BAB 2
PENGERTIAN DASAR TENTANG GRAPH

2.1. Beberapa Definisi Dasar Dalam Graph

Definisi 2.1.1. :

Suatu himpunan tidak kosong berhingga dan terdiri dari vertex-vertex $X = X(G)$ dan edge-edge $E = E(G)$ disebut *Graph*. Dan ditulis $G = (X,E)$.

Untuk menyatakan graph secara geometris maka vertex dinyatakan dengan sebuah titik, sedangkan edge dinyatakan dengan sebuah garis yang menghubungkan dua titik.

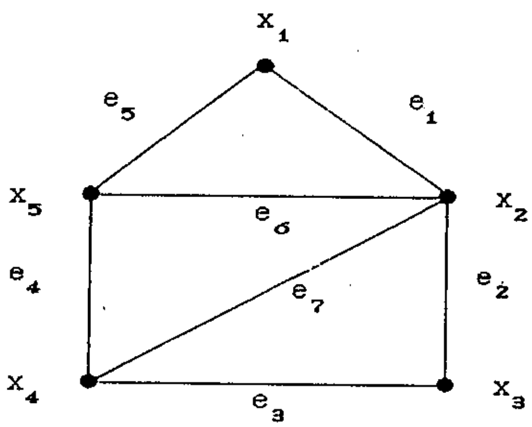
Contoh :

1. X mengandung 5 vertex yaitu : x_1, x_2, x_3, x_4, x_5
2. E mengandung 7 edge yaitu :

$e_1 = \{x_1, x_2\}$; $e_5 = \{x_1, x_5\}$; $e_2 = \{x_2, x_3\}$

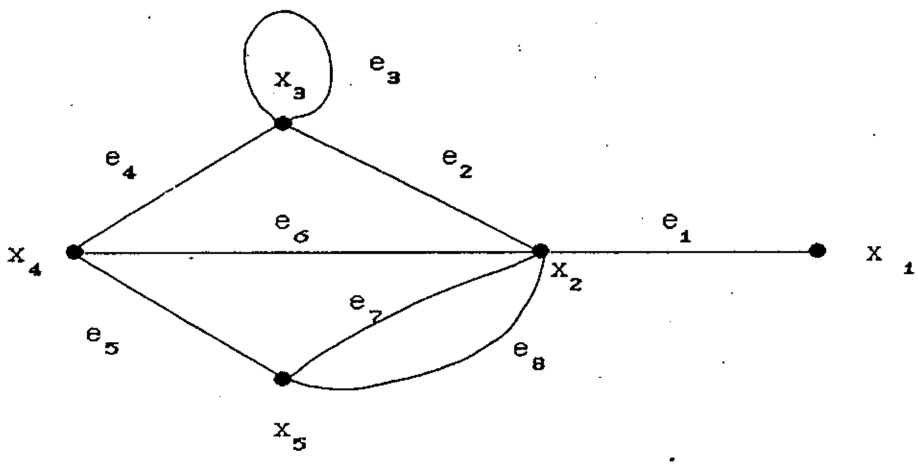
$e_6 = \{x_2, x_5\}$; $e_3 = \{x_3, x_4\}$; $e_7 = \{x_2, x_4\}$

$e_4 = \{x_4, x_5\}$



Gambar 2.1.1

Gambar dibawah ini merupakan suatu graph yang lebih umum, disebut *Multigraph.* Karena mengandung edge berganda (paralel) yaitu 2 buah edge yang mempunyai 2 titik ujung yang sama (e_7 dan e_8) dan edge self loop yaitu kedua titik ujungnya berimpit (e_3).



Gambar 2.1.2

Jadi suatu graph dikatakan sederhana jika graph tersebut tidak mengandung edge sejajar ataupun self loop (lihat gambar 2.1.1.)

Bila sebuah vertex X_j adalah vertex ujung dari beberapa edge e_j maka dikatakan X_j insiden dengan e_j atau e_j insiden dengan X_j. Misalnya dalam gambar 2.1.2 edge e_1, e_2, e_6, e_7, e_8 insiden dengan vertex X_2.

Dua buah edge yang tidak paralel dikatakan bersisian (adjacent) jika kedua edge tersebut insiden pada sebuah vertex yang sama, misalkan pada gambar 2.1.2 yaitu e_4 dan e_5 adalah adjacent

Dengan cara yang sama , dua vertex dikatakan adjacent jika kedua vertex tersebut merupakan vertex ujung dari edge yang sama, misalkan pada gambar 2.1.2 yaitu X_1 dan X_2 adalah adjacent. Tetapi vertex X_1 dan X_3 tidak adjacent.

Definisi 2.1.2. :

Dalam suatu graph terdapat dua hal :

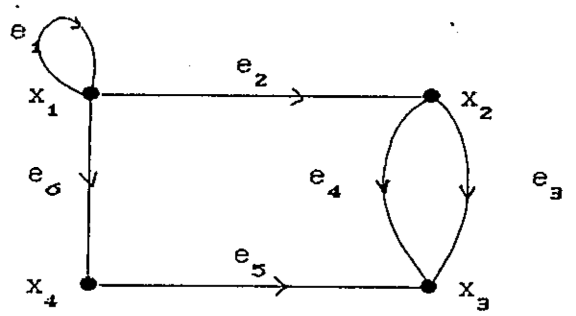
(1). Bila (x_i, x_k) tidak dibedakan dengan (x_k, x_i), G disebut *Graph tak berarah.*

(2). Bila (x_i, x_k) dibedakan dengan (x_k, x_i), G disebut *Graph berarah* atau *Digraph.*

Untuk graph tak berarah, x_i dan x_k disebut vertex ujung dari e_j. Untuk graph berarah $e_j = \{x_i, x_k\}$, x_i disebut vertex awal dan x_k disebut vertex akhir dari e_j.

