B A B II

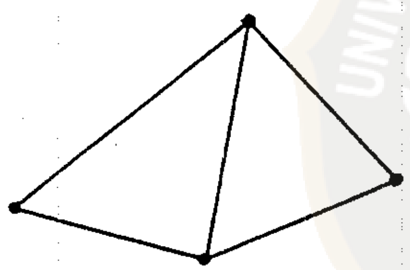
MATERI PENUNJANG

1.2 Teori Graph

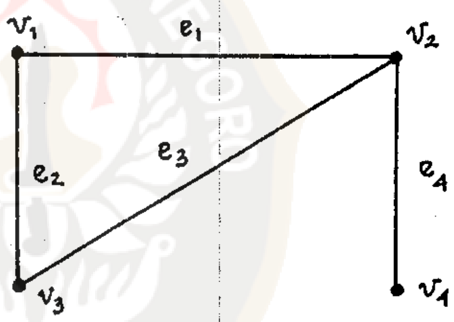
Definisi 1

Suatu Graph G = (V,E) adalah suatu sistem matematika yang terdiri dari himpunan berhingga tak kosong $V = \{ v_1, v_2, v_3, ..., v_n \}$ yang merupakan himpunan titik dan himpunan edge $E = \{ e_1, e_2, e_3, ..., e_n \}$ yang merupakan himpunan garis, dengan relasi dari E yang merelasikan V_i ke V_j .

Contoh 3



gambar 5a



gambar 5b

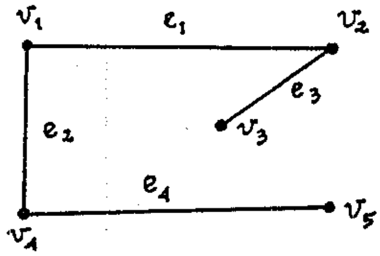
keterangan :

- Gambar 5a merupakan graph yang tidak berlabel atau bertanda
- Gambar 5b merupakan graph yang berlabel atau bertanda dengan $V = \{ v_1, v_2, v_3, v_4 \}$ dan $E = \{ e_1, e_2, e_3, e_4 \}$

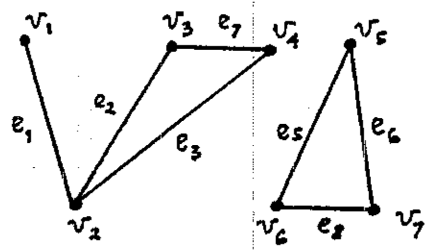
Definisi 2

Graph terhubung adalah graph yang untuk setiap dua titik didalamnya minimal mempunyai sebuah lintasan yang menghubungkan kedua titik tersebut.

Contoh 4



gambar 6a



gambar 6b

keterangan :

- Gambar 6a adalah sebuah graph terhubung
- Gambar 6b adalah sebuah graph tak terhubung

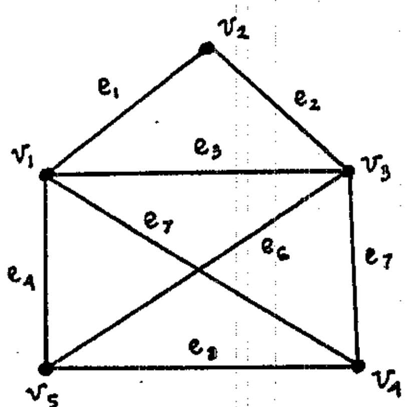
Definisi 3

Suatu lintasan (path) dari suatu graph $G = (V,E)$ adalah barisan edge dimana *terminal verteks*nya bertemu dengan *initial verteks.*

Initial verteks adalah verteks awal dari suatu edge dan *terminal verteks* adalah verteks akhir dari suatu edge.

Lintasan sederhana adalah lintasan dengan terminal verteknya tidak bertemu dengan initial verteks.

Contoh 5



gambar 7. Graph $G=(V,E)$

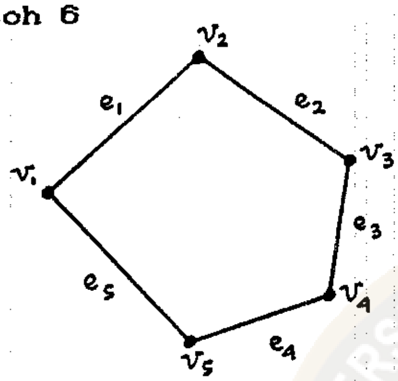
lintasan adalah $\{ e_3, e_7, e_8, e_4 \}$

lintasan sederhana adalah $\{ e_1, e_2, e_7, e_8 \}$

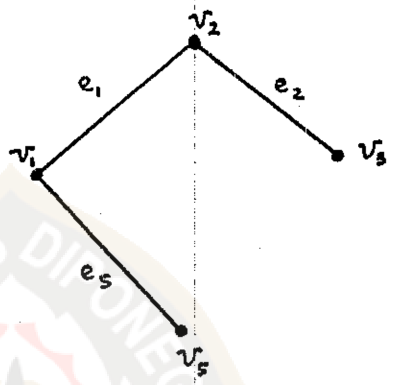
Definisi 4

Graph $G_1 = (V_1, E_1)$ disebut subgraph dari graph $G = (V,E)$ jika V_1 subset dari V dan E_1 subset dari E

