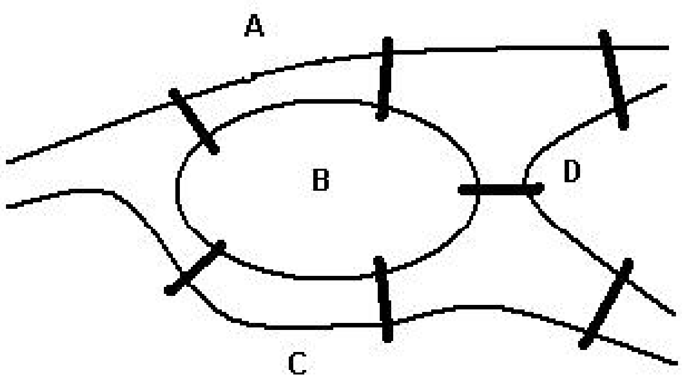
# Matematika Diskrit 2

*Teori Graph*

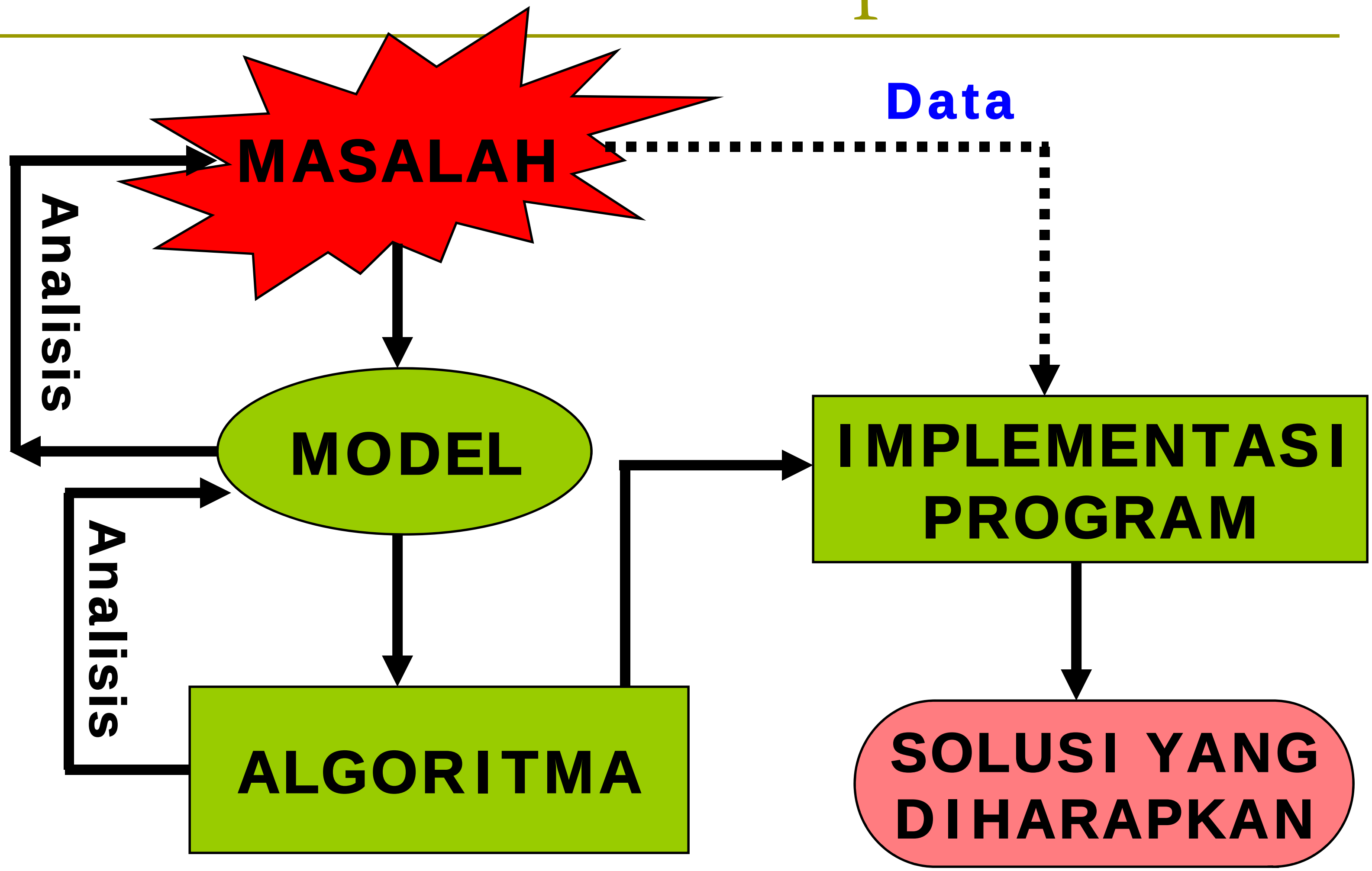
# Kelahiran Teori Graph

- Masalah Jembatan Konigsberg :
  - Mulai dan berakhir pada tempat yang sama, bagaimana caranya untuk melalui setiap jembatan tepat satu kali ?




- 1736: **Leonhard Euler**
  - Basel, 1707-St. Petersburg, 1786
  - Mampu mengungkap misteri Jembatan Konigsberg

# Problem dan Model Graph



# Problem 1

- Setiap  minggu sekali, seorang petugas kantor telepon berkeliling untuk mengumpulkan koin pada telepon umum yang dipasang diberbagai tempat. Berangkat dari kantornya, ia mendatangi satu demi satu telepon umum tersebut, dan akhirnya kembali ke kantor lagi. Problem yang muncul adalah petugas tersebut menginginkan suatu rute perjalanan dengan waktu minimal ?

# Problem 2

- Pada suatu persimpangan jalan yang ramai akan dipasang lampu lalu lintas (TL). Telah diatur bahwa jalan A, C, D, E, dan F satu arah serta jalan B adalah 2 arah. Perjalanan yang diperbolehkan adalah :
  - A → B A → C A → E B → C B → E
  - D → C D → E F → B F → C F → E

  Problemnya adalah bagaimana menentukan pola TL dengan jumlah fase minimal,dan pada setiap fase tidak ada perjalanan yang saling melintas ?

# Problem 3

- Rute perjalanan dari kota A ke P dapat dilakukan dengan berbagai macam alternatif. Dari sekian banyak alternatif yang ada maka tentukanlah rute yang paling minimal untuk ditempuh (misalkan minimal dalam hal jarak tempuh/waktu tempuh) ?

# Model Graph

- Jika kita lakukan analisis terhadap ketiga problem tadi, maka kita akan buatkan model persoalannya ke dalam model Graph.
- Problem 1 pada model Graph dikenal dengan problem ***Travelling Salesman*.**
- Problem 2 pada model Graph dikenal dengan problem ***Coloring Graph* (pewarnaan Graph)**.
- Problem 3 pada model Graph dikenal dengan problem ***Shortest Path.***

