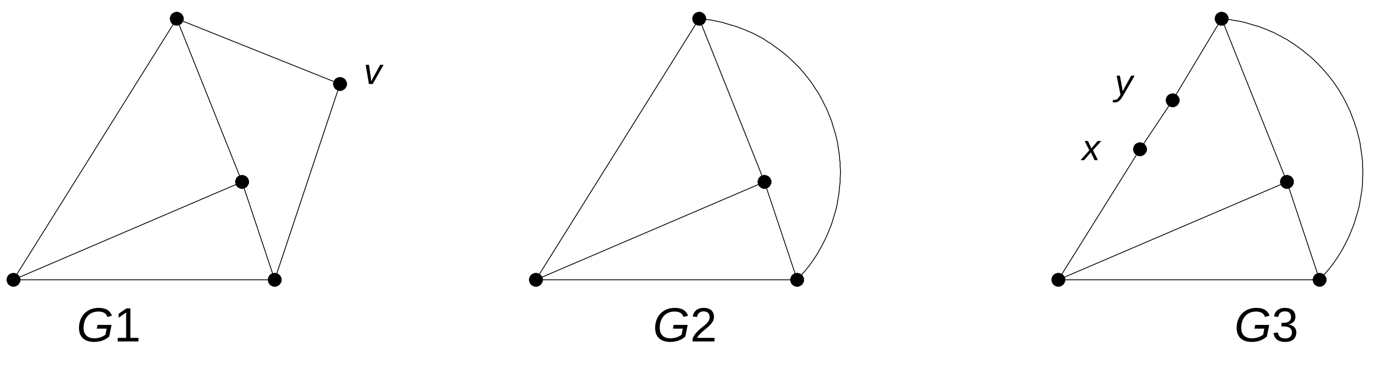
(b) dan (c) Graph Kuratowski kedua (keduanya isomorfik)

Sifat graph Kuratowski adalah:

- Kedua graph Kuratowski adalah graph teratur.
- Kedua graph Kuratowski adalah graph tidak-planar
- Penghapusan sisi atau simpul dari graph Kuratowski menyebabkannya menjadi graph planar.
- Graph Kuratowski pertama adalah graph tidak-planar dengan jumlah simpul minimum, dan graph Kuratowski kedua adalah graph tidak-planar dengan jumlah sisi minimum.

TEOREMA Kuratowski

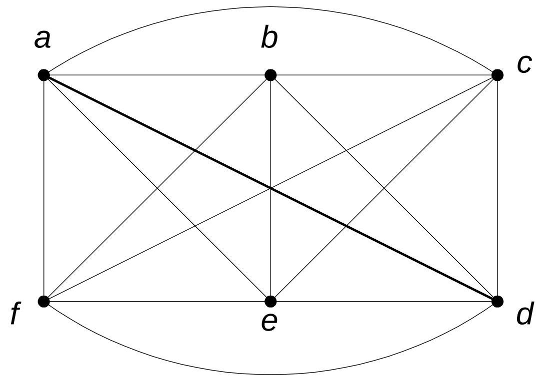
- Graph *G* bersifat planar jika dan hanya jika ia tidak mengandung upagraph yang sama dengan salah satu graph Kuratowski atau homeomorfik (*homeomorphic*) dengan salah satu dari keduanya.



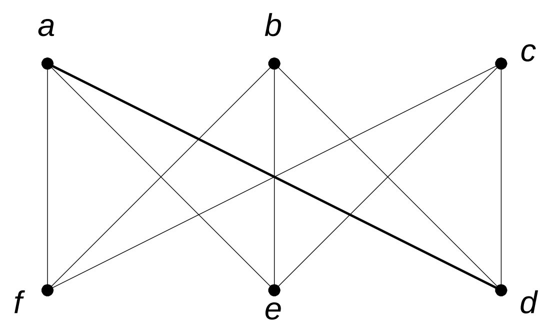
Tiga buah graph yang homemorfik satu sama lain

TEOREMA Kuratowski

- Graph di bawah ini bukan graph planar karena mengandung upagraph (G1) yang sama dengan $K_{3,3}$.



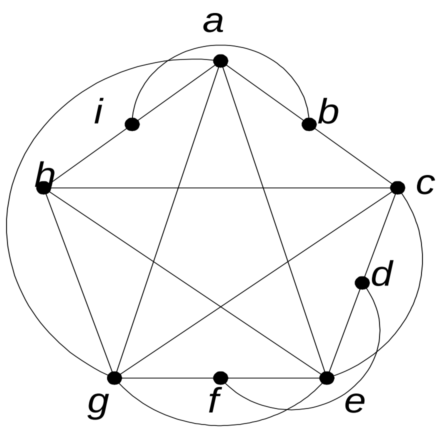
G



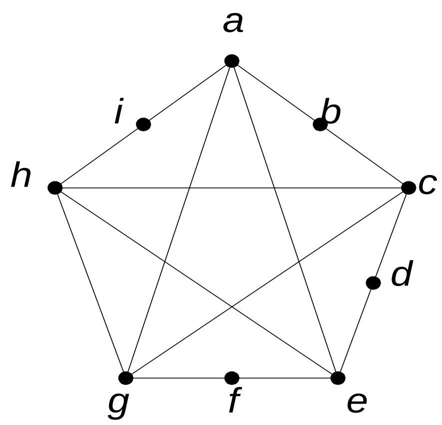
G_1

TEOREMA Kuratowski

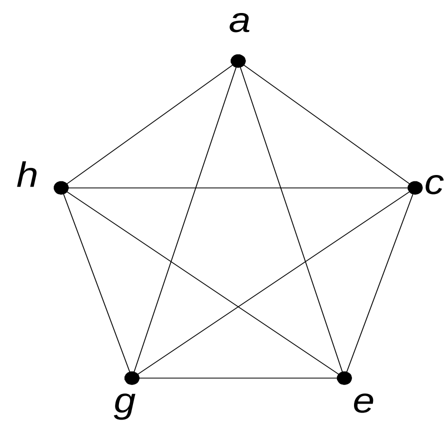
- *G* tidak planar karena mengandung upagraph (*G*1) yang homeomorfik dengan *K*5 (dengan membuang simpul-simpul yang berderajat 2 dari *G*1, diperoleh *K*5).