Contoh 6



gambar 8a



gambar 8b

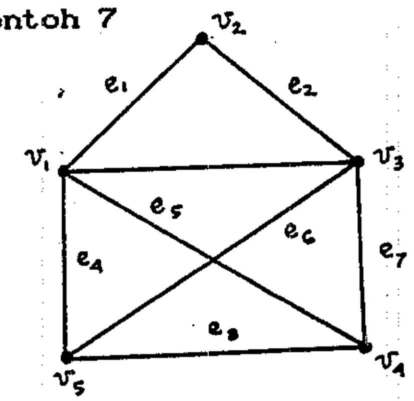
Gambar 8a adalah graph dengan verteks $V = \{v_1, v_2, v_3, v_4, v_5\}$ dan $E = \{e_1, e_2, e_3, e_4, e_5\}$

Gambar 8b, G_1 subgraph G dengan verteks $V = \{v_1, v_2, v_3, v_5\}$ dan $E = \{e_1, e_2, e_5\}$

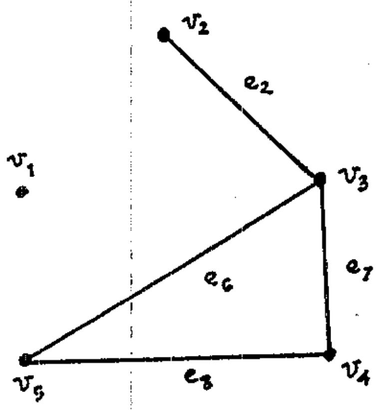
Definisi 5

Spanning subgraph adalah subgraph yang memuat semua verteks dari graphnya.

Contoh 7



gambar 9a. graph G



gambar 9b. Subgraph G

Gambar 9a. Graph $G = (V.E)$, $V = \{v_1, v_2, v_3, v_4, v_5\}$ dan

$E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$

Gambar 9b. Subgraph $G, G_1 = (V_1, E_1)$ dan $V=\{v_1, v_2, v_3, v_4, v_5\}$

dan $E_1 = \{e_2, e_7, e_8, e_6\}$

$G_1 = (V_1, E_1)$ spanning tree dari $G = (V.E)$ bhb $V_1 \in V$ dan $E_1 \subset E$.

POHON (TREE)

Yang disebut dengan struktur tree adalah salah satu bentuk khusus dari suatu graph. Misalkan diberikan suatu graph terhubung, maka untuk sembarang pasang verteks pada graph tersebut dapat dicari suatu lintasan elementer yang menghubungkan kedua verteks tersebut.

Definisi 6

Suatu graph sederhana dimana setiap pasangan verteks yang berbeda dihubungkan oleh hanya satu lintasan yang elementer disebut suatu *tree*.

Karena setiap pasang verteks pada tree hanya dihubungkan oleh satu lintasan, maka pada tree tidak mungkin terdapat suatu sirkuit (lintasan yang tertutup)

Definisi 7

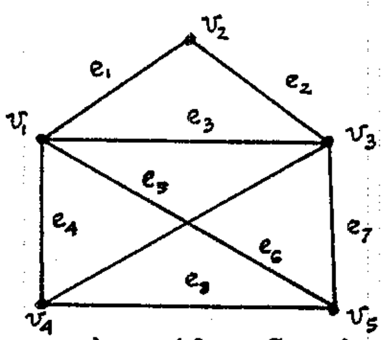
Sirkuit adalah graph terhubung yang setiap verteksnya berderajat dua.

Walk adalah deretan bergantian verteks dan edge yang dimulai dan diakhiri dengan verteks.

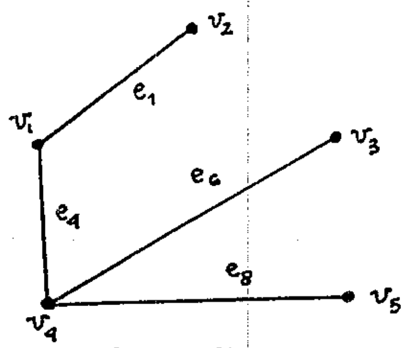
Path adalah suatu barisan edge dan verteks yang berhingga yang tidak boleh diulang.