# Pendahuluan

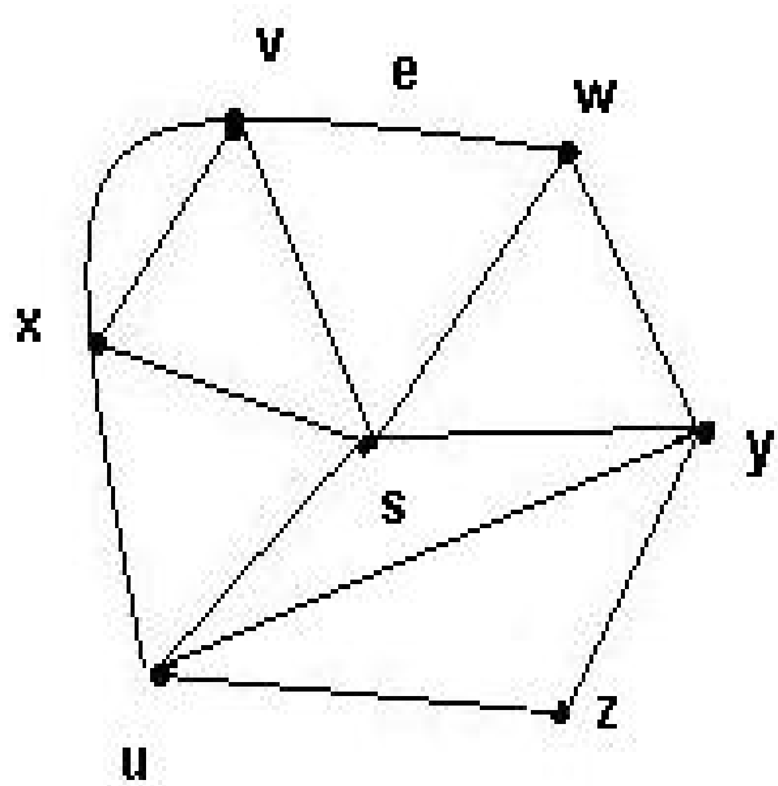
**Definisi 1 :**

- Suatu Graph G = (V,E) adalah koleksi atau pasangan dari dua himpunan V (tidak kosong) dan E dengan
  - V = V(G) = himpunan verteks atau simpul atau node.
  - E = E(G) = himpunan edge atau ruas atau sisi.
- Banyaknya verteks disebut order
- Banyaknya edge disebut size (ukuran)

# Pendahuluan *(Lanjutan)*

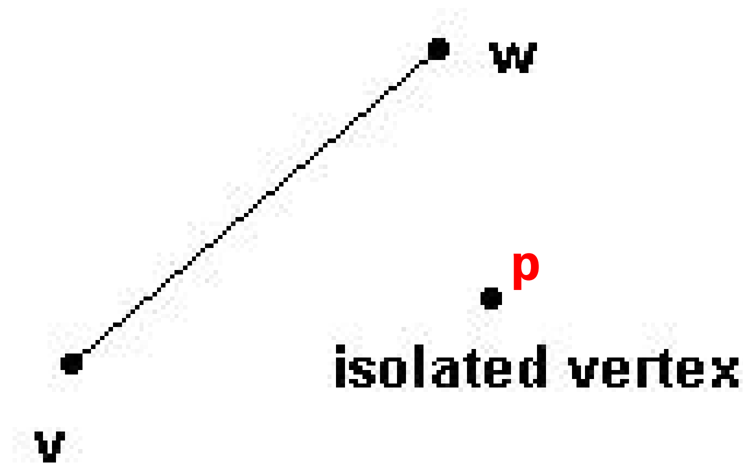
## Contoh 1 :

- V = {s, u, v, w, x, y, z}
- E = {(x,s), $(x,v)_1$, $(x,v)_2$, (x,u), (v,w), (s,v), (s,u), (s,w), (s,y), (w,y), (u,y), (u,z),(y,z)}



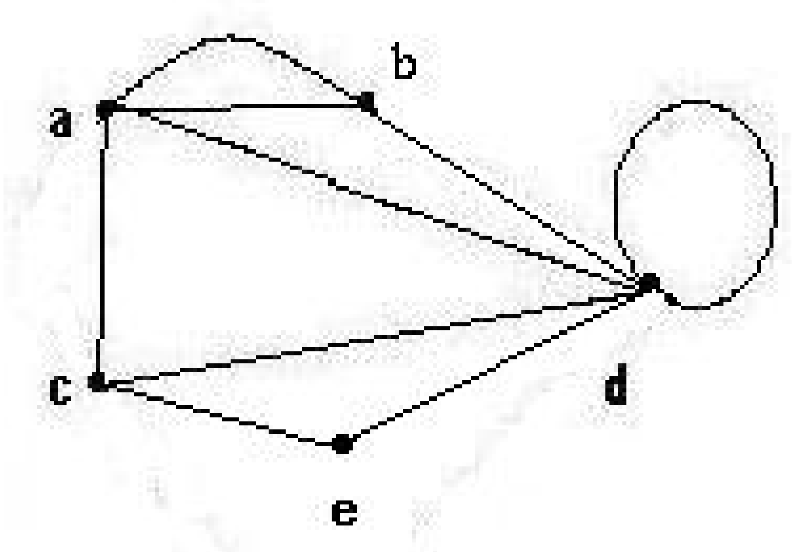
# Edge

- Edge merupakan pasangan tak terurut dari simpul. Misalkan edge $e$ = (v,w) = (w,v).
- Edge $e$ dikatakan *incident* pada v dan w.
- Verteks terpencil (terisolasi) adalah suatu verteks tanpa edge *incident*.



