BAB II

BEBERAPA PENGERTIAN TENTANG GRAPH

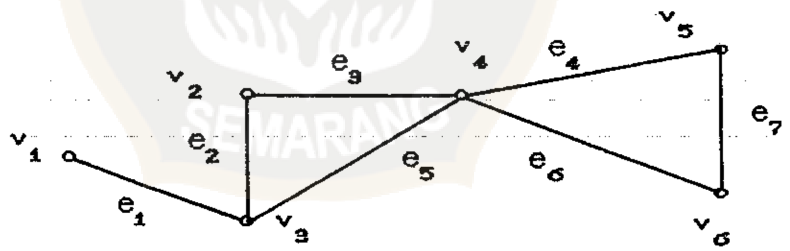
II.1. DASAR-DASAR TEORI GRAPH

Definisi 2.1.1 :

Graph $G(V,E)$ adalah himpunan pasangan berurutan antara himpunan titik (vertices) $V = V(G) = \{v_1, v_2, v_3, \ldots, v_p\}$ dan himpunan garis graph (edge) $E = E(G) = \{e_1, e_2, e_3, \ldots, e_q\}$.

Jika setiap titik dan setiap garis dalam graph G diberi nama, maka disebut *graph berlabel (labeled graph)*, dan jika sebaliknya disebut *graph tidak berlabel (non labeled graph)*. Selanjutnya akan dipakai keduanya.

Contoh :



Gb. 1.

Graph G dengan himpunan

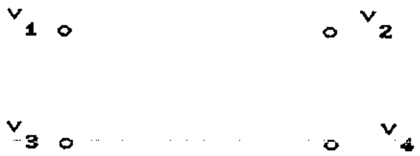
$V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ dan

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

Definisi 2.1.2 :

Jika himpunan garis graph E adalah kosong, maka graphnya menjadi tidak punya garis graph dan disebut *null graph*.

Contoh :

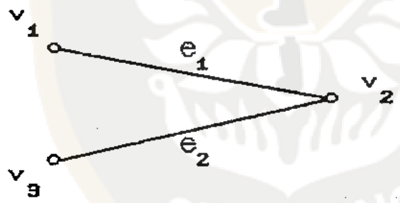


Gb.2. Contoh null graph

Definisi 2.1.3 :

Pada graph G(V,E), v_i *adjacent* v_j jika terdapat e anggota E(G) yang menghubungkan v_i dan v_j.

Contoh :



Gb.3.

Pada Gb.3. v_2 adjacent dengan v_1 dan v_3, sedangkan v_1 dan v_3 tidak adjacent.

Definisi 2.1.4 :

Garis e_i dikatakan insiden dengan titik v_j jika titik v_j merupakan ujung dari beberapa garis e_i.

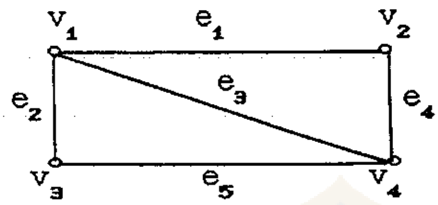
Contoh :

Pada Gb.3. garis e_1 dan e_2 insiden dengan titik v_2.

Definisi 2.1.5 :

Derajat (degree) dari suatu titik v_i adalah banyaknya garis-garis yang insiden dengan titik tersebut, ditulis *deg*(v_i).

Contoh :



Gb. 4

Pada Gb.3. derajat dari v_2 = deg(v_2) = 2, dan deg(v_4) = 3.

Definisi 2.1.6 :

Derajat minimum dari suatu graph G dinotasikan dengan δ(G).

Contoh :

Pada Gb.4, δ(G) = 2.

Definisi 2.1.7 :

Jumlah titik-titik dalam graph G disebut *order*, dan jumlah dari garis-garisnya disebut *size*, atau order dari G adalah |V(G)| dan size dari G adalah |E(G)|.