Contoh 8

Diberikan graph G = (V,E)



gambar 10a. Graph G



gambar 10b. Tree

Walk dari graph tersebut $V_2, e_2, V_3, e_7, V_5, e_8, V_4, e_6, V_3, e_3, V_1$

Path dari graph tersebut $V_2, e_2, V_3, e_7, V_5, e_8, V_4, e_4, V_1$

Sirkuit dari graph tersebut $V_1, e_1, V_2, e_2, V_3, e_3, V_1$

Misalkan suatu graph mempunyai n buah verteks Spanning-tree dari G adalah tree dimana verteksnya adalah verteks dari G, dan edgenya merupakan sebagian dari edge graph. Suatu tree dengan n verteks akan mempunyai sejumlah n-1 edge. Tetapi perlu diperhatikan bahwa suatu graph dengan n buah verteks dan n-1 edge belum tentu membentuk tree. Tetapi bila graph tersebut tidak memuat sirkuit, maka dapat dikatakan bahwa graph tersebut adalah tree.

Theorema 1

Suatu tree dengan n buah verteks akan mempunyai n-1 edge

Bukti :

Pembuktian dengan cara induksi matematik. Jelas bahwa theorema ini berlaku untuk n = 1. Untuk n > 1 misalkan graph G adalah tree dengan n verteks, maka G pasti akan mempunyai satu verteks ujung x,dan e adalah edge yang bertemu dengan x. Misalkan G´ adalah graph baru yang diperoleh dari G dengan menghapuskan verteks x dan edge e . Jelas bahwa G ini adalah graph terhubung juga, dan karena G tidak mempunyai sirkuit, G´ juga tidak mempuyai sirkuit. Karena G´ tidak mempunyai sirkuit danterhubung, maka dapat disimpulkan bahwa G´ adalah suatu tree. Dan karena jumlah verteks dari G´ adalah n - 1, maka menurut langkah induksi G´ mempunyai sejumlah (n-1)-1 edge atau n-2 edge. Jadi G akan mempunyai (n-2)+1 = n - 1 edge.

Terbukti.

Theorema 2

Misalkan G adalah suatu graph dengan n verteks dan tepat n-1 edge. Bila G tidak memuat sirkuit, maka G adalah tree.

Bukti

Karena telah diketahui G tidak mempunyai sirkuit, maka tinggal ditunjukkan bahwa G adalah terhubung, maka bila hal ini bisa ditunjukkan maka jelas G adalah suatu tree. Misalkan G adalah tidak terhubung. Maka G mempunyai paling sedikit $k \geq 2$ komponen terhubung. Sebut $G_1,\ldots,G_k$ adalah komponen-komponen terhubung dari G dengan masing-masing komponen mempunyai $n_1,\ldots,n_k$ verteks, dimana $n_1+\ldots+n_k = n$. Setiap dari komponen tidak mempunyai sirkuit, jadi menurut theorema diatas masing-masing komponen membentuk suatu tree. Jadi jumlah edge pada masing-masing komponen adalah n - 1 untuk i = 1,...,k. Dengan demikian jumlah seluruh edge pada graph G adalah

$$(n_1 - 1) + (n_2 - 1) + \ldots + (n_k - 1) = n-k \leq n-2$$

Tetapi hal ini bertentangan dengan pemisalan bahwa G mempunyai n-1 edge. Jadi pemisalan bahwa G tak terhubung adalah tidak benar. Dan G adalah graph yang terhubung tidak mempunyai sirkuit. Dengan demikian terbukti bahwa G adalah suatu tree.

Definisi 8

Derajat (degree) dari verteks V_1 dinotasikan $d^-(V_i)$ adalah banyaknya edge yang menuju verteks V_i dan $d^+(V_i)$ adalah banyaknya edge yang keluar dari verteks V_i. Dan in-degree sama dengan $d^-(V_i)$.

Contoh 8 dari gambar 10a. Degree dari graph tersebut adalah $d(V_1) = 4$ dan $d(V_2) = 2$

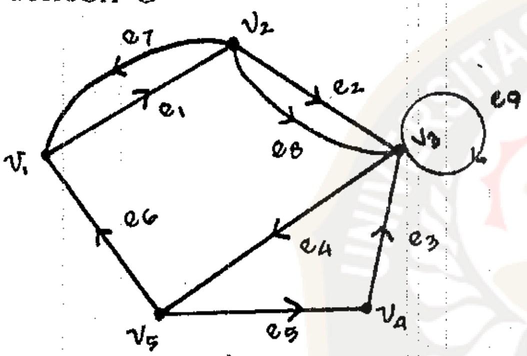
Misalkan suatu graph mempunyai N buah verteks, spanning-tree dari G adalah tree dimana verteksnya adalah verteks dari G, dan edgenya merupakan sebagian dari edge graph. Suatu tree dengan n verteks akan mempunyai sejumlah n-1 edge. Tetapi perlu diperhatikan suatu graph dengan n buah verteks dan n-1 edge belum tentu membentuk tree. Tetapi bila graph tersebut tidak memuat sirkuit, maka dapat dikatakan bahwa graph tersebut adalah tree

Definisi 9

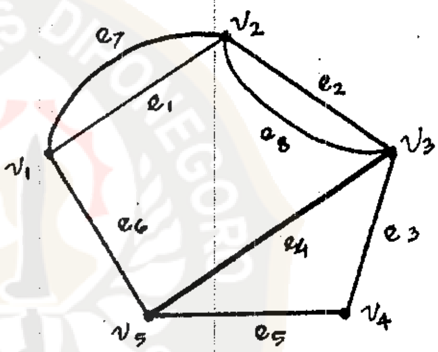
Graph berarah (*Directed graph* = *Digraph*) adalah graph dimana setiap edge telah diberikan arah. Suatu graph berarah mempunyai verteks khusus yaitu root dengan in-degree 0.