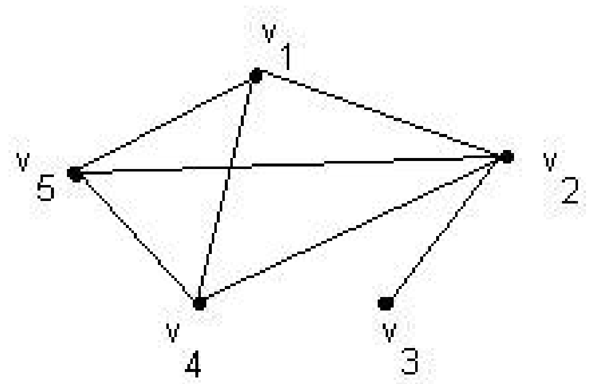
# Edge Khusus

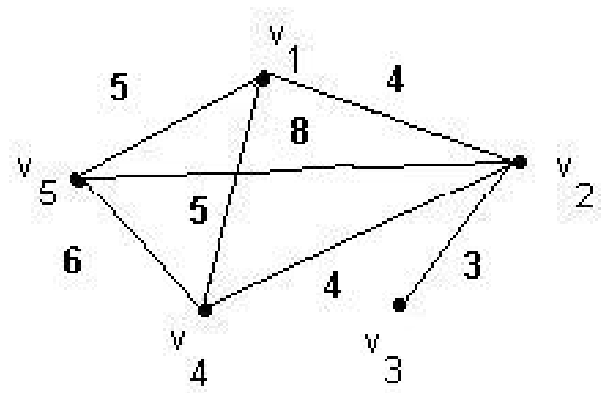
- Edge Paralel
  - Dua edge atau lebih yang mempunyai kedua verteks ujung yang sama.
    - Graph disamping : edge **(a,b)** merupakan edge paralel atau edge sejajar.
- Loop (self-loops)
  - Suatu edge yang kedua verteks ujungnya sama.
    - Graph disamping, edge **(d,d)** → ***self-loops***.



# Graph Khusus

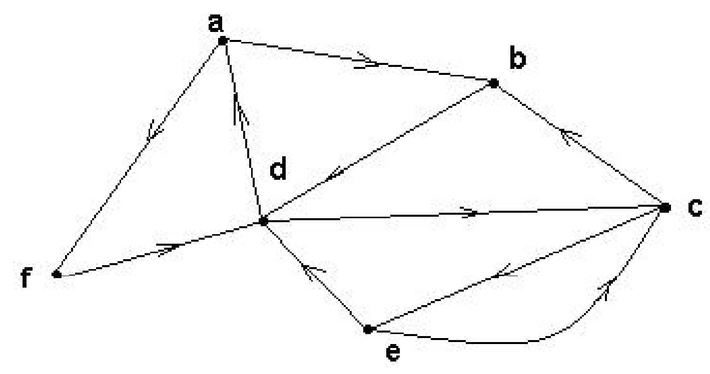
- Simple graph (Graph sederhana)
  - Suatu graph yang tidak memiliki self-loops dan ruas sejajar.
- Weighted graph (Graph berlabel / berbobot)
  - Suatu graph yang setiap ruasnya dikaitkan dengan besaran tertentu (“bobot”).



# Graph Berarah (Digraph)

G disebut graph berarah atau *directed graph/ digraph* jika setiap ruas merupakan pasangan terurut dari simpul. (dpl. Setiap ruasnya memiliki arah).