Contoh :

Suatu graph G = (X,E) terdiri dari 4 vertex dan 6 edge.



Gambar 2.1.3 : Digraph atau Graph Berarah

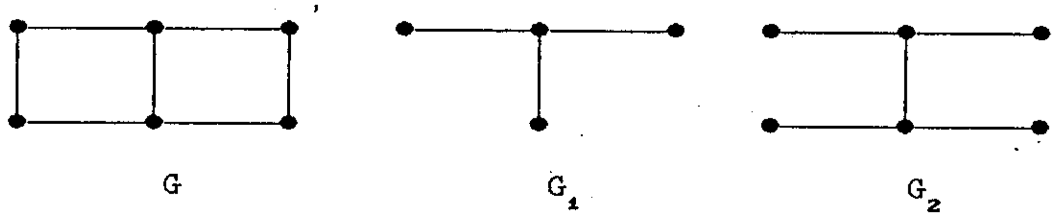
Definisi 2.1.3. :

Diberikan $A \subset X$, maka subgraph yang dihasilkan oleh A adalah graph $G_A = (A, E_A)$ yang mana vertex-vertex-nya adalah elemen-elemen A dan edge-edge pada E_A

adalah edge dari G yang memiliki dua titik akhir di A.

$G'(X',E')$ dikatakan *Spaning Subgraph* dari G apabila G' mengandung semua vertex-vertex G ($X' = X$).

Contoh :



Gambar 2.1.4 : Sebuah graph dan dua buah subgraph

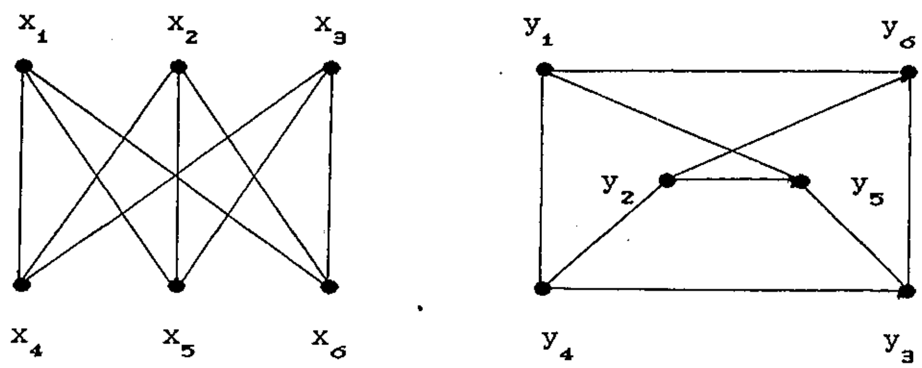
G_1 merupakan subgraph dan G_2 merupakan spaning subgraph.

<u>Definisi 2.1.4.</u> :

Dua buah graph G_1 dan G_2 disebut *Isomorfis* jika ada fungsi one-one onto $f : X(G_1) \longrightarrow X(G_2)$ sedemikian sehingga untuk setiap edge (Y,X) di G_1, berlaku jika dan hanya jika (f(Y),f(X)) adalah edge di G_2. Isomorfis dari graph G_1 dan G_2 ditulis :

$$G_1 \cong G_2$$

Contoh :



Gambar 2.1.5 : Dua graph Isomorfis

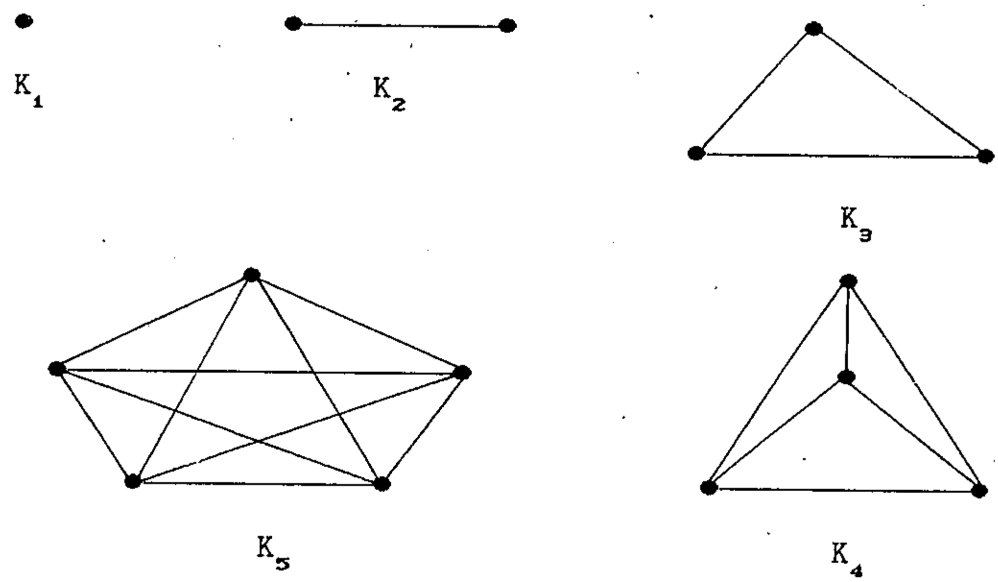
2.2. Graph-graph Yang Khusus

<u>Definisi 2.2.1</u> :

