

# Pertemuan 12

## GRAPH, MATRIK PENYAJIAN GRAPH



# GRAPH

Suatu Graph mengandung 2 himpunan, yaitu :

1. Himpunan  $V$  yang elemennya disebut simpul (Vertex atau Point atau Node atau Titik)
2. Himpunan  $E$  yang merupakan pasangan tak urut dari simpul. Anggotanya disebut Ruas (Edge atau rusuk atau sisi)

Graph seperti dimaksud diatas, ditulis sebagai  $G(E,V)$ .

Banyaknya simpul (vertex) disebut **Order**, sedangkan banyaknya ruas (edge) disebut **Size** dari Graph.

## Contoh :

Gambar berikut menanyakan Graph  $G(E,V)$  dengan :

1.  $V$  mengandung 4 simpul, yaitu simpul  $A, B, C, D$ .
2.  $E$  mengandung 5 ruas, yaitu :

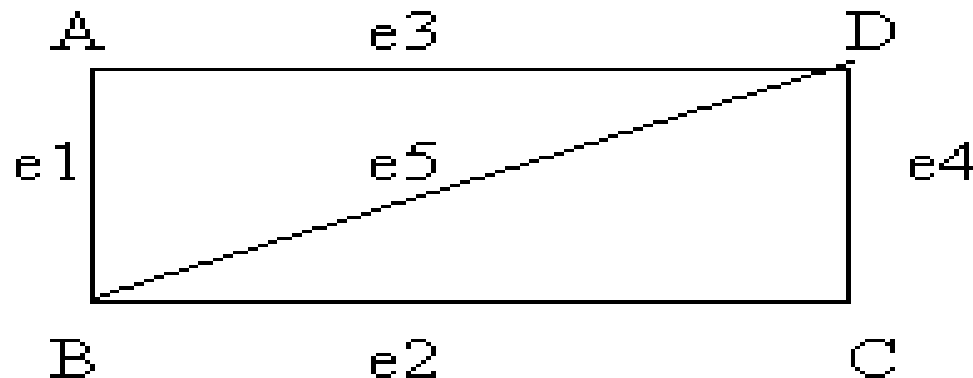
$$e1 = (A,B)$$

$$e4 = (C,D)$$

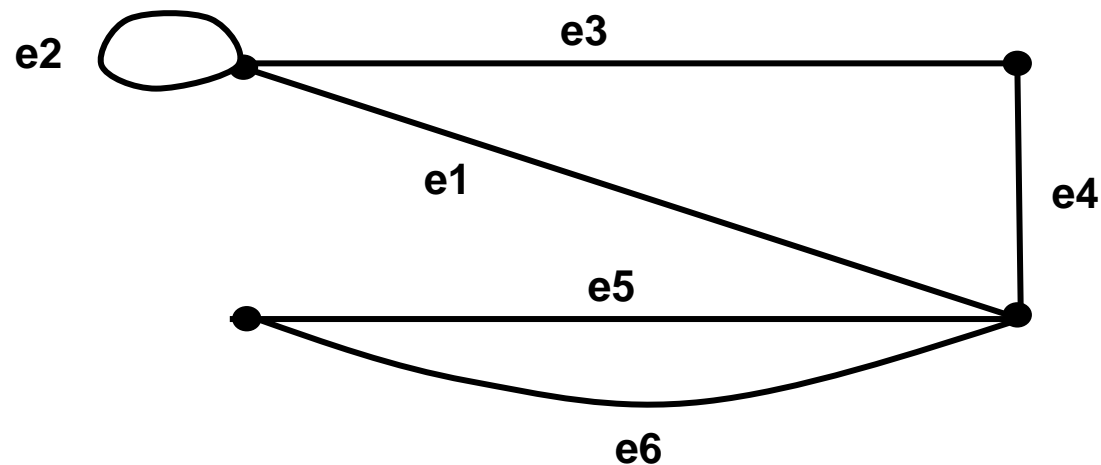
$$e2 = (B,C)$$

$$e5 = (B,D)$$

$$e3 = (A,D)$$



Gambar dibawah ini menyatakan suatu **Multigraph**. Disini, ruas e2 pada kedua titik ujungnya adalah simpul yang sama, yaitu simpul A. Ruas semacam ini disebut **Gelung atau Self-Loop**. Sedangkan ruas e5 dan e6 mempunyai titik ujung yang sama, yaitu simpul-simpul B dan C. Kedua ruas ini disebut ruas **berganda** atau ruas **sejajar**.

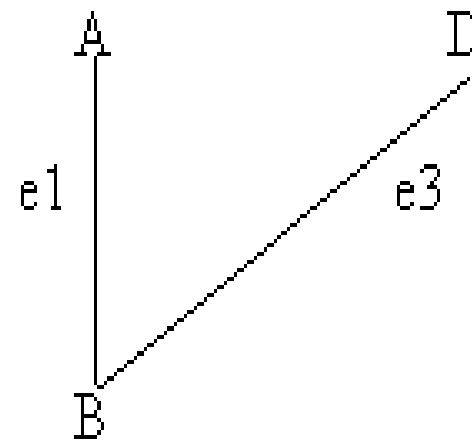
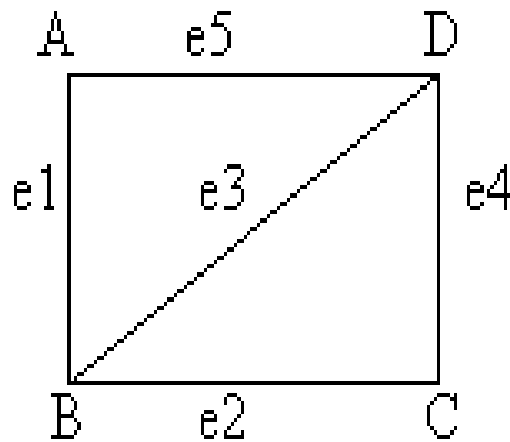


Suatu Graph yang tidak mengandung ruas sejajar maupun self-loop, sering disebut juga sebagai **Graph sederhana** atau **simple Graph**.