# Graph Berarah (Digraph)

Definisi:

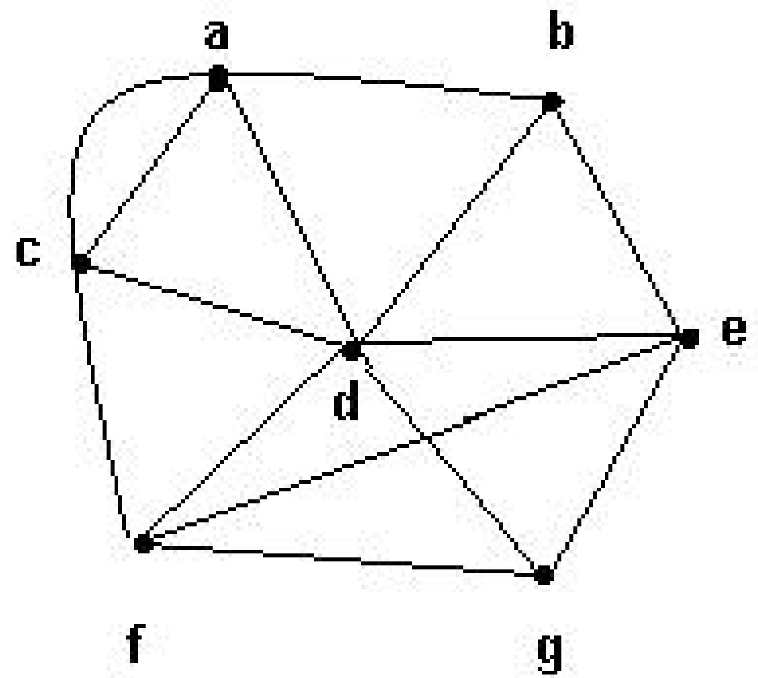
Jika (u,v) adalah edge dari graph berarah G, u dikatakan adjacent ke v dan v dikatakan adjacent dari u.

Vertex u dikatakan sebagai verteks inisial dari (u,v) dan v dikatakan sebagai verteks terminal atau verteks akhir (u,v).

verteks inisial dan verteks terminal dari sebuah loop adalah sama

# Derajat Verteks

- Derajat dari simpul v, dinotasikan dgn δ(v), adalah banyaknya ruas yang melalui v
- Contoh :
  - δ(a) = 4, δ(b) = 3,
  - δ(c) = 4, δ(d) = 6,
  - δ(e) = 4, δ(f) = 4,
  - δ(g) = 3.



# Derajat pada Graph

Teorema (The Handshaking Theorem):

jika G suatu graph tidak berarah dengan m edge dan n verteks maka jumlah derajat semua verteks adalah 2m.

$$\sum_{i=1}^{n} \delta(v_i) = 2m$$

→ jumlah dari derajat semua verteks pada graph tidak berarah adalah genap.

# Derajat pada Graph

Teorema :

Sebuah graph tidak berarah G memiliki sejumlah genap vertek berderajat ganjil

# Derajat pada Graph

Definisi:

Sebuah graph berarah G memiliki derajat masuk dari sebuah verteks (*in-degree of a vertex*) v, dinotasikan sebagai $\delta^-(v)$, menyatakan banyaknya edge dengan v sebagai verteks terminal. Derajat keluar dari sebuah verteks (*out-degree of a vertex*) v, dinotasikan sebagai $\delta^+(v)$, menyatakan banyaknya edge dengan v sebagai verteks inisial.