Suatu graph dimana setiap dua vertex selalu dihubungkan oleh satu dan hanya satu edge disebut *Graph Complete*. Biasa ditulis K_n, n menunjukkan banyaknya vertex. Pada K_n terdapat :

$$\begin{bmatrix} n \\ 2 \end{bmatrix} = \frac{1}{2} \; n (n - 1) \text{ edge}$$

Contoh :



Gambar 2.2.1

<u>Definisi 2.2.2</u> :

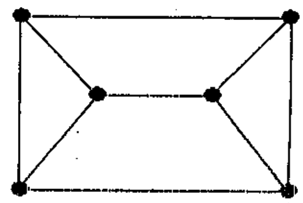
Graph disebut *Graph Regular* berderajat k bila setiap vertex mempunyai derajat yang sama yaitu k.

Dapat dicatat bahwa graph lengkap K_n adalah Graph Regular dengan derajat :

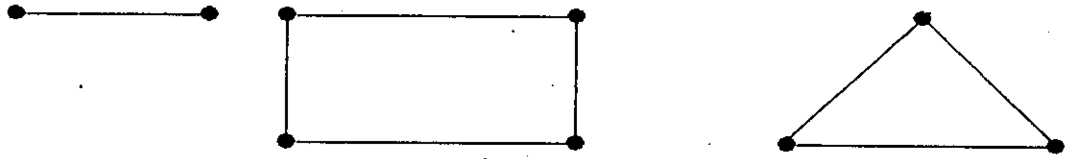
$$k = n - 1$$

Contoh :

derajat 1



derajat 3



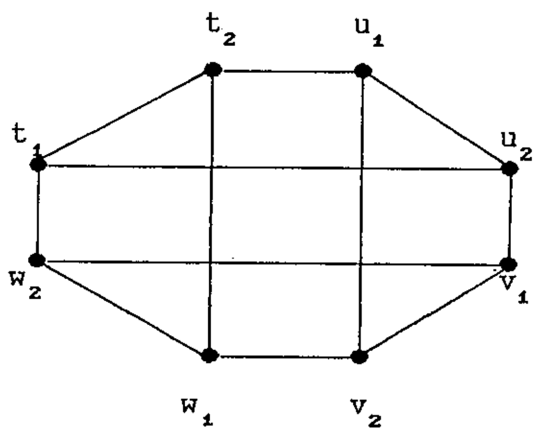
derajat 2

Gambar 2.2.2.

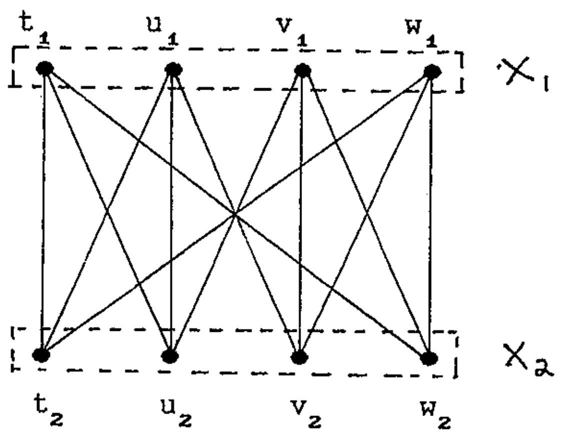
<u>Definisi 2.2.3.</u> :

Bipartisi suatu Graph K adalah suatu graph dengan himpunan vertex X(G) dipartisi dalam dua himpunan bagian X_1 dan X_2 sedemikian sehingga setiap edge menghubungkan X_1 dengan X_2.

Contoh :



G



K

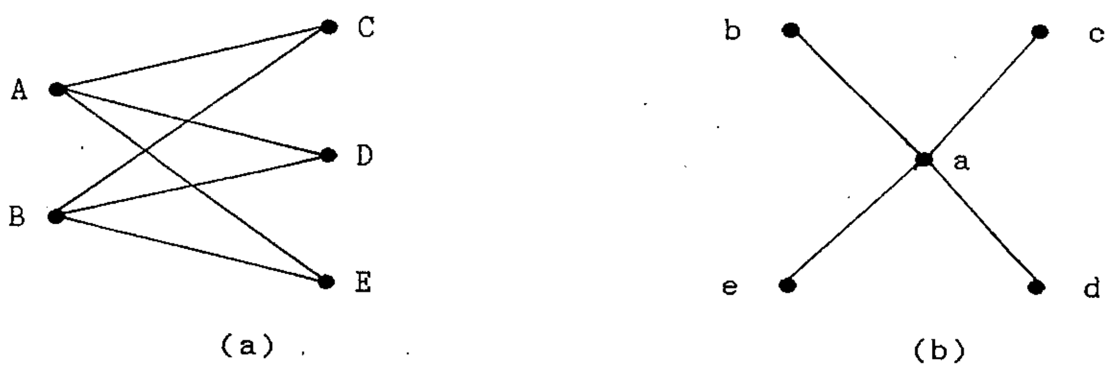
Gambar 2.2.3

Suatu graph G disebut *Graph Bipartisi* jika dan hanya jika panjang setiap sirkuit dalam graph tersebut adalah genap (lihat gambar 2.8).

<u>Definisi 2.2.4.</u> :

Suatu graph Bipartisi disebut *Bipartisi lengkap* jika ada edge antara setiap vertex $u \in X_1$ ke setiap vertex $v \in X_2$. Jadi jika X_1 dan X_2 mempunyai m dan n vertex maka graph Bipartisi lengkap $K_{m,n}$ mempunyai (m x n) edge.