G



*G*1



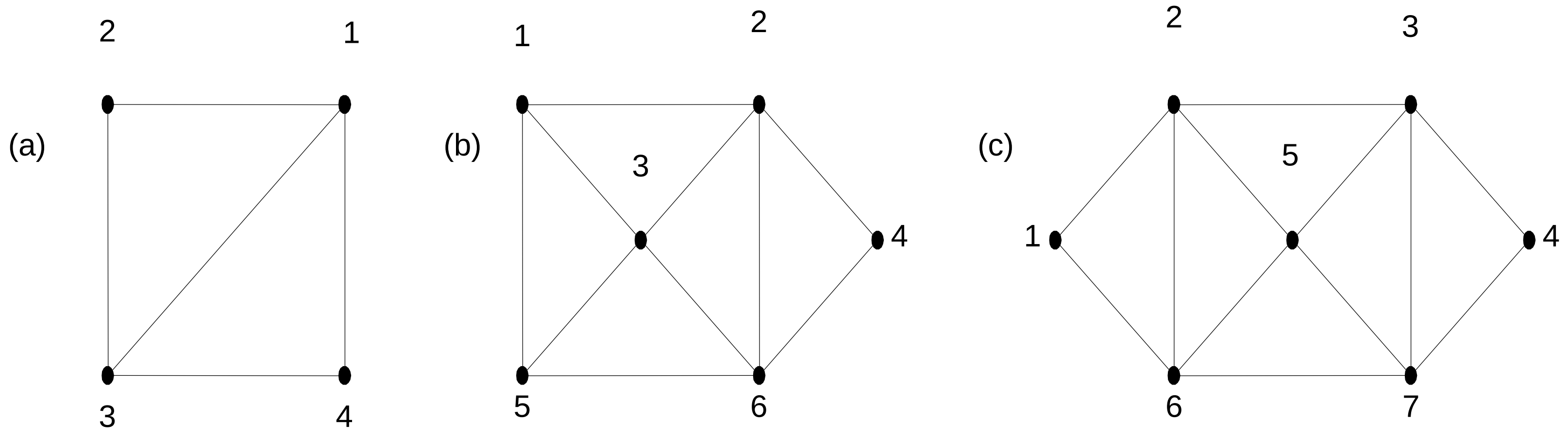
*K*5

Lintasan dan Sirkuit Euler

- **Lintasan Euler** ialah lintasan yang melalui masing-masing sisi di dalam graph tepat satu kali.
- **Sirkuit Euler** ialah sirkuit yang melewati masing-masing sisi tepat satu kali.
- Graph yang mempunyai sirkuit Euler disebut **graph Euler** (*Eulerian graph*). Graph yang mempunyai lintasan Euler dinamakan juga graph **semi-Euler** (*semi-Eulerian graph*).

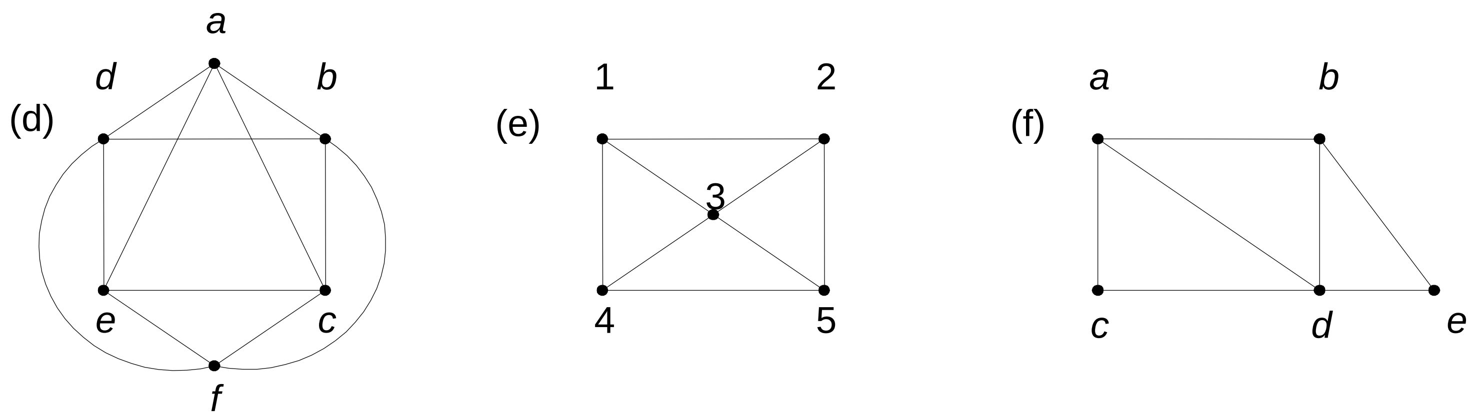
Lintasan dan Sirkuit Euler

- Lintasan Euler pada graph (a) : 3, 1, 2, 3, 4, 1
- Lintasan Euler pada graph (b) : 1, 2, 4, 6, 2, 3, 6, 5, 1, 3
- Sirkuit Euler pada graph (c) : 1, 2, 3, 4, 7, 3, 5, 7, 6, 5, 2, 6,1