# Graph Berarah (Digraph)

$\delta^-(a) = 1$, $\delta^+(a) = 2$,

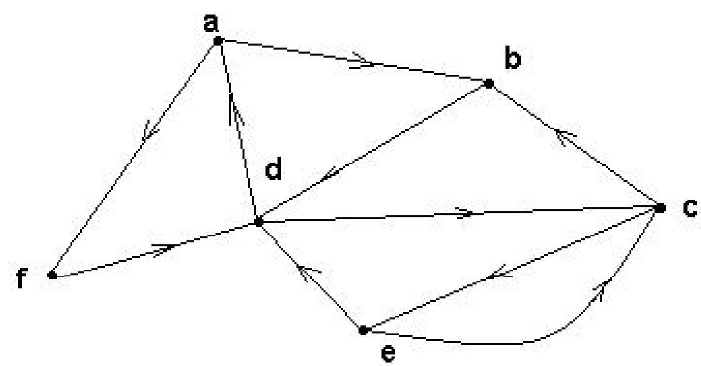
$\delta^-(b) = 2$, $\delta^+(b) = 1$,

$\delta^-(c) = 2$, $\delta^+(c) = 2$,

$\delta^-(d) = 3$, $\delta^+(d) = 2$,

$\delta^-(e) = 1$, $\delta^+(e) = 2$,

$\delta^-(f) = 1$, $\delta^+(f) = 1$



$a_0, a_1, \cdots, a_k$

# Derajat pada Graph

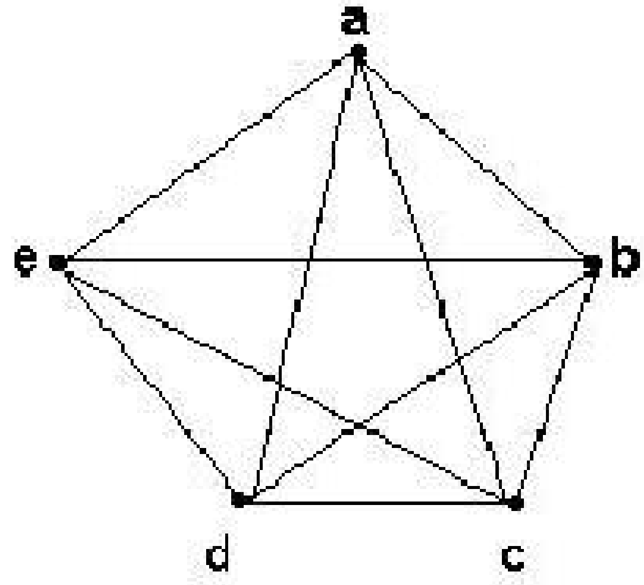
Teorema :

Misalkan G = (V,E) adalah graph berarah maka:

$$\sum_{v \in V} \delta^{-}(v) = \sum_{v \in V} \delta^{+}(v) = |E|$$

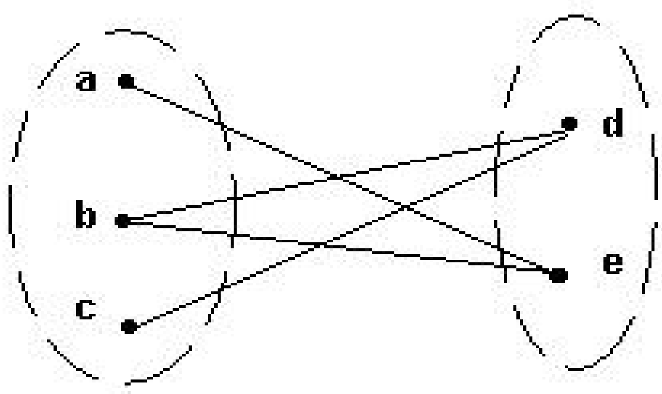
# Graph Lengkap $K_n$

- Misalkan $n \geq 3$
- Graph Lengkap (*complete graph) $K_n$* adalah graph dengan n simpul dan setiap pasang simpulnya terhubung oleh satu ruas. Derajat setiap vertex sama
- Contoh di samping merupakan Graph lengkap $K_5$

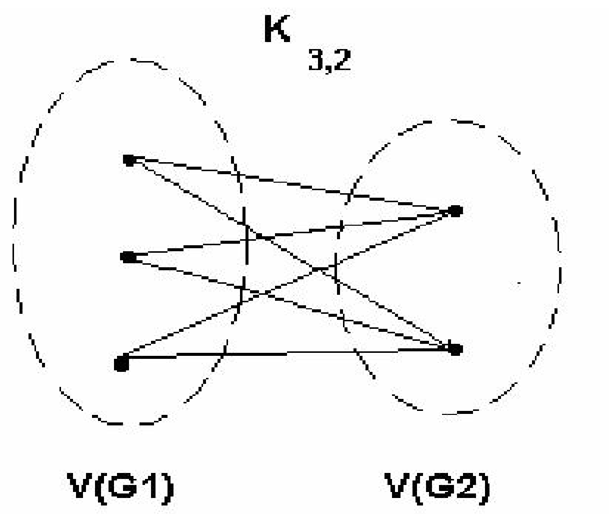


# Graph Bipartisi

- Graph bipartisi G adalah suatu graph sedemikian sehingga berlaku
  - $V(G) = V(G_1) \cup V(G_2)$
  - $|V(G_1)| = m$, $|V(G_2)| = n$
  - $V(G_1) \cap V(G_2) = \varnothing$
  - Tidak terdapat edge antara sembarang verteks pada subset $V(G_k)$ yang sama; $k = 1,2$.