Contoh :



Gambar 2.2.4.

Gambar 2.9 (a) adalah $K_{2,3}$, X_1 = (A,B) , X_2 = (C,D,E)

Gambar 2.9 (b) adalah $K_{1,4}$, X_1 = (a) , X_2 = (b,c,d,e)

$K_{1,4}$ disebut juga *Star* (bintang).

<u>Definisi 2.2.5.</u> :

Suatu graph terhubung yang tidak mempunyai cycle disebut *Tree*.

Beberapa sifat dari Tree :

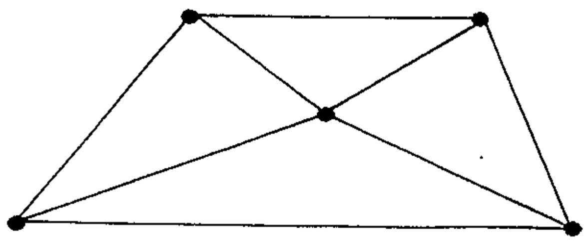
1. Hanya ada satu path (yaitu suatu jalur dimana garis tidak boleh diulang, titik boleh) untuk setiap dua

vertex.

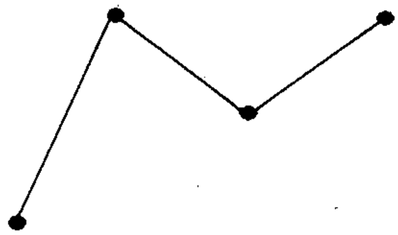
2. Jika banyaknya vertex sama dengan n, maka banyaknya edge sama dengan (n - 1).

Dan graph yang lain tak mempunyai cycle dan tidak connected disebut *Forest.* Jadi componen-componen dari suatu Forest adalah Tree.

Contoh :



G



K

Gambar 2.2.5.

2.3. Pengertian Istilah-Istilah

2.3.1. Componen Connected

Definisi 2.3.1.1. :

Graph disebut *Connected* jika dan hanya jika setiap dua vertex dihubungkan dengan sekurang-kurangnya satu edge, apabila tidak demikian maka G disebut *Disconnected.*

Definisi 2.3.1.2. :

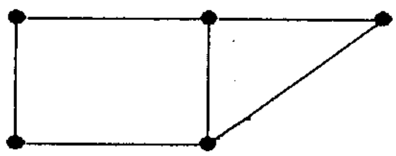
Suatu maximal connected Subgrap G disebut *Componen Connected* atau suatu *Componen G.*

Jadi suatu disconected graph sedikitnya mempunyai dua componen.

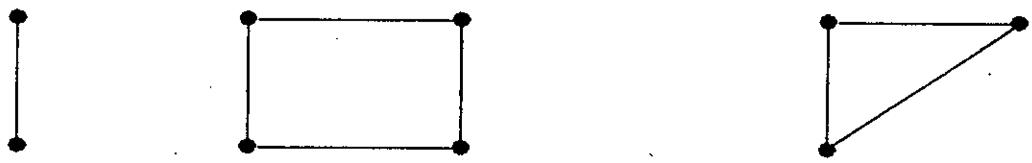
Untuk lebih jelasnya dapat dilihat pada gambar dibawah ini.

Contoh :

Bila Graph G :



Maka subgraph G adalah :



(a) (b) (c)

Jadi graph G diatas mempunyai 3 componen connected.

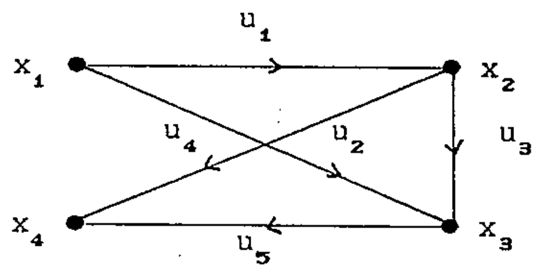
2.3.2. Matrix Dalam Graph

Definisi 2.3.2.1 :

Matrix Incidence vertex - arc suatu graph G = (X,U) adalah suatu matrix A = a_{ij} , dengan n vertex yaitu $x_1, x_2, \ldots\ldots x_n$ dan m arc $u_1, u_2, \ldots\ldots u_m$ yang tidak memiliki loop dan berukuran n x m dengan elemen-elemen :

$$a_{ij} = \begin{cases} 1, & \text{jika vertex } x_i \text{ adalah vertex awal dari arc } u_i. \\ -1, & \text{Jika vertex } x_i \text{ adalah vertex akhir dari akhir dari arc } u_i. \\ 0, & \text{jika arc } u_i \text{ tidak berpangkal pada vertex } x_i. \end{cases}$$

Contoh :



Gambar 2.3.2.1.