Graph tidak berarah adalah graph dimana setiap edge tidak diberikan arah.

Contoh 9



gambar 11 a



gambar 11b

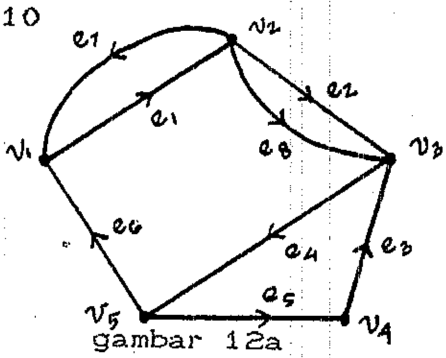
Gambar 11a. Graph berarah

Gambar 11b. Graph tak berarah

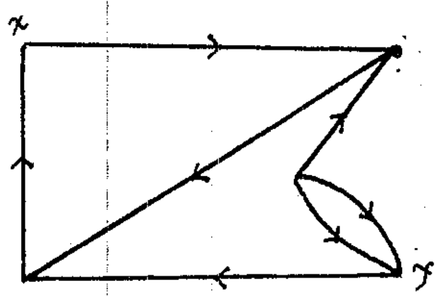
Definisi 10

Suatu graph terhubung terhubung dikatakan *terhubung kuat (Stronglly Connected)* jika hanya jika ada lintasan yang menghubungkan setiap pasang verteks

Contoh 10



gambar 12a



gambar 12b

Gambar 12a. menunjukkan graph yang terhubung erat

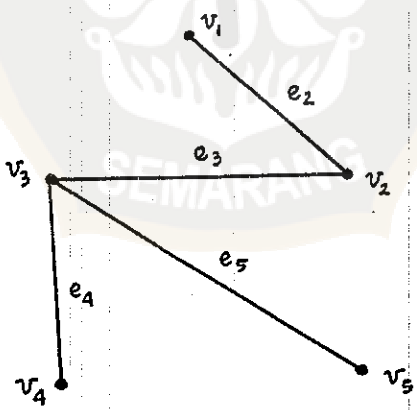
Gambar 12b. menunjukkan graph yang tidak terhubung erat, karena tidak ada lintasan dari x ke y

Definisi 11

Jika dimisalkan garis e_k menghubungkan titik v_i dengan v_j atau dituliskan $e_k = (v_i, v_j)$ maka dikatakan garis e_k insiden terhadap titik v_i dan juga insiden terhadap titik v_j atau dikatakan titik v_i dan v_j insiden terhadap garis e_k, dan titik v_i, v_j dikatakan adjacent(*terhubung langsung*).

Pada gambar 13 garis e_5 adalah contoh insiden terhadap titik V_3 dan titik V_5. dan titik V_3 dan V_5 dikatakan adjacent.

Contoh 11



gambar 13

Definisi 12

Branch adalah sebuah edge yang terdapat dalam spanning-tree T

Chord adalah sebuah edge yang tidak terdapat dalam spanning-tree, tetapi berada dalam graph G

Definisi 13

Vertek V_i dan V_j dihubungkan dengan edge e_k disebut *endverteks dari* e_k

Sebuah edge mempunyai verteks yang sama sedangkan keduanya endverteks disebut *self-loop*

Contoh 12

Gambar 11 a edge e_9 adalah merupakan self-loop dan V_4,V_5 endverteks dari e_5.

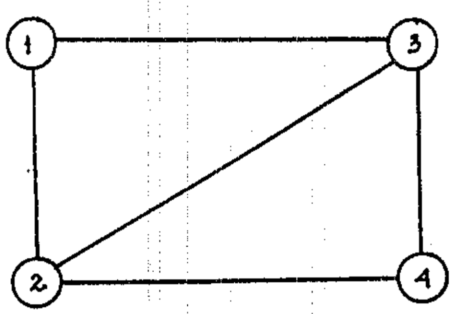
2.1.1 GRAPH TERHUBUNG TAK BERARAH

Definisi 14

Spanning-tree dari sebuah graph adalah subgraph yang memuat semua verteks dari graph yang merupakan tree