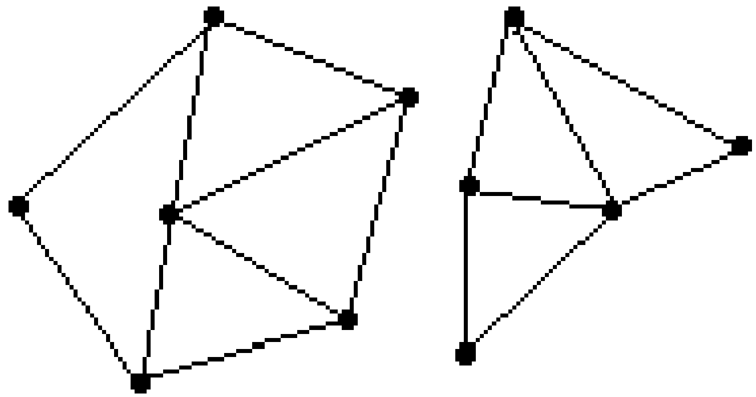
# Complete bipartite graph $K_{m,n}$




- Suatu graph bipartisi adalah graph bipartisi lengkap *(Complete bipartite graph)* $K_{m,n}$ jika setiap simpul pada $V(G_1)$ terhubung dengan simpul pada $V(G_2)$ dan sebaliknya,
- $|V(G_1)| = m$
- $|V(G_2)| = n$

# Graph Terhubung

- Suatu Graph dikatakan terhubung *(Connected)* jika setiap pasang dari verteks dapat dilalui dengan suatu jalur.



2 connected components

- Setiap subgraph terhubung dari suatu graph tak terhubung G disebut *component dari* G

# Jalur(Path) dan *Cycle*

- Suatu Jalur *(Path)* dengan panjang *n* adalah barisan dari n + 1 verteks dan n edge secara berurutan.

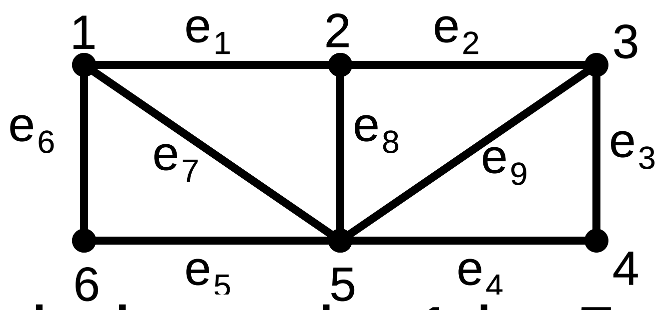
  → $(v_0, e_1, v_1, e_2, v_2, e_3, \ldots, v_{n-1}, e_n, v_n)$

- Suatu C*ycle* adalah jalur dengan verteks awal dan verteks akhirnya sama.

# Jalur (Path) dan *Cycle* *(Lanjutan)*

Contoh :

- Diketahui suatu Graph G :




- Jalur dari verteks 1 ke 5 : 1, 5 atau 1, 2, 5 atau 1, 2, 3, 4, 5 atau 1, 2, 3, 5, atau

  1, 6, 5
- *Cycle* dgn panjang 3 : 1, 2, 5, 1 atau 2, 3, 5, 2
- *Cycle* dgn panjang 6 : 1, 2, 3, 4, 5, 6, 1