Matrix Incidence vertex - arc :

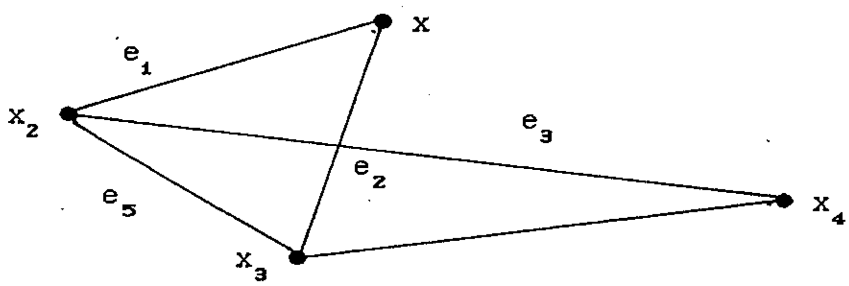
$$A = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \begin{array}{c} \begin{array}{ccccc} u_1 & u_2 & u_3 & u_4 & u_5 \end{array} \\ \begin{bmatrix} +1 & +1 & 0 & 0 & 0 \\ -1 & 0 & +1 & +1 & 0 \\ 0 & -1 & -1 & 0 & +1 \\ 0 & 0 & 0 & -1 & -1 \end{bmatrix} \end{array}$$

<u>Definisi 2.3.2.2.</u> :

Matrix Incidence Vertex - Edge pada suatu graph $G = (X,E)$ adalah suatu matrix $B = b_{ij}$ dengan n vertex yaitu $x_1, x_2, \ldots. x_n$ dan m edge $e_1, e_2, \ldots. e_m$ dan berukuran n x m dengan elemen-elemen :

$$b_{ij} = \begin{cases} 1, \text{ jika } x_i \text{ dan } e_j \text{ adalah inciden.} \\ 0, \text{ jika } x_i \text{ dan } e_j \text{ tidak inciden} \end{cases}$$

Contoh :



Gambar 2.3.2.2.

Maka matrix incidence vertex - arc adalah :

$$B = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{array}\begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

2.3.3. Bilangan Cyclomatik

Definisi 2.3.3.1 :

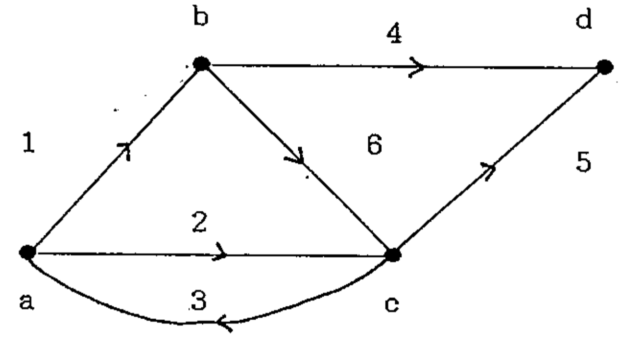
Suatu *Basis Cycle* (cycle dasar) didefinisikan sebagai himpunan $(\mu^1, \mu^2, \ldots\ldots\ldots \mu^k)$ dari cycle-cycle yang bebas sedemikian sehingga terdapat sembarang cycle μ ,dapat ditulis sebagai berikut :

$$\mu = r_1.\mu^1 + r_2.\mu^2 + \ldots\ldots\ldots + r_k.\mu^k$$

Dimana $r_1, r_2 \ldots\ldots r_k$ bilangan real.

Sehingga himpunan tersebut terdiri dari k vektor. Sehingga k = dimensi ruang bagian R^m diperoleh dengan cycle dan tidak bergantung pada pemilihan basis. Konstanta K disebut *Bilangan Cyclomatik G,* dan dinotasikan dengan ν (G).

Contoh :



Gambar 2.3.3.1.

Dari gambar diatas dapat dicari :

$$\mu^1 = (1,6,2) = (a\ b\ c\ a)$$

$$\mu^2 = (1,6,3) = (a\ b\ c\ a)$$

$$\mu^3 = (2,3) = (a\ c\ a)$$

$$\mu^4 = (1,4,5,2) = (a\ b\ d\ c\ a)$$

$$\mu^5 = (6,5,4) = (b\ c\ d\ b)$$

$$\mu^6 = (1,4,5,3) = (a\ b\ d\ c\ a)$$

Cycle-cycle tersebut bergantung karena bila kita ambil :

$$\mu^1 - \mu^2 + \mu^3 = 0$$

Sehingga cycle μ^2,μ^3,μ^6 adalah bentuk suatu basis cycle Oleh karenanya k = 3 atau ν (G) = 3.

Bilangan cyclomatik dari suatu graph G = (X,U) dengan N vertex , M arc dan p componen connected adalah :

$$\nu (G) = M - N + p$$

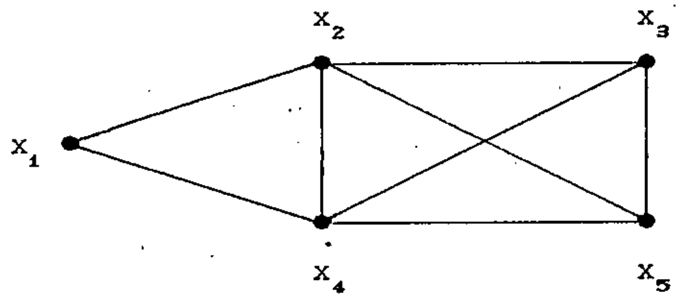
2.3.4. Kelompok Maksimal dari Graph

Definisi 2.3.4.1. :

Diberikan suatu graph G = (X,E). Misalkan $\mathscr{E}$ = (E_j) merupakan himpunan kelompok maksimal graph $(H)_z$. Maka semua himpunan stabil graph G sesuai dengan mathcing hypergraph H^* ,yaitu dual hypergraph H = $(X,\mathscr{E})$

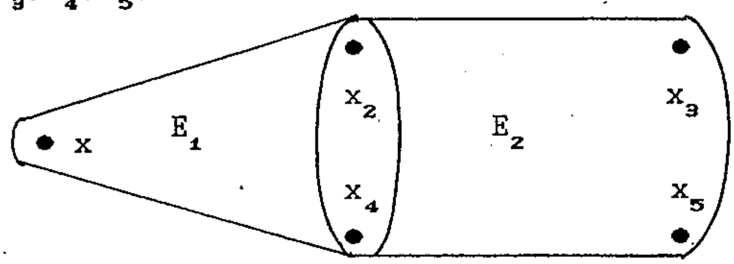
Contoh :

Ikuti graph dibawah ini, kemudian mencari cligue maximal graph dibawah ini .