Suatu Graph  $G'(E', V')$  disebut Sub Graph dari  $G(E, V)$ , bila  $E'$  himpunan bagian dari  $E$  dan  $V'$  himpunan bagian dari  $V$ .

Jika  $E'$  mengandung semua ruas dari  $E$  yang titik ujungnya di  $V'$ , maka  $G'$  disebut Subgraph yang direntang oleh  $V'$  (Spanning Subgraph).

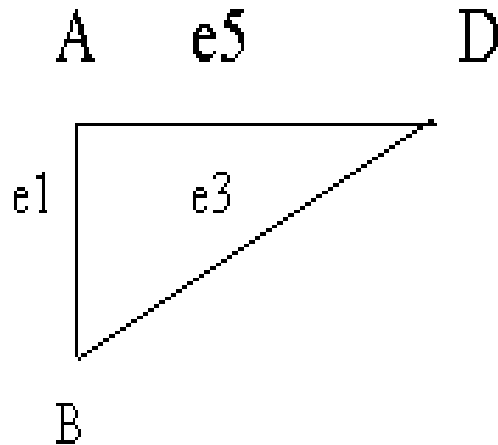
## Contoh Sub Graph:



$G'$  Subgraph dari  $G$  (namun bukan dibentuk oleh  $V' = \{A, B, D\}$ ).

# Contoh Spanning Sub Graph

$G'$  Subgraph yang dibentuk oleh  $V' = \{A, B, D\}$



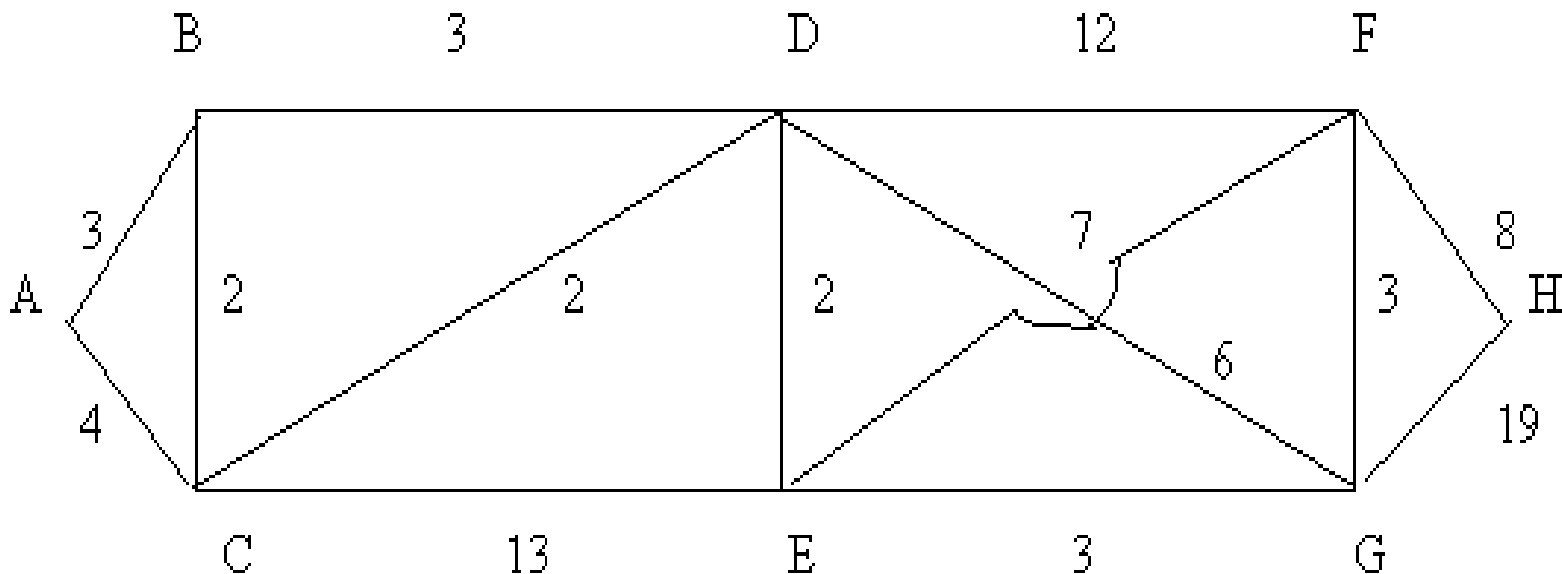
# GRAPH BERLABEL

Graph  $G$  disebut berlabel jika ruas dan atau simpulnya dikaitkan dengan suatu besaran tertentu. Khususnya jika setiap Ruas  $e$  dari  $G$  dikaitkan dengan suatu bilangan non negatif  $d(e)$ , maka  $d(e)$  disebut bobot atau panjang dari ruas  $e$ .



## Contoh :

Gambar berikut ini menyajikan hubungan antar kota.  
 Disini simpul menyatakan kota dan label  $d(e)$  menyatakan jarak antara dua kota.