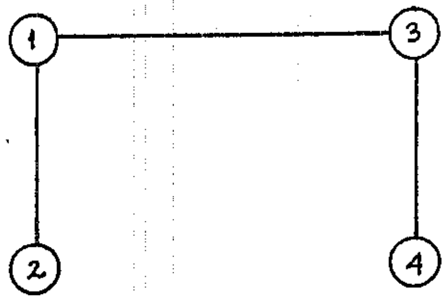
COntoh 12

Sebuah graph G terhubung tak berarah dengan himpunan verteks V = { 1,2,3,4 } dan himpunan edge E = { (1,2),(1,3),(2,4),(2,3),(3,4) } graphnya sebagai berikut :



gambar 14

sebuah spanning tree dari graph tersebut adalah ;



gambar 15

Pada setiap graph terhubung selalu dapat dicari spanning-treenya atau dengan perkataan lain, setiap graph terhubung mempunyai paling sedikit satu spanning-tree.

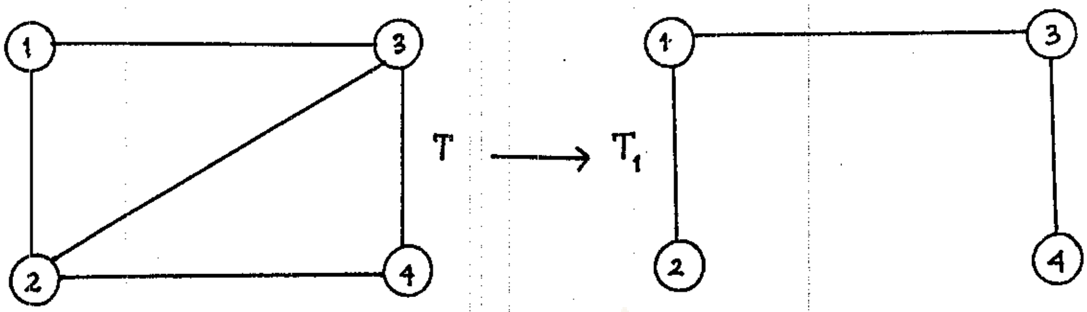
Pada contoh dapat dilihat bahwa spanning-tree dari suatu graph, pada umumnya tidaklah tunggal. Dengan melakukan *penukaran edge (edge Exchange)* akan diperoleh spanning-tree yang lain dari graph terhubung.

PROSEDUR PENUKARAN EDGE (EDGE EXCHANGE)

Pertama yang dilakukan adalah membuat sebuah spanning-tree dari graph terhubung tersebut. Anggaplah telah dimiliki sebuah spanning-tree dari sebuah graph terhubung (gambar 14), maka untuk memperoleh spanning-tree yang lain dapat dilakukan prosedur penukaran edge sebagai berikut :

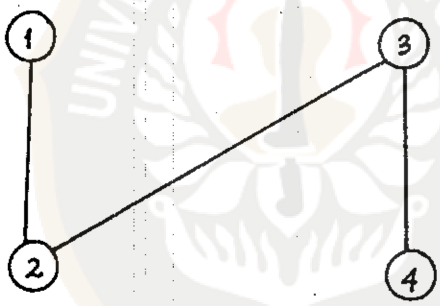
- Langkah 1 : menambahkan sebuah *chord* pada spanning-tree awal
- Langkah 2 : menghapus *branch* agar tidak terjadi sirkuit.

Prosedur inilah yang disebut *penukaran edge (Edge Exchange)*, sesuai dengan prosedur tersebut, maka dapat dicari semua spanning-tree dari graph G pada contoh sebelumnya. Prosedur ini selalu bertumpu pada gambar 14 dan 15 yang disebut T_1 (spanning-tree pertama), yaitu :



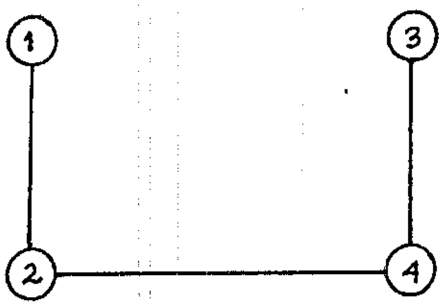
gambar 16. graph T

- Dengan menambahkan *chord* (2,3) dan menghapus *branch* (1,3) dari T_1 maka spanning-tree kedua adalah :



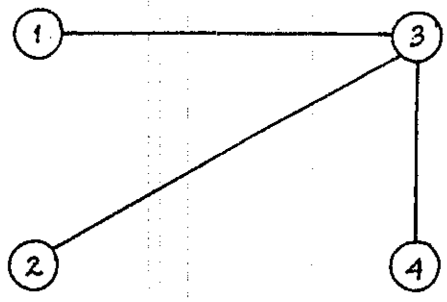
gambar 17. T_2

- Dengan menambahkan *chord* (2,4) dan menghapus *branch* (1,3) dari T_1 maka spanning-tree ketiga adalah :