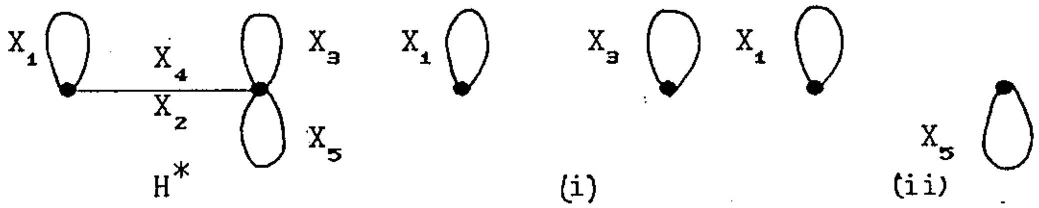
Gambar 2.3.4.1.

1. Stabel Set graph G adalah suatu subset I dari X(G) jika dan hanya jika I adalah clique $\overline{G}$, sehingga stabel Set graph G adalah { (x_1,x_3) , (x_1,x_5) }.
2. Graph G = (X,E) dapat diasosiasik sebagai sebuah hypergraph dari clique maksimal yaitu $E_1 = (x_1,x_2,x_4)$, $E_2 = (x_2,x_3,x_4,x_5)$



Gambar 2.15

3. Cari mathcing dari hypergraph H^*.



4. Terlihat bahwa stabel set graph G sesuai dengan mathcing dari hypergraph H^*.

   Dari definisi tersebut diambil kesimpulan bahwa apabila stabel set G tidak sesuai dengan mathcing hypergraph H^* maka $\mathcal{E} = (E_j)$ bukan himpunan clique maksimal graph $(H)_2$

2.3.5. Partisi Suatu Graph

Degree $d_1, d_2, \ldots . d_p$ dari vertex (titik) untuk suatu graph dalam bentuk sebuah persamaan bilangan bulat non-negativ (positif), yang mana jumlahnya tentu saja 2q. Degree dari sebuah graph tanpa vertex terasing yang demikian menentukan partisi dari 2q.

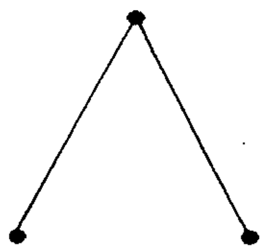
Definisi 2.3.5.1 :

Partisi dari sebuah graph adalah partisi dari 2q sebagai penjumlahan degree dari vertex , ditulis ;

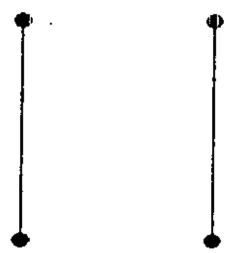
$$2q = \Sigma\, d_i$$

Hanya dua dari 5 partisi 4 dalam penjumlahan nonnegativ kepunyaan suatu graph.

Contoh :



2+1+1



1+1+1+1

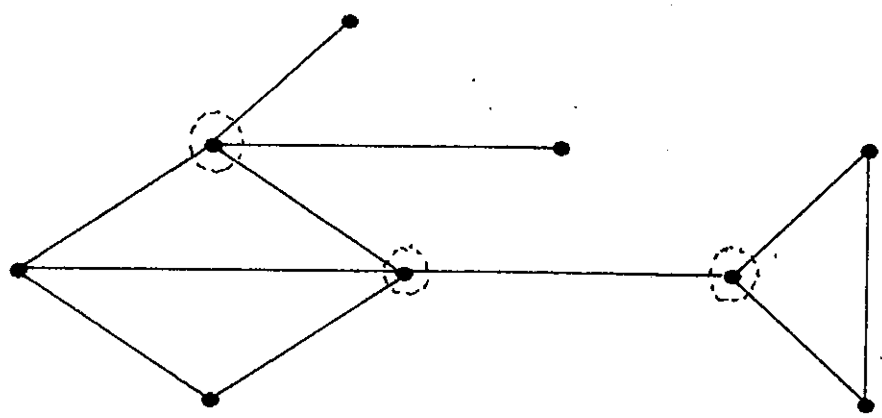
Gambar 2.3.5.1 : Graphical partisi dari 4

2.3.6. Vertex Artikulasi

Definisi 2.3.6.1. :

Suatu vertex yang penghapusannya terhadap pertambahan bilangan (nomor) componen connected sebuah graph atau menyebabkan terjadinya graph yang connected (menghapuskan adanya disconnected graph).