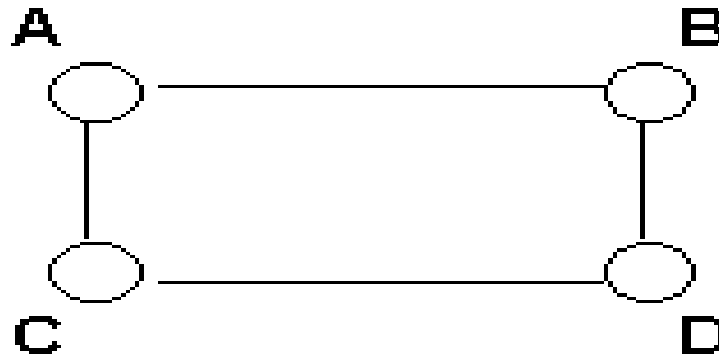


# DERAJAT GRAPH

Derajat simpul  $V$ , ditulis  $d(v)$  adalah banyaknya ruas yang menghubungkan  $v$ . Karena setiap ruas dihitung dua kali ketika menentukan derajat suatu Graph, maka :

***Jumlah derajat semua simpul suatu Graph (derajat) = dua kali banyaknya ruas Graph (Size) Atau dapat dituliskan :***

$$\text{Derajat Graph} = 2 \times \text{Size}$$



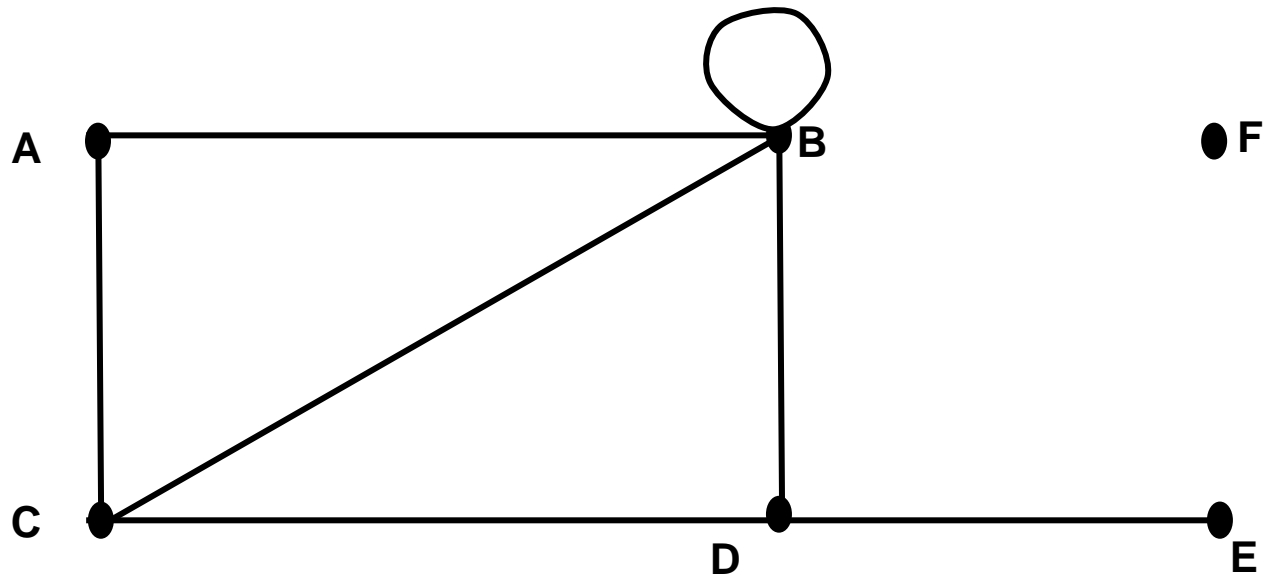
Pada gambar diatas Jumlah Semua Simpul = 4, maka  
Jumlah Derajat Semua Simpul = 8

Jika Derajat masing-masing simpul pada Graph berjumlah  
Genap maka Graph tersebut disebut **EULER Graph**

Suatu simpul disebut genap/ganjil tergantung apakah derajat simpul tersebut genap/ganjil.

Kalau terdapat self-loop, maka self-loop dihitung 2 kali pada derajat simpul.

**Contoh :**



Pada gambar diatas, banyak ruas/size = 7, sedangkan derajat masing-masing simpul adalah :

$$d(A) = 2$$

$$d(B) = 5$$

$$d(C) = 3$$

$$d(D) = 3$$

$$d(E) = 1$$

$$d(F) = 0$$

maka, total jumlah derajat simpul adalah : 14

E disebut simpul bergantung/akhir, yaitu simpul yang berderajat satu. Sedangkan F disebut simpul terpencil, yaitu simpul yang berderajat Nol.

# KETERHUBUNGAN

Walk atau perjalanan dalam Graph  $G$  adalah barisan simpul dan ruas berganti-ganti :  $V_1, e_1, V_2, e_2, \dots, e_{n-1}, V_n$

Disini ruas  $e_i$  menghubungkan simpul  $V_i$  dan  $V_{i+1}$ .

Banyaknya ruas disebut Panjang Walk. Walk dapat ditulis lebih singkat dengan hanya menulis deretan ruas :

$e_1, e_2, \dots, e_{n-1}$  atau deretan simpul :  $V_1, V_2, \dots, V_{n-1}, V_n$

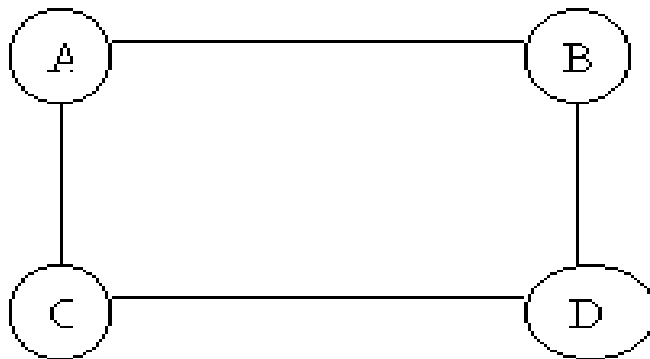
dimana :  $V_1$  = simpul awal

$V_n$  = simpul akhir.

Walk disebut tertutup bila  $V_1 = V_n$

# Istilah Pada Graph

1. Walk disebut **tertutup**, yang menghubungkan  $V_1$  dan  $V_n$ , yaitu Setiap Ruas menghubungkan Simpul Awal dan Akhir



2. Trail adalah Walk dengan semua ruas dalam barisan adalah berbeda.
3. Path atau Jalur adalah Walk yang semua simpul dalam barisan adalah berbeda. Jadi suatu Path pastilah sebuah Trail.