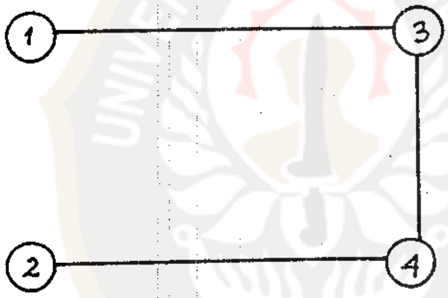
gambar 18. T_3

- Dengan menambahkan *chord* (2,3) dan menghapus *branch* (1,2) dari T_1 maka spanning-tree keempat adalah :



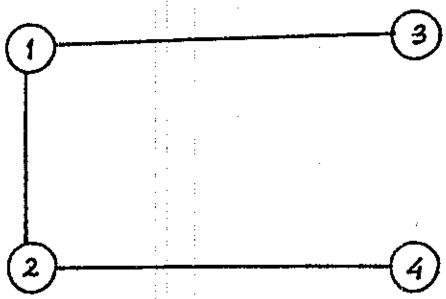
gambar 19. T_4

- Dengan menambahkan *chord* (2,4) dan menghapus *branch* (1,2) dari T_1 maka spanning-tree kelima adalah :



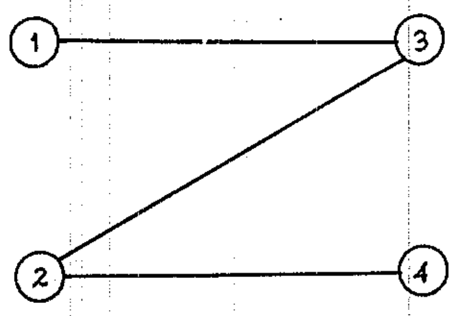
gambar 20. T_5

- Dengan menambahkan *chord* (2,4) dan menghapus *branch* (3,4) dari T_1 maka spanning-tree keenam adalah :



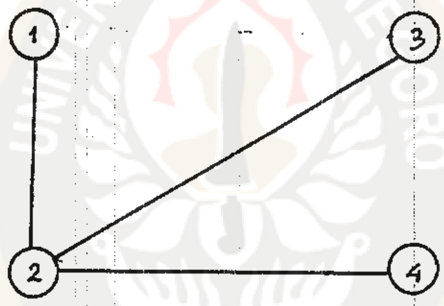
gambar 21. T_6

- Dengan menambahkan *chord-chord* (2,3),(2,4) dan menghapus *branch-branch* (1,2),(3,4) dari T_1 maka spanning-tree ketujuh adalah :



gambar 22. T_7

- Dengan menambahkan *chord-chord* (2,3),(2,4) dan menghapus *branch-branch* (1,3),(3,4) dari T_1 maka spanning-tree kedelapan adalah :



gambar 23. T_8

2.1.2 GRAPH TERHUBUNG BERARAH

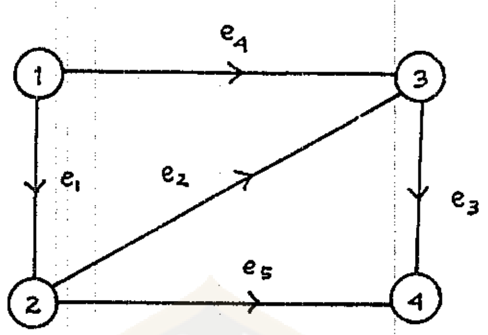
Sebuah graph terhubung berarah mempunyai spanning-tree jika graph tersebut merupakan graph terhubung *quasy-strongly.*

Untuk mendapatkan spanning-tree dari sebuah graph terhubung, terlebih dahulu diambil sebuah verteks sebagai *root* pada graph tersebut, lalu dibentuk spanning-treenya.

Untuk mendapatkan spanning-tree yang lain pada graph terhubung berarah, caranya sama denga graph terhubung tak berarah, dengan memakai prosedur *penukaran edge (Edge*

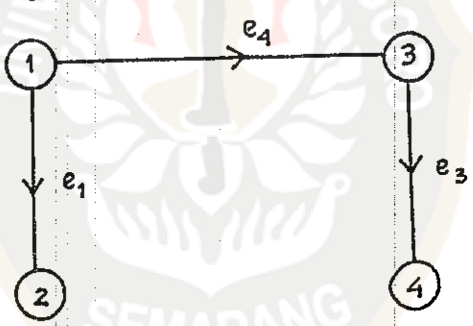
Exchange).