Contoh :



Gambar 2.3.6.1. : sebuah graph dengan 3 vertex artikulasi

2.4. Operasi-operasi pada Graph

Pandang dua graph $G_1 = (X_1, E_1)$ dan $G_2 = (X_2, E_2)$. Operasi-operasi terhadap G_1 dan G_2 diberikan dibawah ini

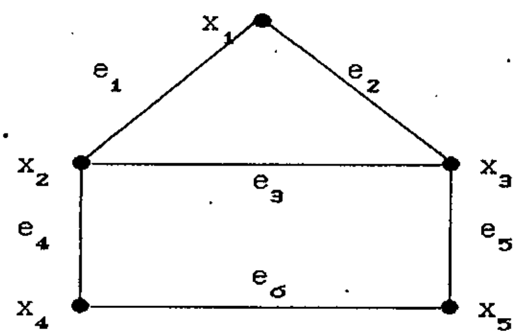
1. Irisan G_1 dan G_2 , $G = G_1 \cap G_2$ didefinisikan segai $G = (X,E)$ dimana $X = X_1 \cap X_2$ dan $E = E_1 \cap E_2$
2. Gabungan G_1 dan G_2 , $G = G_1 \cup G_2$ didefinisikan sebagai $G = (X,E)$ dimana $X = X_1 \cup X_2$ dan $E = E_1 \cup E_2$.
3. Ring Sum G_1 dan G_2, $G = G_1 + G_2$ didefinisikan sebagai $G = (G_1 \cup G_2) - (G_1 \cap G_2)$.
4. Produk G_1 dan G_2, $G = G_1 \times G_2$ didefinisikan sebagai $G = (X,E)$ dan dimana $X = X_1 + X_2$, dua buah titik $u = (u_1, u_2)$ dan $v = (v_1, v_2)$ adjacent dalam G jika ($u_1 = v_1$ dan u_2 adjacent v_2) atau

($u_2 = v_2$ dan u_1 adjacent v_1)

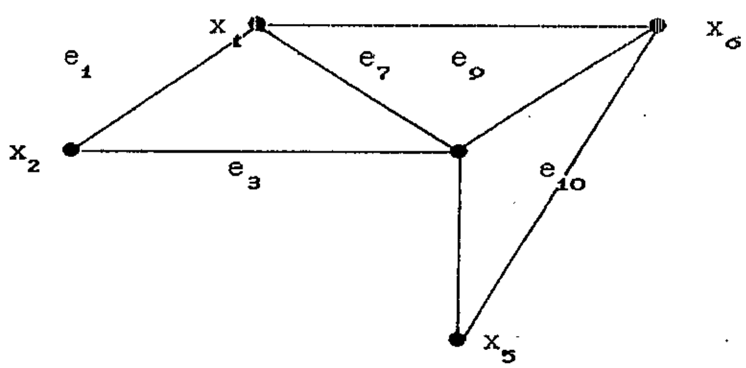
5. Komposisi G_1 dan G_2, $G = G_1(G_2)$ didefinisikan sebagai $G = (X,E)$ dimana $X = X_1 \times X_2$, dua buah titik yaitu $u = (u_1,u_2)$ dan $v = (v_1,v_2)$ adjacent jika (u_1 adjacent v_1) atau ($u_1 = v_1$ dan u_2 adjacent v_2).

$$G_1\ (G_2) \neq G_2\ (G_1)$$

Contoh :

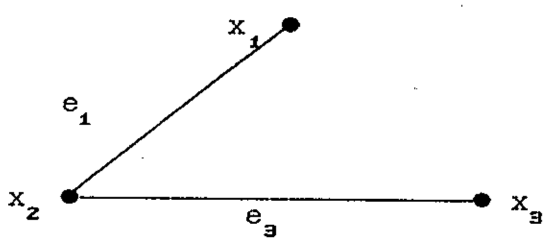


G_1

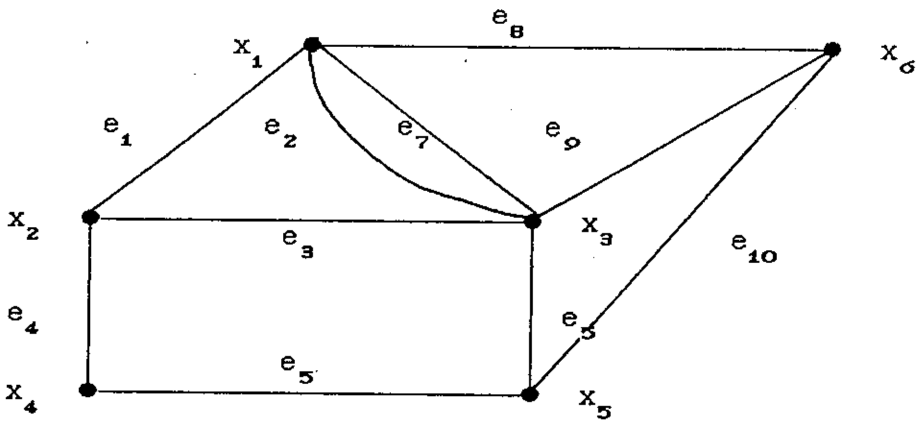


G_2

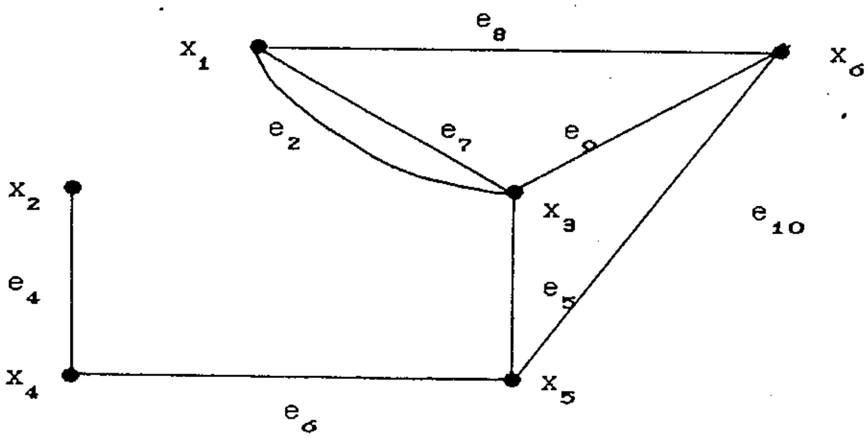
1. Operasi $G = G_1 \cap G_2$.