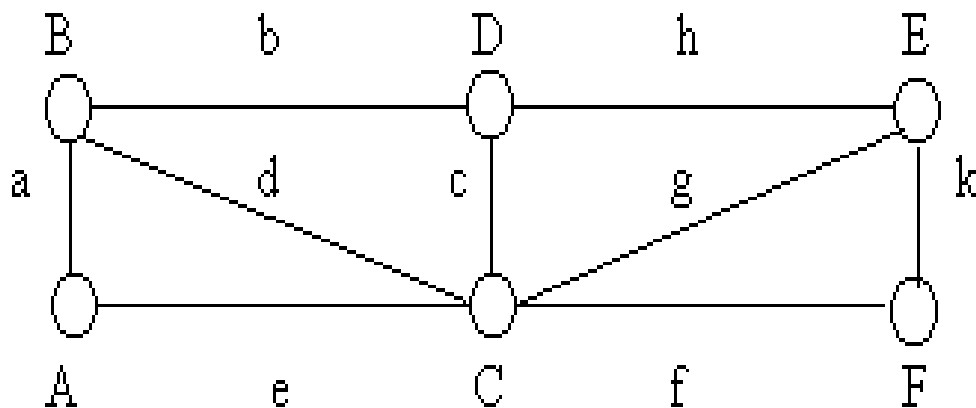
Graph merupakan Walk Terbuka, karena tidak ada ruas yang menghubungkan Simpul U dan T.

Merupakan suatu Path atau Trail terbuka dengan derajat setiap simpulnya = 2, kecuali simpul awal U dan simpul akhir T berderajat = 1.



1. Cycle atau sirkuit adalah suatu Trail tertutup dengan derajat setiap simpul = 2. Cycle dengan panjang  $k$  disebut dengan  $k$ -cycle. Demikian pula Jalur dengan panjang  $k$  disebut  $k$ -jalur.

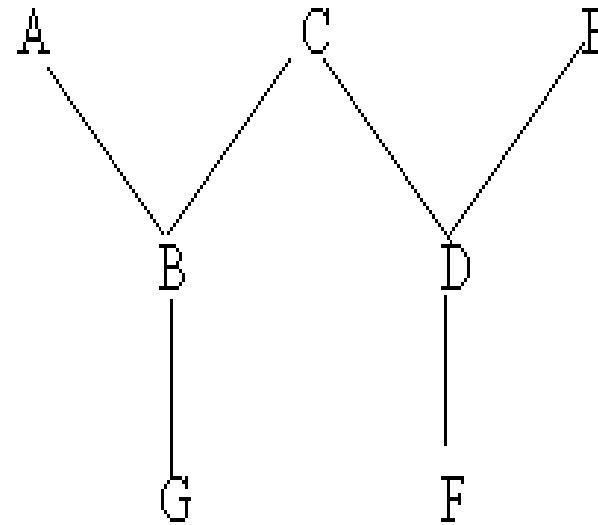
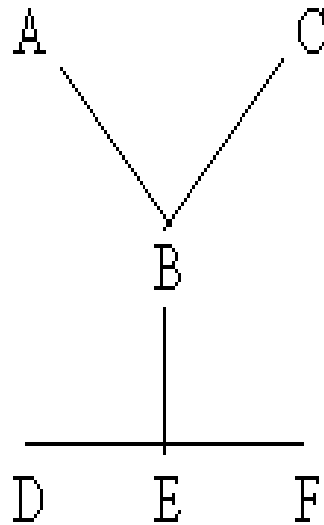
Contoh :



- Barisan ruas a,b,c,d,b,c,g,h adalah Walk bukan Trail (karena ruas b dua kali muncul).
- Barisan simpul A, B, E, F bukan Walk (karena tdk ada ruas yang menghubungkan simpul B ke F).
- Barisan simpul A, B, C, D, E, C, F adalah Trail bukan Jalur/Path (karena c dua kali muncul)
- Barisan ruas a, d, g, k adalah Jalur/Path karena menghubungkan A dengan F
- Ruas a, b, h, g, e, a, adalah Cycle.

Graph yang tidak mengandung Cycle disebut Acyclic.  
Contoh dari Graph Acyclic adalah pohon atau Tree.

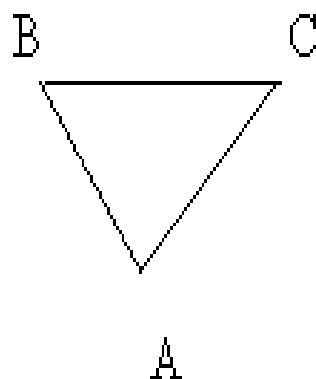
## Contoh dari Acyclic



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

Subgraph yang terhubung pada suatu Graph disebut komponen dari G bila Subgraph tersebut tidak terkandung dalam Subgraph terhubung lain yang lebih besar.

Contoh :



Terlihat misalnya antara D dan A  
Tak ada jalur.

## GRAPH TERARAH (DIRECTED GRAPH / DIGRAPH)

Graph terarah adalah Graph yang dapat menghubungkan  $V_1$  ke  $V_2$  saja (1 arah).