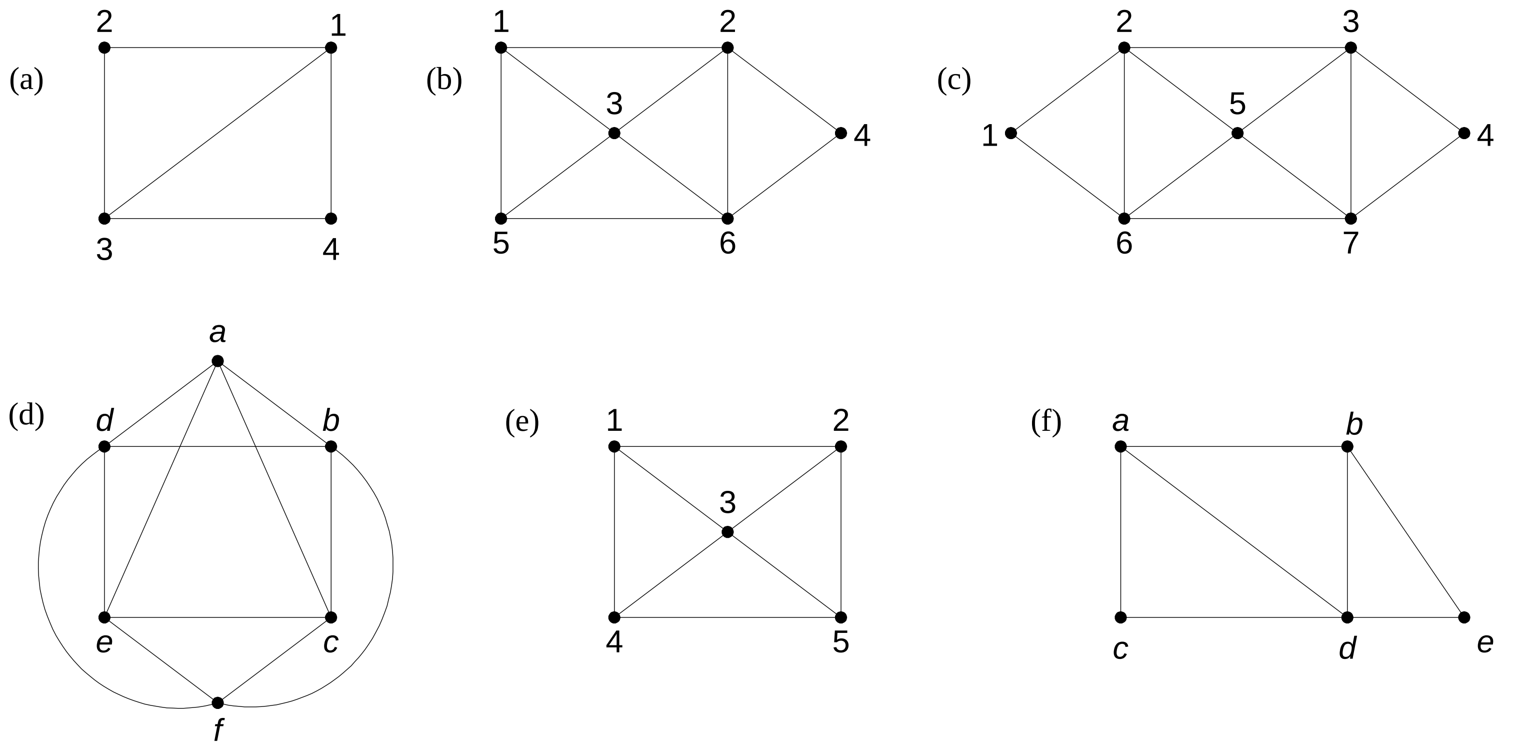
Lintasan dan Sirkuit Euler

- Sirkuit Euler pada graph (d) : *a*, *c*, *f*, *e*, *c*, *b*, *d*, *e*, *a*, *d*, *f*, *b*, *a*
- Graph (e) dan (f) tidak mempunyai lintasan maupun sirkuit Euler



Lintasan dan Sirkuit Euler

- (a) dan (b) graph semi-Euler   (c) dan (d) graph Euler
- (e) dan (f) bukan graph semi-Euler atau graph Euler



TEOREMA

- Graph tidak berarah memiliki lintasan Euler jika dan hanya jika terhubung dan memiliki dua buah simpul berderajat ganjil atau tidak ada simpul berderajat ganjil sama sekali

TEOREMA

- Graph tidak berarah *G* adalah graph Euler (memiliki sirkuit Euler) jika dan hanya jika setiap simpul berderajat genap.

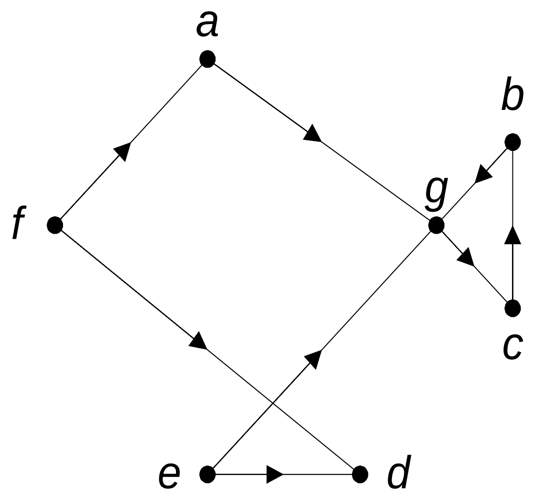
- (Catatlah bahwa graph yang memiliki sirkuit Euler pasti mempunyai lintasan Euler, tetapi tidak sebaliknya)

TEOREMA

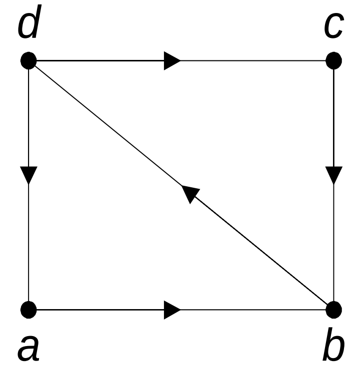
- Graph berarah *G* memiliki sirkuit Euler jika dan hanya jika *G* terhubung dan setiap simpul memiliki derajat-masuk dan derajat-keluar sama. *G* memiliki lintasan Euler jika dan hanya jika *G* terhubung dan setiap simpul memiliki derajat-masuk dan derajat-keluar sama kecuali dua simpul, yang pertama memiliki derajat-keluar satu lebih besar derajat-masuk, dan yang kedua memiliki derajat-masuk satu lebih besar dari derajat-keluar.