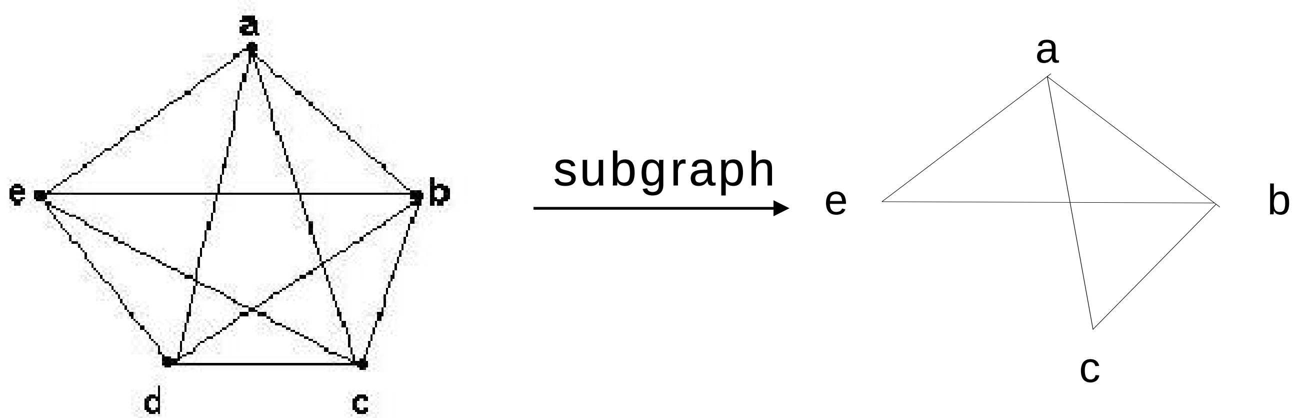
# Subgraph

Definisi :

- Misal G=(V,E) suatu Graph, G' =(V',E') disebut *subgraph* dari G jika :
  - $V' \subseteq V$ dan $E' \subseteq E$

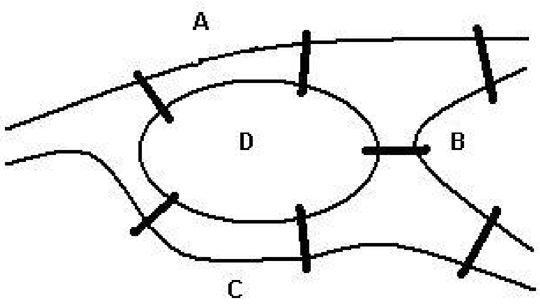
Contoh:

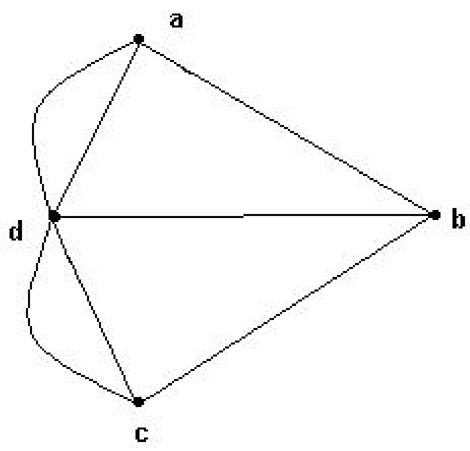
- Diketahui graph G sebagai berikut :



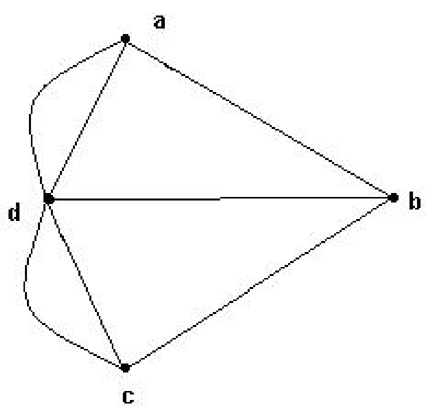
# Perjalanan Euler

- Sebuah perjalanan Euler *(Euler cycle)* pada graph G adalah sebuah *cycle* sederhana yang melalui setiap edge di G hanya sekali.
- Problem jembatan Königsberg:
  - Apakah memungkinkan untuk memulai dan mengakhiri suatu perjalanan dari titik yang sama melalui ke 7 jembatan hanya sekali?
- Problem dapat dinyatakan dengan sebuah graph
- Edge menyatakan jembatan dan setiap verteks menyatakan daerah *(region)*.



# Graph Euler




- Sebuah graph G adalah graph *Euler* jika memiliki *Euler cycle*.

Teorema:

G adalah Graph Euler jika dan hanya jika G terhubung dan semua vertex memiliki derajat genap.

- Graph terhubung merepresentasikan problem jembatan KÖnigsberg.
- Graph tersebut bukan Graph Euler.
- Berarti problem jembatan KÖnigsberg tidak memiliki solusi.