Maksimum jumlah busur dari  $n$  simpul adalah :  $n (n - 1)$

Suatu Graph Berarah (Directed Graph)  $D$  terdiri atas 2 himpunan :

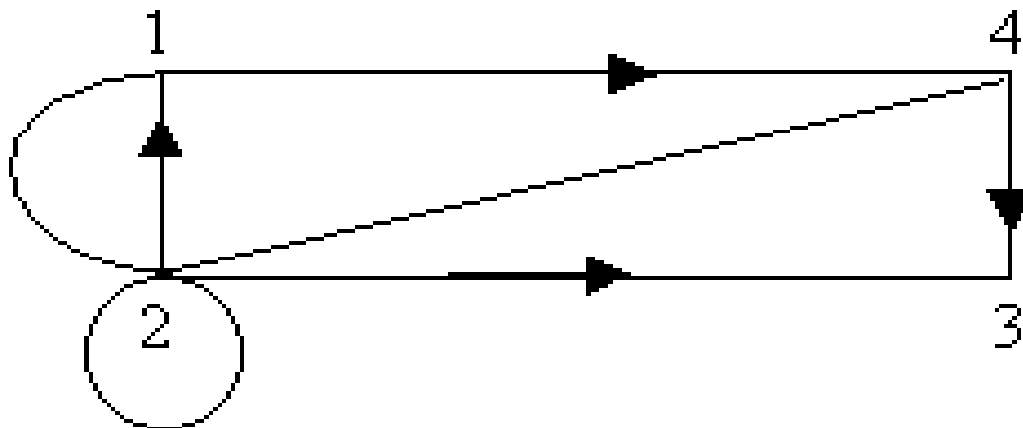
- 1) Himpunan  $V$ , anggotanya disebut simpul.
- 2) Himpunan  $A$ , merupakan himpunan pasangan terurut, yang disebut ruas berarah atau arkus.

## Contoh :

Gambar dibawah ini adalah sebuah Graph Berarah  $D(V,A)$  dengan :

1.  $V$  mengandung 4 simpul, yaitu 1, 2, 3 dan 4
2.  $A$  mengandung 7 arkus, yaitu  $(1,4)$ ,  $(2,1)$ ,  $(2,1)$ ,  $(4,2)$ ,  $(2,3)$ ,  $(4,3)$  dan  $(2,2)$

Arkus  $(2,2)$  disebut gelung (self-loop), sedangkan arkus  $(2,1)$  muncul lebih dari satu kali, disebut arkus sejajar atau arkus ganda.



Bila arkus suatu Graph Berarah menyatakan suatu bobot, maka Graph Berarah tersebut dinamakan jaringan / Network. Biasanya digunakan untuk menggambarkan situasi dinamis.

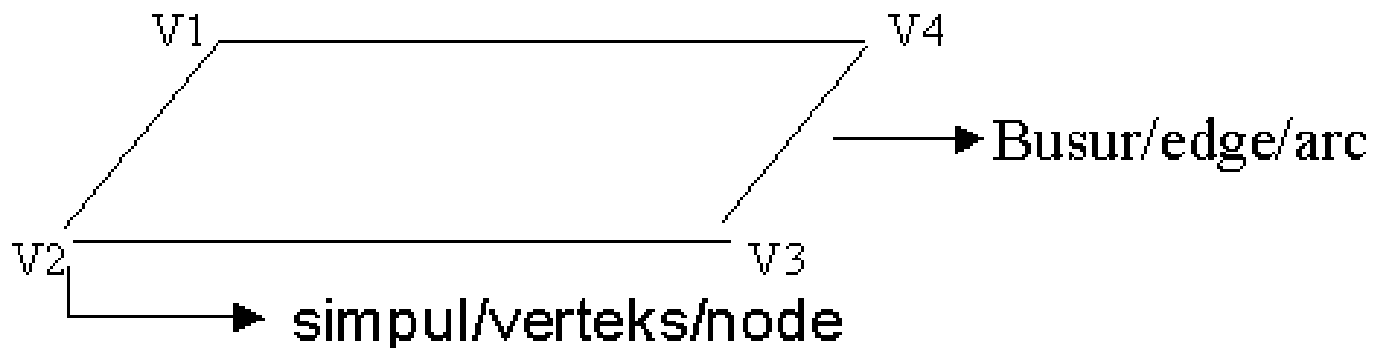
*Bila  $V'$  himpunan bagian dari  $V$  serta  $A'$  himpunan bagian dari  $A$ , dengan titik ujung anggota  $A'$  terletak di dalam  $V'$ , maka dikatakan bahwa  $D'(V', A')$  adalah Graph bagian (Subgraph) dari  $D(V, A)$ .*

*Bila  $A'$  mengandung semua arkus anggota  $A$  yang titik ujungnya anggota  $V'$ , maka dikatakan bahwa  $D'(V', A')$  adalah Graph Bagian yang dibentuk atau direntang oleh  $V'$ .*

# Undirected Graph

## GRAPH TAK TERARAH (UNDIRECTED GRAPH)

Graph Tak Terarah adalah Graph yang menghubungkan 2 verteks  $V1$  ke  $V2$  dan  $V2$  ke  $V1$  (2 arah). Bila Verteks =  $n$ , maka Graph tak terarah komplit akan mempunyai busur edge sama dengan :  $\boxed{n(n-1)/2}$



Yang dapat dilakukan adalah :

$$V1 - V2 = V2 - V1$$

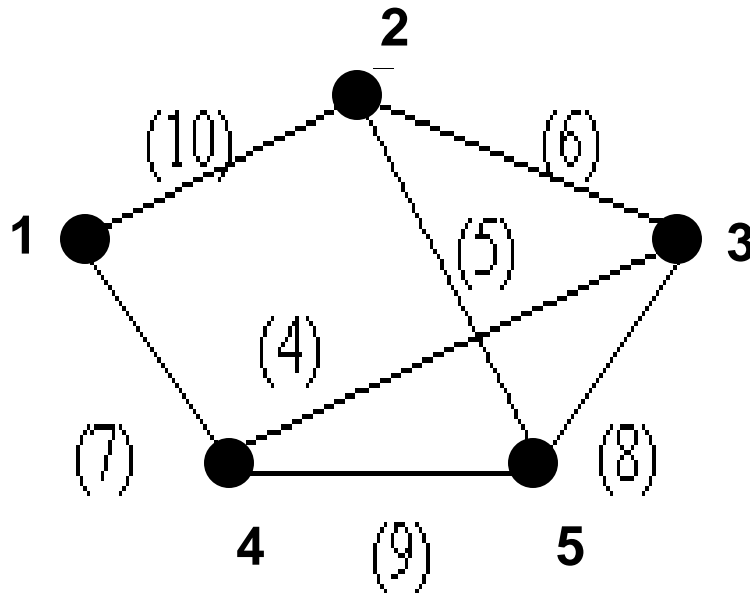
$$V1 - V4 = V4 - V1$$

dan seterusnya.