Misalkan graph terhubung berarah sesuai dengan gambar berikut :



gambar 24. Graph T

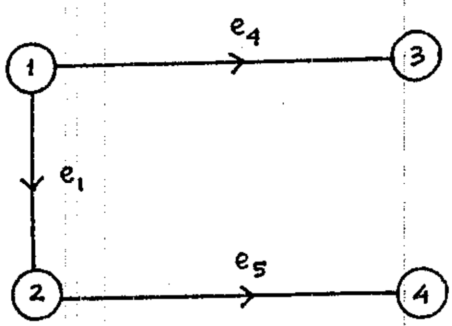
Graph T terhubung berarah, dengan verteks sebagai *root*, maka spanning-treenya adalah :



gambar 25. T^1

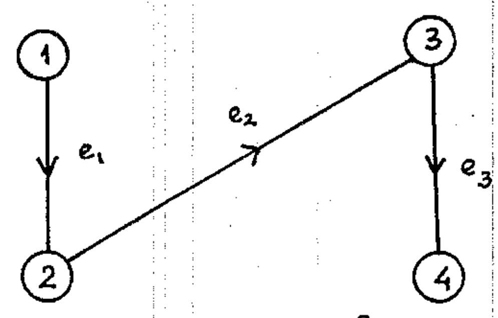
Sedangkan spanning-tree yang lain dapat dibuat dengan prosedur *Edge Exchange* yaitu :

- Dengan menambahkan *chord* (2,4) dan menghapus *branch* (3.4) dari T^1 maka didapat spanning-tree kedua adalah :



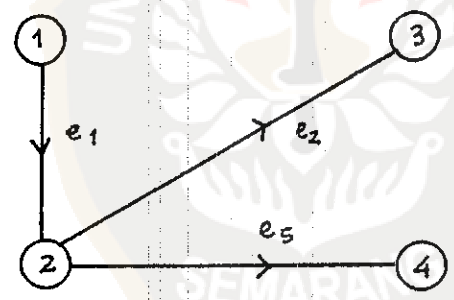
gambar 26. T^2

- Dengan menambahkan *chord* (2,3) dan menghapus *branch* (1.3) dari T^1 maka didapat spanning-tree ketiga adalah :



gambar 27. T^3

- Dengan menambahkan *chord-chord* (2,3),(2,4) dan menghapus *branch-branch* (1,3),(3,4) dari T^1 maka spanning-tree keempat adalah :



gambar 28. T^4

2.2 TEORI MATRIKS

Definisi 15

Matriks adalah susunan persegi panjang dari bilangan bilangan yang diatur dalam baris dan kolom. Matriks ditulis sebagai berikut :

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

Susunan diatas disebut sebuah matriks m kali n (mxn) karena memiliki m baris dan n kolom . Matriks lazimnya dinotasikan dengan sebuah huruf besar yang dicetak tebal (**A**,**B** dan seterusnya), dan elemen-elemen dinotasikan dengan huruf kecil yang dicetak miring (a_{ij}, b_{ij} dan seterusnya) dimana indeks i menyatakan baris ke-i dan indeks j menyatakan kolom ke-j dari elemen tersebut.

Operasi Pada Matriks

a. Penjumlahan Matriks

Definisi 16

Jika **A** = (a_{ij}) dan **B** = (b_{ij}), matriks berukuran sama, maka **A** + **B** adalah suatu matriks **C** = (c_{ij})

dimana : $c_{ij} = a_{ij} + b_{ij}$, untuk setiap i dan j.

Contoh 13

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 2 & 5 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 0 & 3 \\ 1 & 4 \end{bmatrix}$$

maka :

$$\mathbf{A} + \mathbf{B} = \begin{bmatrix} 1 & 2 \\ 2 & 5 \end{bmatrix} + \begin{bmatrix} 0 & 3 \\ 1 & 4 \end{bmatrix}$$

$$= \begin{bmatrix} 1 + 0 & 2 + 3 \\ 2 + 1 & 5 + 4 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 5 \\ 3 & 9 \end{bmatrix}$$

a. Perkalian Matriks

Definisi 17

Jika **A** = (a_{ij}) berukuran mxr dan B = (b_{ij}) berukuran rxn maka perkalian **AB** adalah suatu matriks **C** = (c_{ij}) berukuran mxn dimana :

$$c_{ij} = a_{i1} b_{1j} + a_{i2} b_{2j} + \dots + a_{ir} b_{rj}$$

untuk setiap i = 1,2,...,n dan j = 1,2,...,m.

Contoh 14

A = [3 2 0] ordo 1x3 $\quad$ **B** = $\begin{bmatrix} 1 \\ 4 \\ 2 \end{bmatrix}$ ordo 3x1

maka ukuran matriks **C** adalah 1x1

$$c_{11} = a_{11}\ b_{11} + a_{12}\ b_{21} + a_{13}\ b_{31}$$
$$= 1.3 + 2.4 + 0.2$$
$$= 11$$

atau

$$\mathbf{C} = \mathbf{AB} = [\ 3\ 2\ 0\ ] \begin{bmatrix} 1 \\ 4 \\ 2 \end{bmatrix} = 3.1 + 2.4 + 0.2 = 11$$

Transpose Dari Suatu Matriks

Definisi 18

Jika suatu matriks **A** = (a_{ij}) berukuran mxn maka transpose dari **A** adalah matriks $\mathbf{A}^t$ berukuran nxm yang didapatkan dari **A** dengan menuliskan baris ke-i dari **A** dimana i = 1,2,3,...,m sebagai kolom ke-i dari A^t.