Lintasan dan Sirkuit Euler

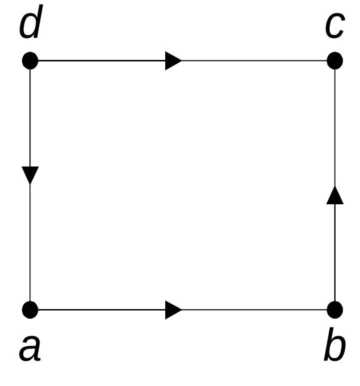
- (a) Graph berarah Euler (*a*, *g*, *c*, *b*, *g*, *e*, *d*, *f*, *a*)
- (b) Graph berarah semi-Euler (*d*, *a*, *b*, *d*, *c*, *b*)
- (c) Graph berarah bukan Euler maupun semi-Euler



(a)



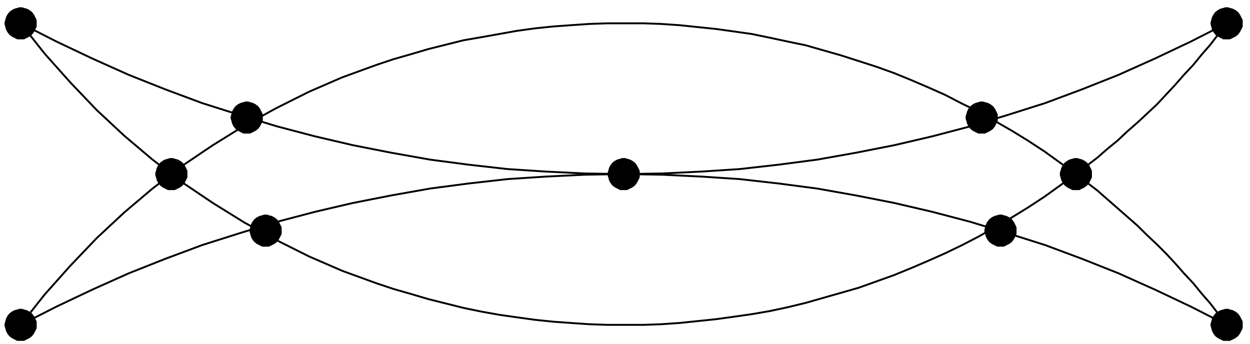
(b)



(c)

Lintasan dan Sirkuit Euler

- Bulan sabit Muhammad



Lintasan dan Sirkuit Hamilton

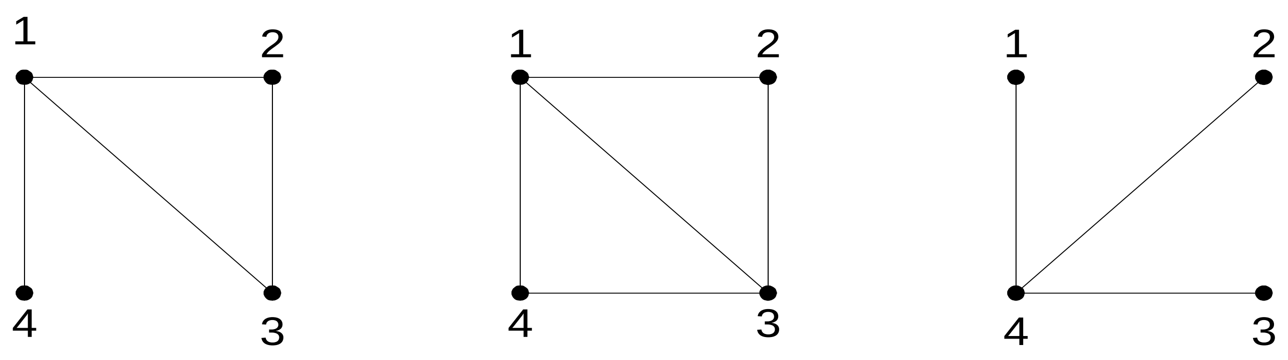
- **Lintasan Hamilton** ialah lintasan yang melalui tiap simpul di dalam graph tepat satu kali.
- **Sirkuit Hamilton** ialah sirkuit yang melalui tiap simpul di dalam graph tepat satu kali, kecuali simpul asal (sekaligus simpul akhir) yang dilalui dua kali.
- Graph yang memiliki sirkuit Hamilton dinamakan **graph Hamilton**, sedangkan graph yang hanya memiliki lintasan Hamilton disebut **graph semi-Hamilton**.

Lintasan dan Sirkuit Hamilton

(a) graph yang memiliki lintasan Hamilton (misal: 3, 2, 1, 4)

(b) graph yang memiliki lintasan Hamilton (1, 2, 3, 4, 1)

(c) graph yang tidak memiliki lintasan maupun sirkuit Hamilton

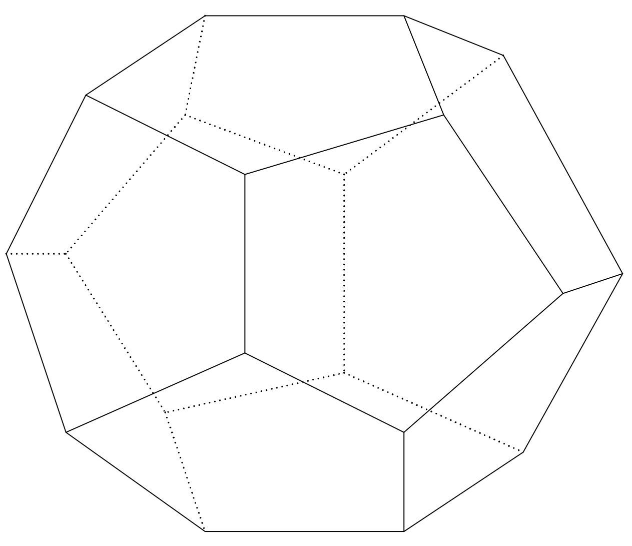


(a) (b) (c)