# Critical Path

Menggunakan Graph berbobot dan Mempunyai Arah



Simpul asal : 1

Simpul Tujuan : 5

# Critical Path

	PATH	BOBOT
Alternatif :	1-->4 -->5	16
	1-->2-->5	15
	1-->2-->3-->5	24
	1-->4-->3-->5	19
	1-->2-->3-->4-->5	29
	1-->4-->3-->2-->5	22

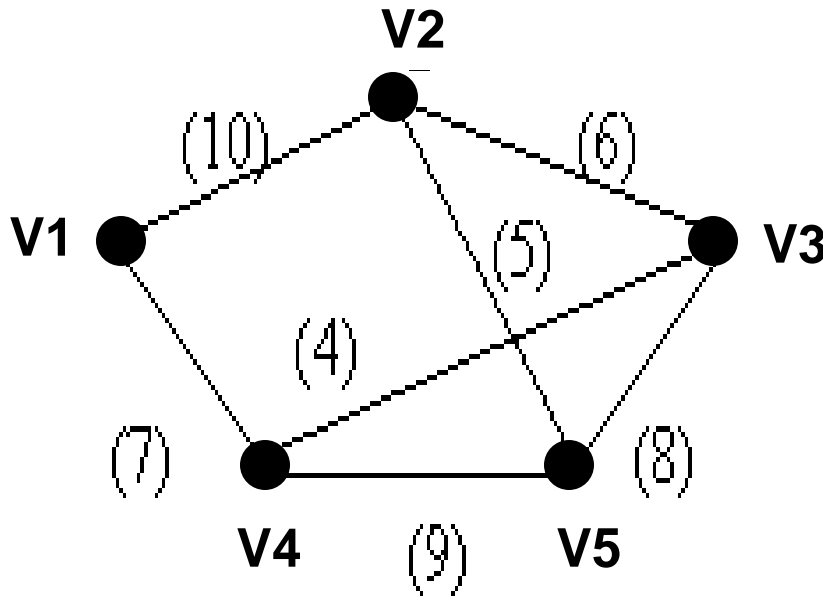
Diperoleh : Critical Path ( Lintasan Kritis ) = 29

Shortest Path ( Lintasan Terpendek ) = 15

# Minimum Spanning Tree

Merupakan Spanning Tree yang mempunyai Bobot dan tidak mempunyai arah dengan hasil penjumlahan bobotnya adalah minimum.

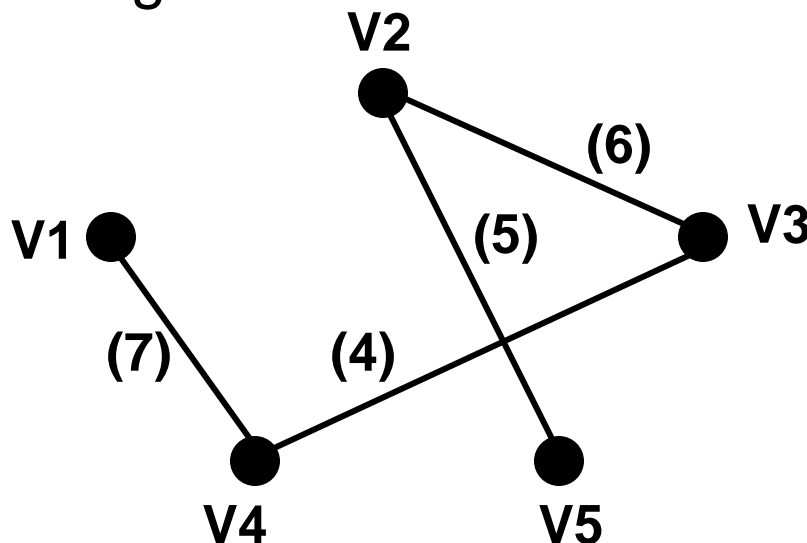
Lihat gambar Graph G berikut :



# Minimum Spanning Tree

Langkah yang dilakukan untuk membentuk minimum spanning tree adalah :

Bentuk kembali semua simpul tetapi tanpa ruas.  
Gambar dan telusuri ruas dengan bobot paling kecil, seterusnya (secara ascending) hingga semua simpul terhubung



Total Minimum  
Spanning Tree = 22

# MATRIKS PENYAJIAN GRAPH

Misalnya disajikan Graph G dalam Matriks ruas B ukuran  $(M \times 2)$ , maka setiap baris Matriks menyatakan ruas, misalnya baris (4 7) menyatakan ada ruas menghubungkan simpul 4 dan 7.

***Matriks Adjacency*** dari Graph G, yaitu Matriks yang menghubungkan Vertex dengan Vertex, tanpa ruas sejajar adalah Matriks A berukuran  $(N \times N)$  yang bersifat :

$$a_{ij} = \begin{cases} 1, & \text{bila ada ruas } (V_i, V_j) \\ 0, & \text{bila dalam hal lain.} \end{cases}$$

**Matriks Adjacency** merupakan matriks simetri.

Untuk Graph dengan ruas sejajar, Matriks Adjacency didefinisikan sebagai berikut :