Contoh 15

$$A = \begin{bmatrix} 1 & 3 \\ 2 & 5 \end{bmatrix}$$

maka bila baris 1 ditulis sebagai kolom 1 dan baris 2 ditulis sebagai kolom 2 maka A^t adalah :

$$A^t = \begin{bmatrix} 1 & 2 \\ 3 & 5 \end{bmatrix}$$

Matriks Identitas

Definisi 19

Matriks bujursangkar yang semua elemen diluar diagonal utama adalah nol dan semua elemen diagonal utamanya adalah 1.

Dengan perkataan lain (u_{ij}) adalah matriks identitas bila $u_{ij} = 1$ untuk $i = j$, dan $u_{ij} = 0$ untuk $i \neq j$, ditulis dengan I atau I_n dimana n menunjukkan ukuran matriks bujur sangkar tersebut.

Contoh 16

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad I_n = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & & 1 \end{bmatrix}$$

Determinan

Definisi 20

Determinan dari matriks bujursangkar A berordo nadalah jumlah dari semua n ! hasilkali bertanda dari elemem-elemen matriks A tersebut.

$$\det(A) = |A| = \sum \delta(j_1, j_2, \ldots, j_n)\ a_{1j} . a_{2j}, \ldots, a_{n\ jn}$$

Definisi 21

Yang dimaksud dengan sebuah inversi pada suatu permutasi $(j_1, j_2, \ldots, j_n)$ ialah apabila $j_k < j_i$ (j_k mendahului j_i) padahal $j_i < j_k$ $(i, k = 1, 2, \ldots, n)$

Definisi 22

Barisan bilangan-bilangan $(j_1, j_2, \ldots, j_3)$ di mana berlaku $j_1 \neq j_k$, untuk $i \neq k$. $(i$ dan $k = 1, 2, \ldots, n)$ serta j_i salah satu dari bilangan asli $(1, 2, \ldots, n)$ disebut suatu permutasi.

Contoh 17

a. $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ maka terdapat $n! = 2! = 2.1 = 2$ buah hasilkali berikut :

1. $a_{11}\ a_{12}$, permutasi (1,2), banyaknya inversi adalah 0 (permutasi genap) maka $\delta(1,2) = +1$, jadi $+ a_{11}\ a_{12}$
2. $a_{21}\ a_{22}$, permutasi (2,1), banyaknya inversi adalah 1 (permutasi ganjil) maka $\delta(2,1) = -1$, jadi $-a_{12}a_{21}$

b. $A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$ berordo 3 dan terdapat 3! = 3.2.1 = 6 hasilkali berikut :

1. $a_{11}\ a_{22}\ a_{33}$, permutasi (1,2,3) banyaknya inversi 0(+)
2. $a_{12}\ a_{22}\ a_{31}$, permutasi (2,3,1) banyaknya inversi 2(+)
3. $a_{13}\ a_{21}\ a_{32}$, permutasi (3,1,2) banyaknya inversi 2(+)
4. $a_{13}\ a_{22}\ a_{31}$, permutasi (3,2,1) banyaknya inversi 3(-)
5. $a_{12}\ a_{21}\ a_{33}$, permutasi (2,1,3) banyaknya inversi 1(-)
6. $a_{11}\ a_{23}a_{32}$, permutasi (1,3,2) banyaknya inversi 1(-)

Jadi det(A) = $+a_{11}a_{22}a_{33}\ \ +a_{12}a_{23}a_{31}\ \ +a_{13}a_{21}a_{32}$
$-a_{13}a_{22}a_{31}\ \ -a_{12}a_{21}a_{33}\ \ -a_{11}a_{23}a_{32}$

Definisi 23

Suatu matriks bujursangkar A disebut singular apabila det(A) = 0, jika det(A) $\neq$ 0 maka disebut matriks yang non singular.

Definisi 24

Himpunan m buah vektor $\{ V_1, V_2, V_3, \ldots, V_m \}$ bergantung linier bila terdapat skalar-skalar $\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_m$ yang tidak semua nol sedemikian sehingga $\lambda_1 u_1 + \lambda_2 u_2 + \ldots + \lambda_m u_m = 0$ (0 = vektor 0). Didalam hal lain himpunan $\{u_1, u_2, \ldots, u_m\}$ disebut bebas linier, apabila $\lambda_1 u_1 + \lambda_2 u_2 + \ldots + \lambda_m u_m = 0$ hanya terpenuhi oleh $\lambda_1 = \lambda_2 = \ldots = \lambda_m = 0$