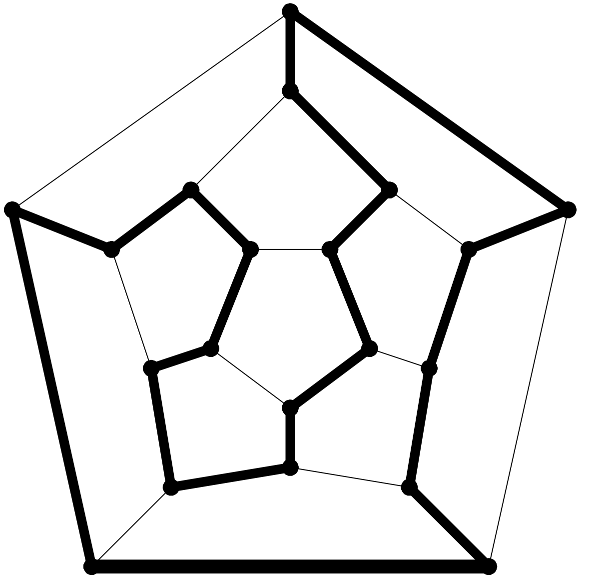
Lintasan dan Sirkuit Hamilton

(a) *Dodecahedron* Hamilton

(b) graph yang mengandung sirkuit Hamilton



(a)



(b)

TEOREMA

- Syarat cukup (jadi bukan syarat perlu) supaya graph sederhana G dengan n (≥ 3) buah simpul adalah graph Hamilton ialah bila derajat tiap simpul paling sedikit $n/2$ (yaitu, $d(v) \geq n/2$ untuk setiap simpul v di G).

TEOREMA

- Setiap graph lengkap adalah graph Hamilton
-
- Di dalam graph lengkap *G* dengan *n* buah simpul ($n \geq 3$), terdapat $(n - 1)!/2$ buah sirkuit Hamilton.

TEOREMA

- Di dalam graph lengkap *G* dengan *n* buah simpul ($n \geq 3$ dan n ganjil), terdapat

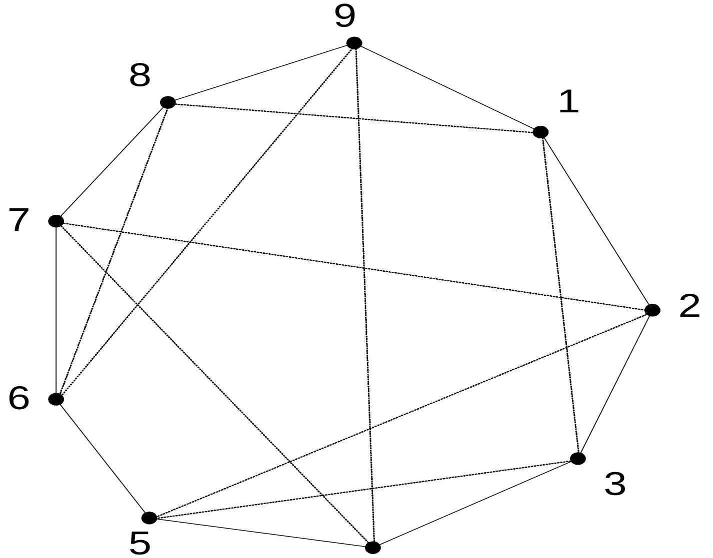
  (n - 1)/2 buah sirkuit Hamilton yang saling lepas (tidak ada sisi yang beririsan). Jika *n* genap dan $n \geq 4$, maka di dalam G terdapat (n - 2)/2 buah sirkuit Hamilton yang saling lepas.

Contoh

(Persoalan pengaturan tempat duduk). Sembilan anggota sebuah klub bertemu tiap hari untuk makan siang pada sebuah meja bundar. Mereka memutuskan duduk sedemikian sehingga setiap anggota mempunyai tetangga duduk berbeda pada setiap makan siang. Berapa hari pengaturan tersebut dapat dilaksanakan?

Jumlah pengaturan tempat duduk yang berbeda adalah (9 - 1)/2 = 4.

Lintasan dan Sirkuit Hamilton



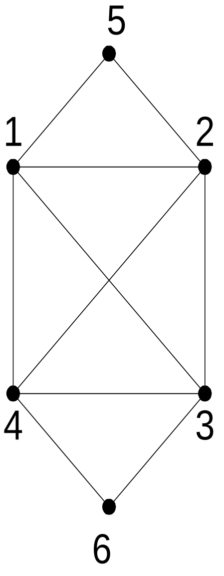
- Graph yang merepresentasikan persoalan pengaturan tempat duduk.

Lintasan dan Sirkuit Hamilton/ Euler

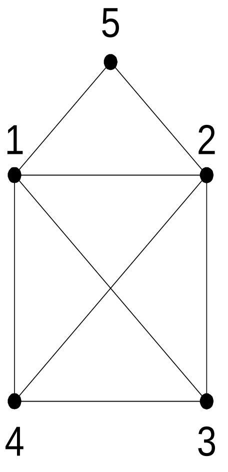
- Beberapa graph dapat mengandung sirkuit Euler dan sirkuit Hamilton sekaligus, mengandung sirkuit Euler tetapi tidak mengandung sirkuit Hamilton, mengandung sirkuit Euler dan lintasan Hamilton, mengandung lintsan Euler maupun lintasan Hamilton, tidak mengandung lintasan Euler namun mengandung sirkuit Hamilton, dan sebagainya!).

Lintasan dan Sirkuit Hamilton/ Euler

- Graph (a) mengandung sirkuit Hamilton maupun sirkuit Euler
- graph (b) mengandung sirkuit Hamilton dan lintasan Euler (periksa!).



(a)



(b)

Beberapa Aplikasi Graf

a. Lintasan Terpendek (*Shortest Path)*

- graf berbobot (*weighted graph*),
- lintasan terpendek: lintasan yang memiliki total bobot minimum.

Contoh aplikasi:

- Menentukan jarak terpendek/waktu tempuh tersingkat/ongkos termurah antara dua buah kota
- Menentukan waktu tersingkat pengiriman pesan (*message*) antara dua buah terminal pada jaringan komputer.