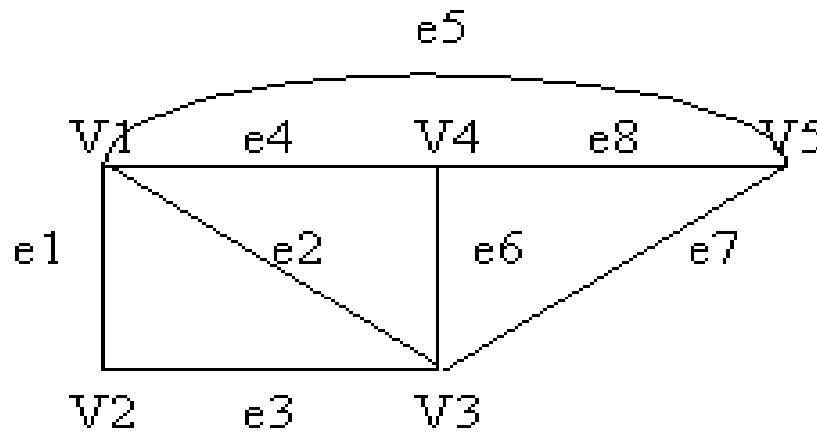
$$a_{ij} = \begin{cases} P, & \text{bila ada } p \text{ buah ruas menghubungkan} \\ & (V_i, V_j) (p > 0) \\ 0, & \text{bila dalam hal lain.} \end{cases}$$

**Matriks Incidence** dari Graph G, yaitu Matriks yang menghubungkan Vertex dengan Edge, tanpa self-loop didefinisikan sebagai Matriks M berukuran (NXM) sebagai berikut :

$$m_{ij} = \begin{cases} 1, & \text{bila ada ruas } e_j \text{ berujung di simpul } V_i \\ 0, & \text{dalam hal lain.} \end{cases}$$

# Matriks Adjacency

Contoh :



Matriks Ruas

1	2
1	3
1	4
1	5
2	3
3	4
3	5
4	5

atau :

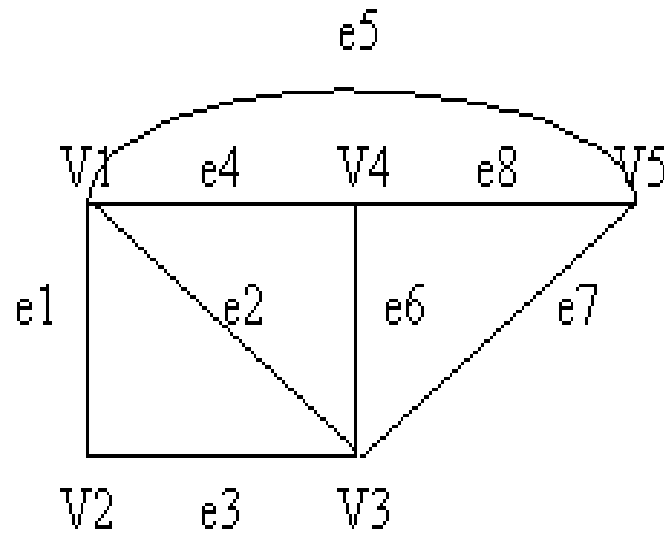
1	1	1	1	2	3	3	4
2	3	4	5	3	4	5	5

atau secara pasangan

$\{(1,2),(1,3),(1,4),(1,5),(2,3),(3,4),(3,5),(4,5)\}$

# Matriks Adjacency

Contoh :

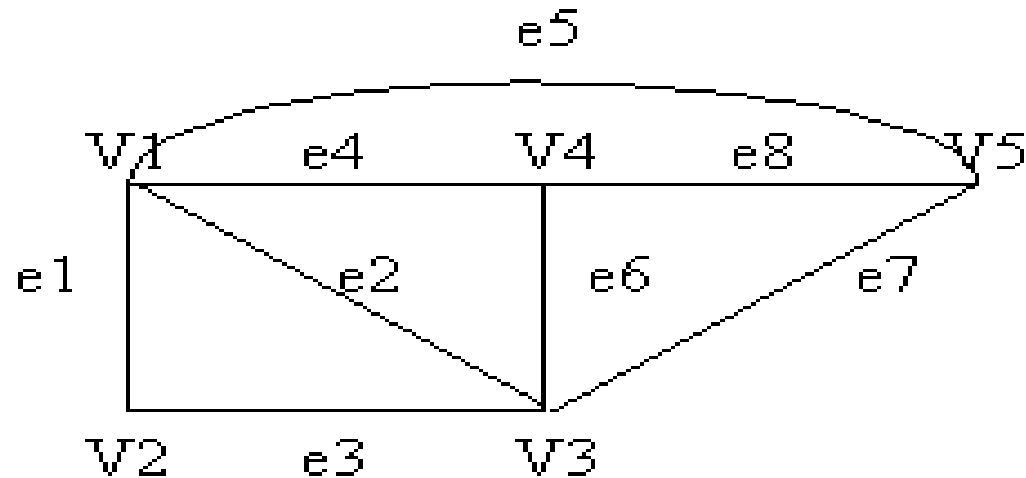


Matriks Adjacency

	V1	V2	V3	V4	V5
V1	0	1	1	1	1
V2	1	0	1	0	0
V3	1	1	0	1	1
V4	1	0	1	0	1
V5	1	0	1	1	0



Contoh :



Matriks Incidence

	V2	e3	V3					
	e1	e2	e3	e4	e5	e6	e7	e8
V1	1	1	0	1	1	0	0	0
V2	1	0	1	0	0	0	0	0
V3	0	1	1	0	0	1	1	0
V4	0	0	0	1	0	1	0	1
V5	0	0	0	0	0	1	0	1

## GRAPH TERARAH (DIRECTED GRAPH / DIGRAPH)

Graph terarah adalah Graph yang dapat menghubungkan  $V_1$  ke  $V_2$  saja (1 arah).

Maksimum jumlah busur dari  $n$  simpul adalah :  $n(n - 1)$

Suatu Graph Berarah (Directed Graph)  $D$  terdiri atas 2 himpunan :

- 1) Himpunan  $V$ , anggotanya disebut simpul.
- 2) Himpunan  $A$ , merupakan himpunan pasangan terurut, yang disebut ruas berarah atau arkus.