2. Operasi $G = G_1 \cup G_2$

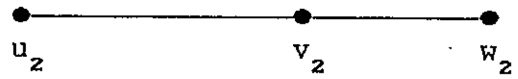


3. Operasi $G = G_1 + G_2$

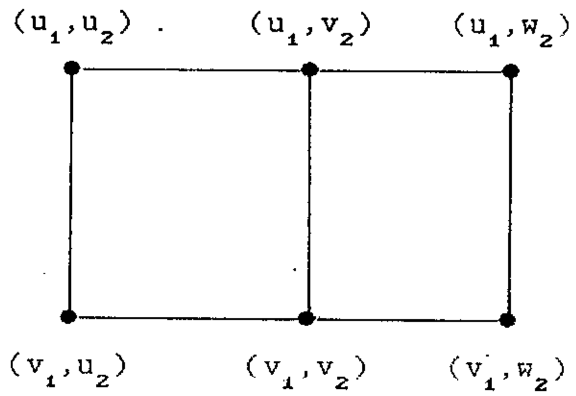


4. Operasi Product dari 2 graph yaitu : $G = G_1 \times G_2$

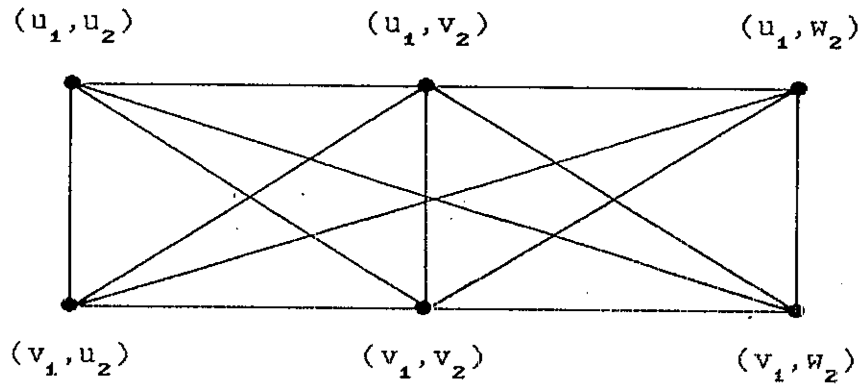
G_1



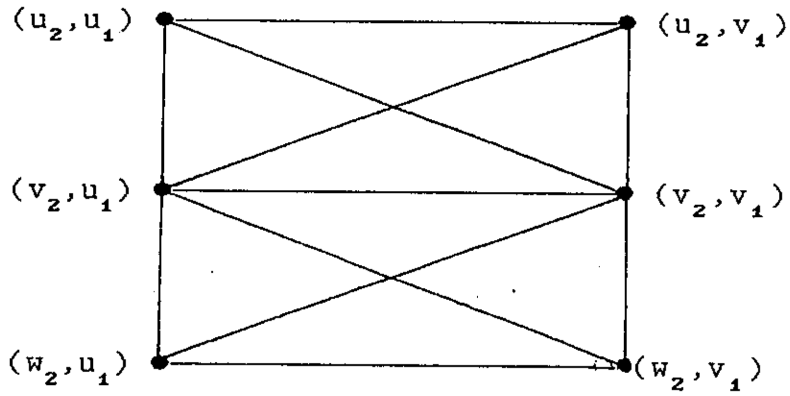
G_2



5. Komposisi dari 2 graph



$G_1\ (G_2)$



G_2 (G_1)

Irisan, gabungan dan Ring Sum dari graph G_1dan G_2 mempunyai sifat komutatif, yaitu :

1. $G_1 \cup G_2 = G_2 \cup G_1$
2. $G_1 \cap G_2 = G_2 \cap G_1$
3. $G_1 + G_2 = G_2 + G_1$

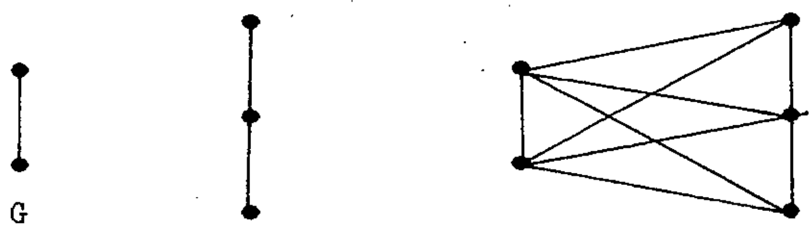
Jika graph G_1 dan G_2 edge-edgenya saling asing, maka $G = G_1 \cap G_2$ adalah graph kosong yaitu graph yang tidak mempunyai edge dan $G_1 + G_2 = G_1 \cup G_2$.

Jika graph G_1 dan G_2 vertex-vertexnya saling asing, maka $G_1 \cap G_2$ adalah kosong.

Dan jika $G_1 = G_2$ maka $G_1 \cap G_2 = G_1 \cup G_2$ dan $G_1 + G_2$ adalah graph kosong.

Join dari 2 graph G dan H yang saling asing dalam suatu graph yang didapat dengan menghubungkan setiap vertex G ke vertex H dan mengandung $G \cup H$. Ditulis $G \vee H$.

Contoh :



G H $G \vee H$