Lintasan Terpendek

Terdapat beberapa jenis persoalan lintasan terpendek, antara lain:

- Lintasan terpendek antara dua buah simpul tertentu.
- Lintasan terpendek antara semua pasangan simpul.
- Lintasan terpendek dari simpul tertentu ke semua simpul yang lain.
- Lintasan terpendek antara dua buah simpul yang melalui beberapa simpul tertentu.

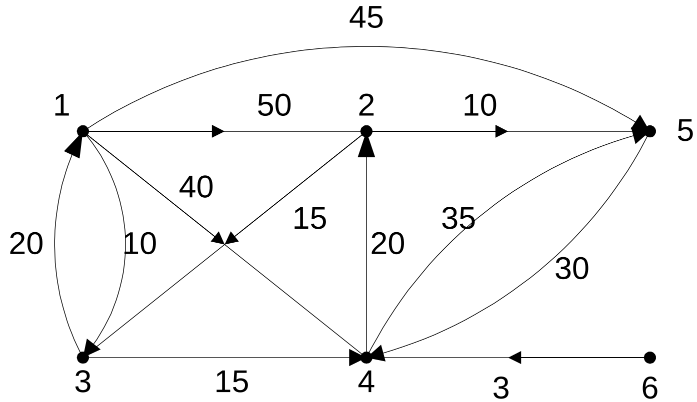
==> Di dalam kuliah ini kita memilih jenis persoalan 3.

Lintasan Terpendek

- Uraian persoalan
- Diberikan graf berbobot $G = (V, E)$ dan sebuah simpul a. Tentukan lintasan terpendek dari a ke setiap simpul lainnya di G. Asumsi yang kita buat adalah bahwa semua sisi berbobot positif.

Lintasan Terpendek

- Graph



Simpul asal	Simpul Tujuan	Lintasan terpendek	Jarak
1	3	$1 \rightarrow 3$	10
1	4	$1 \rightarrow 3 \rightarrow 4$	25
1	2	$1 \rightarrow 3 \rightarrow 4 \rightarrow 2$	45
1	5	$1 \rightarrow 5$	45
1	6	tidak ada	-

Algoritma Dijkstra

Merupakan Algoritma menentukan lintasan terpendek yang terkenal.

Properti algoritma Dijkstra:

1. Matriks ketetanggaan $M[m_{ij}]$

   m_{ij} = bobot sisi (i, j) (pada graf tak-berarah $m_{ij} = m_{ji}$)

   $m_{ii} = 0$

   $m_{ij} = \infty$, jika tidak ada sisi dari simpul i ke simpul j

2. Larik $S = [s_i]$ yang dalam hal ini,

   s_i = 1, jika simpul i termasuk ke dalam lintasan terpendek

   s_i = 0, jika simpul i tidak termasuk ke dalam lintasan terpendek

3. Larik/tabel $D = [d_i]$ yang dalam hal ini,

   d_i = panjang lintasan dari simpul awal s ke simpul i

Beberapa Aplikasi Graf

b. Persoalan Perjalanan Pedagang (*Travelling Salesperson Problem* - TSP)

- Diberikan sejumlah kota dan jarak antar kota. Tentukan sirkuit terpendek yang harus dilalui oleh seorang pedagang bila pedagang itu berangkat dari sebuah kota asal dan menyinggahi setiap kota tepat satu kali dan kembali lagi ke kota asal keberangkatan.

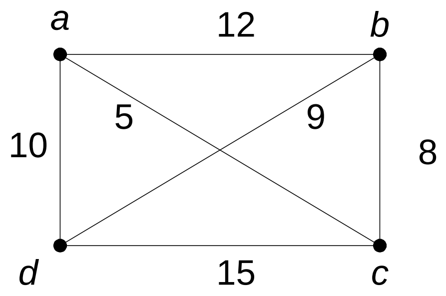
==> menentukan sirkuit Hamilton yang memiliki bobot minimum.

Aplikasi TSP

- Pak Pos mengambil surat di kotak pos yang tersebar pada *n* buah lokasi di berbagai sudut kota.
- Lengan robot mengencangkan *n* buah mur pada beberapa buah peralatan mesin dalam sebuah jalur perakitan.
- Produksi *n* komoditi berbeda dalam sebuah siklus.

Travelling Salesperson Problem

- Jumlah sirkuit Hamilton di dalam graf lengkap dengan n simpul: $(n - 1)!/2$.

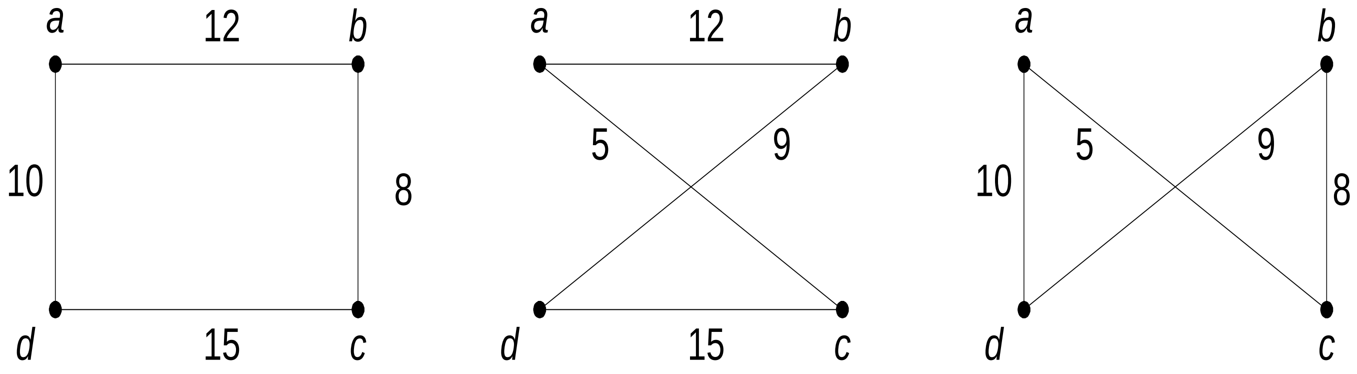


- Graf di atas memiliki (4 – 1)!/2 = 3 sirkuit Hamilton, yaitu:

I_1 = (*a*, *b*, *c*, *d*, *a*) atau (*a*, *d*, *c*, *b*, *a*) ==> panjang = 10 + 12 + 8 + 15 = 45
I_2 = (*a*, *c*, *d*, *b*, *a*) atau (*a*, *b*, *d*, *c*, *a*) ==> panjang = 12 + 5 + 9 + 15 = 41
I_3 = (*a*, *c*, *b*, *d*, a) atau (*a*, *d*, *b*, *c*, *a*) ==> panjang = 10 + 5 + 9 + 8 = 32

Travelling Salesperson Problem

- Jadi, sirkuit Hamilton terpendek adalah I3 = (*a*, *c*, *b*, *d*, *a*) atau (*a*, *d*, *b*, *c*, *a*) dengan panjang sirkuit = 10 + 5 + 9 + 8 = 32.



- Jika jumlah simpul n = 20 akan terdapat (19!)/2 sirkuit Hamilton atau sekitar 6×10^{16} penyelesaian.

Beberapa Aplikasi Graf

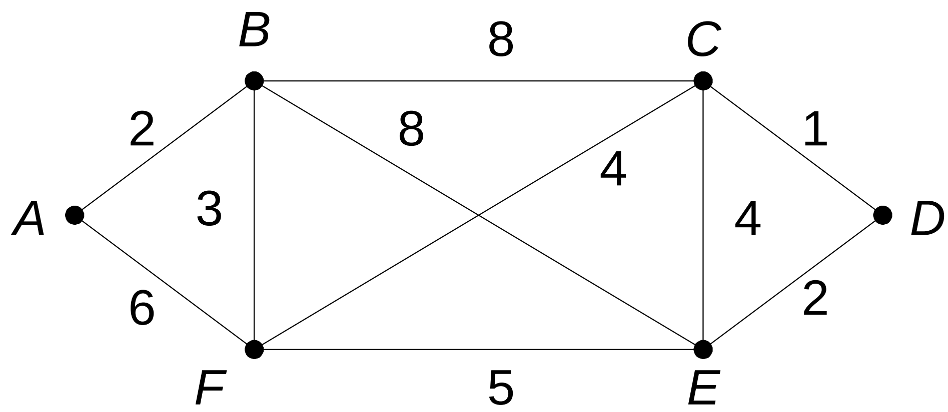
c. Persoalan Tukang Pos Cina (*Chinese Postman Problem*)

- Dikemukakan oleh Mei Gan (berasal dari Cina) pada tahun 1962.
- Masalahnya adalah sebagai berikut: *seorang tukang pos akan mengantar surat ke alamat-alamat sepanjang jalan di suatu daerah. Bagaimana ia merencanakan rute perjalanannya supaya ia melewati setiap jalan tepat sekali dan kembali lagi ke tempat awal keberangkatan.*

===> menentukan sirkuit Euler di dalam graf.

Chinese Postman Problem

- Lintasan yang dilalui tukang pos: *A*, *B*, *C*, *D*, *E*, *F*, *C*, *E*, *B*, *F*, *A*.



PEWARNAAN GRAPH

- Sebuah pewarnaan dari graph G adalah sebuah pemetaan warna-warna ke simpul-simpul dari G sedemikian hingga simpul relasinya mempunyai warna warna yang berbeda.

BILANGAN KROMATIK

- Bilangan kromatik dari G adalah jumlah warna minimum yang diperlukan untuk mewarnai graph G, dilambangkan dgn χ (G) {χ adalah huruf Yunani chi }
- Berapa bilangan kromatik dari graph lengkap K_6, K_{10} dan K_n ?

$$\chi\ (Kn) = n$$

ALGORITMA WELCH-POWELL

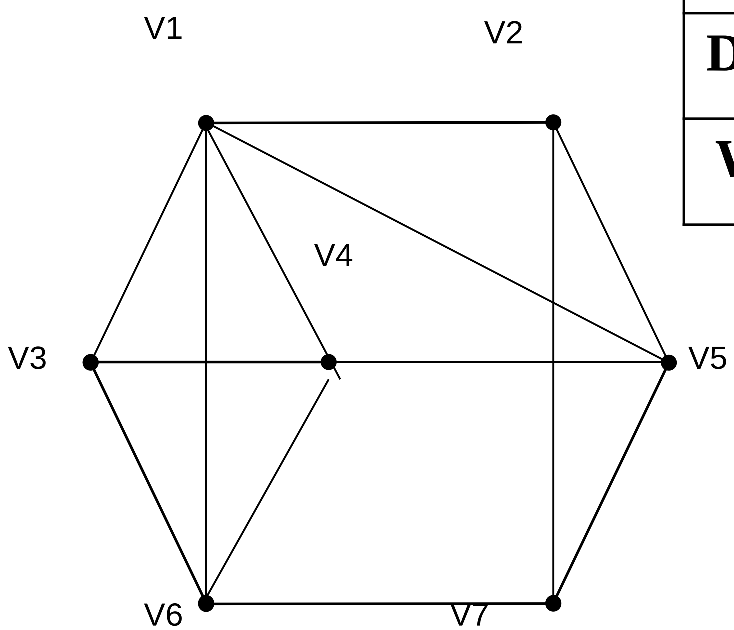
Algoritma Welch-Powell adalah sebuah cara efisien untuk mewarnai sebuah graph G

Algoritma Welch-Powell :

- Urutkan simpul-simpul G dalam derajat yang menurun. Urutan ini mungkin tidak unik karena bbrp simpul mempunyai derajat sama
- Gunakan satu warna untuk mewarnai simpul pertama dan untuk mewarnai, dalam urutan yang berurut setiap simpul dari daftar yang tidak berelasi dengan simpul sebelumnya.
- Mulai lagi dengan dengan daftar paling tinggi dan ulangi proses pewarnaan simpul yang tidak berwarna sebelumnya dengan menggunakan warna kedua.
- Terus ulangi dengan penambahan warna sampai semua simpul telah diwarnai