# Graph

## PENELUSURAN GRAPH

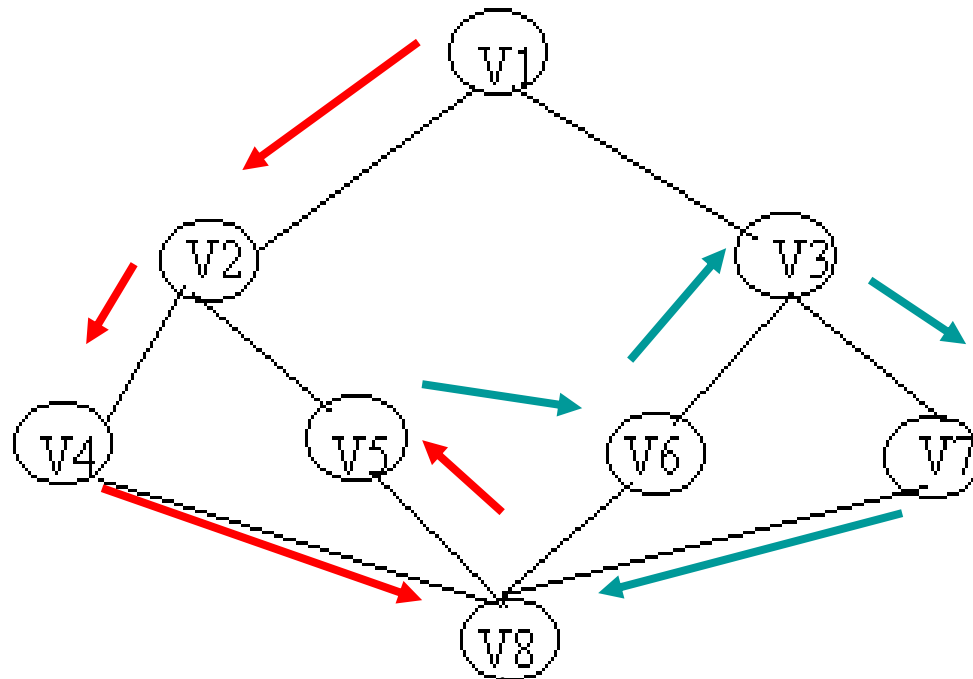
Dapat dilakukan dengan 2 cara , yaitu :

1. Depth First Search (DFS)
2. Breadth First Search (BFS)

### 1. Depth First Search (DFS)

Penelusuran dengan DFS pada Graph Tak Berarah dengan melakukan pengecekan pada Node dengan kedalaman pertama dari Node yang ditinjau.

# Depth First Search



Karena V8 sudah dilewati setelah penelusuran ke V4, maka penelusuran yang berikutnya dianggap tidak dilewati lagi

Klik Animasi

**Dari gambar diatas dapat terbentuk matriks sbb :**

	V1	V2	V3	V4	V5	V6	V7	V8
V1	0	1	0	0	0	0	0	0
V2	0	0	0	1	0	0	0	0
V3	0	0	0	0	0	0	1	0
V4	0	0	0	0	0	0	0	1
V5	0	0	0	0	0	1	0	0
V6	0	0	1	0	0	0	0	0
V7	0	0	0	0	0	0	0	1
V8	0	0	0	0	1	0	0	0

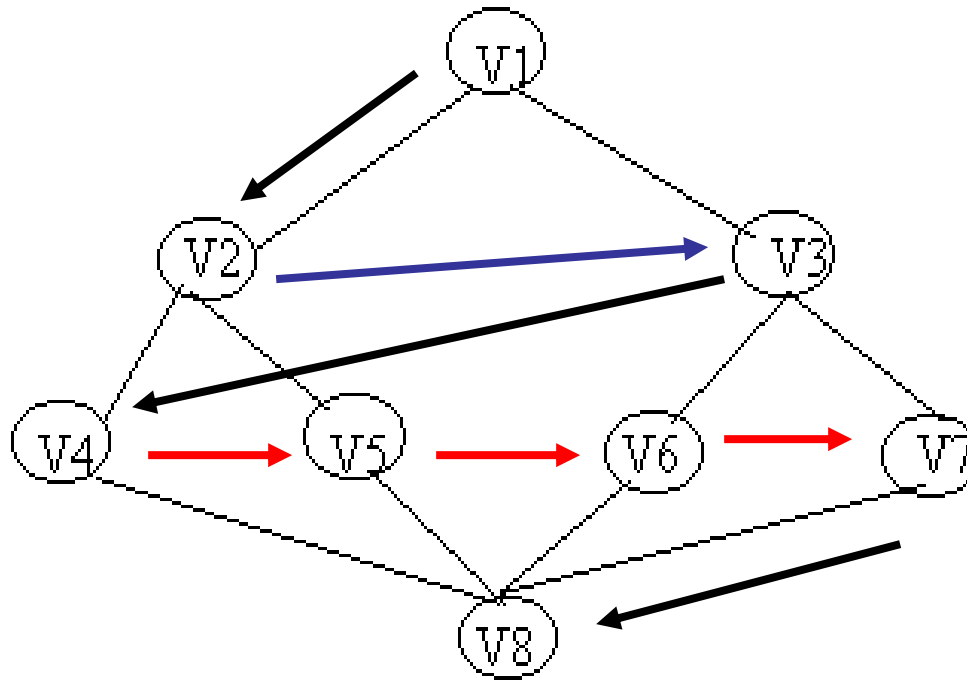
**Dari Matriks diatas, akan diperoleh urutan sbb:**

**V1 -> V2 -> V4 -> V8 -> V5 -> V6 -> V3 -> V7**

## 2. Breadth First Search (BFS).

Berbeda dengan cara DFS, dengan BFS penelusuran akan diawasi dari Node-1, kemudian melebar pada Adjacent Node dari Node-1 dan diteruskan pada Node-2, Node- 3 dan seterusnya.

# Breadth First Search



Dari gambar di atas akan diperoleh urutan :  
**V1** , **V2** ---> **V3** , **V4** ---> **V5** ---> **V6** ---> **V7** , **V8**

Klik Animasi