Contoh

Graph H

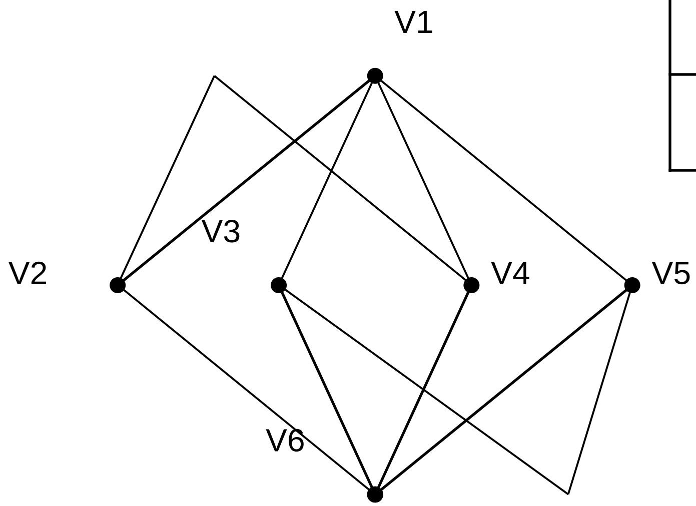


Simpul	V1	V4	V5	V6	V2	V3	V7
Derajat	**5**	**4**	**4**	**4**	**3**	**3**	**3**
Warna	**a**	**b**	**c**	**d**	**b**	**c**	**a**

Jadi $\chi(H) = 4$

Contoh

- Graph G

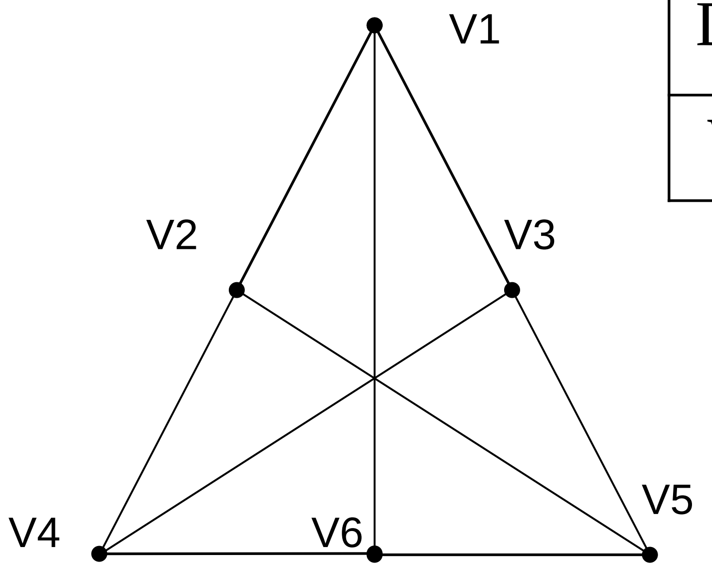


Simpul	V1	V6	V2	V3	V4	V5
Derajat	4	4	3	3	3	3
Warna	a	a	b	b	c	c

Jadi $\chi(G) = 3$

Contoh

- Graph H

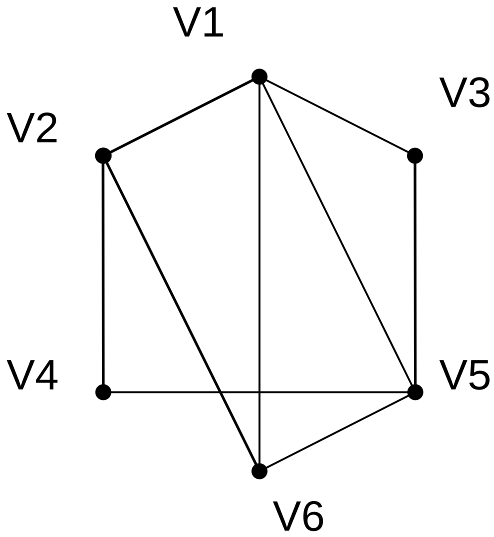


Simpul	V1	V2	V3	V4	V5	V6
Derajat	3	3	3	3	3	3
Warna	a	b	b	a	a	b

Jadi $\chi(H) = 2$

Contoh

- Graph G

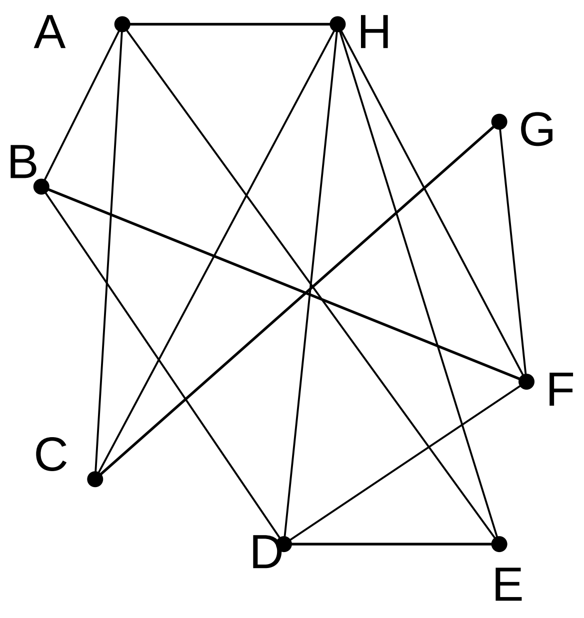


Simpul	V1	V5	V2	V6	V3	V4
Derajat	4	4	3	3	2	2
Warna	a	b	b	c	c	a

Jadi $\chi(G) = 3$

Contoh

- Graph H



Simpul	H	A	D	F	B	C	E	G
Derajat	5	4	4	4	3	3	3	2
Warna	a	b	b	c	a	c	c	a

Jadi $\chi(H) = 3$

Contoh

- Adakah graph dengan 1 warna????