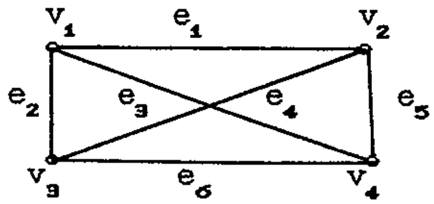
Contoh :

Pada Gb.4. graph G memiliki order = |V(G)| = 4 dan size = |E(G)| = 5.

Definisi 2.1.8 :

Suatu graph dengan order p dan size q disebut *(p,q) graph.*

Contoh :



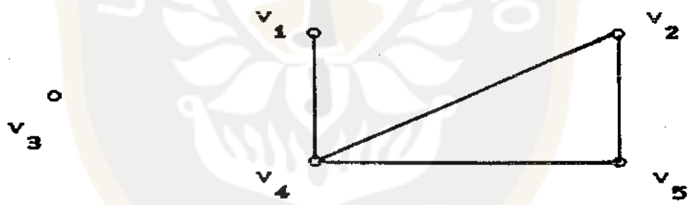
Gb.5. Suatu graph dengan order 4 dan size 6 sehingga dapat ditulis (4,6) graph.

Definisi 2.1.9 :

Suatu titik dengan derajat = 0 disebut *titik terisolir (titik asing)*, sedang titik dengan derajat 1 disebut *titik akhir (end vertex)*.

Contoh :

G :



Gb. 6.

Pada Gb.6. titik v_3 terisolir karena mempunyai derajat 0, dan titik v_1 adalah titik akhir karena mempunyai derajat 1.

Theorema 2.1.1 :

Misal G adalah suatu graph dengan order p dan size q, dengan $V(G) = \{v_1, v_2, \ldots, v_p\}$, maka :

$$\sum_{i=1}^{p} \deg(v_i) = 2q$$

Bukti :

Pada setiap graph, setiap garis pasti insiden dengan dua titik. Sehingga setiap garis memberi kontribusi 2 pada jumlah derajat-derajatnya. Sehingga :

$$\sum_{i=1}^{p} \deg(v_i) = 2q$$

Misal :

Pada Gb.6. graph G memiliki p = 5 dan q = 4

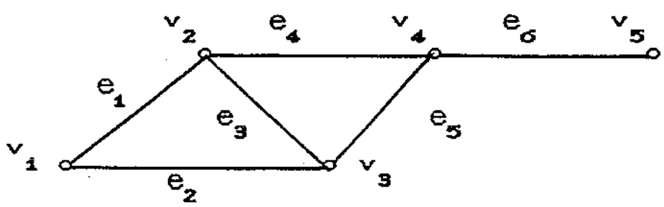
$$\deg(v_1) = 1$$
$$\deg(v_2) = 2$$
$$\deg(v_3) = 0$$
$$\deg(v_4) = 3$$
$$\deg(v_5) = 2$$
$$\sum_{i=1}^{5} \deg(v_i) = 8 = 2q$$

Definisi 2.1.10 :

Suatu *walk* pada graph G adalah suatu deretan (lintasan) yang terdiri atas titik-titik dan garis-garis secara berganti-ganti, dimana setiap garis insiden dengan dua titik terdekat yang terletak disebelah kanan kirinya dan deretan tersebut diawali dan diakhiri pada suatu titik.

Contoh :

G :



Gb. 7.

Pada Gb.7. deretan v_1 e_1 v_2 e_3 v_3 e_5 v_4 e_6 v_5 merupakan walk.

Pada walk, titik-titik dan garis-garis boleh diulang. Apabila titik mula sama dengan titik akhir disebut *walk tertutup*, jika tidak demikian disebut *walk terbuka*.

Definisi 2.1.11 :

Apabila semua garis-garis pada suatu walk tersebut berlainan, maka walk tersebut disebut *Trail (Chain)*.

Contoh :

Pada Gb. 7, v_1 e_2 v_3 e_3 v_2 e_4 v_4 e_5 v_3 adalah trail. Pada trail titik boleh diulang, sedang garis tidak boleh diulang.

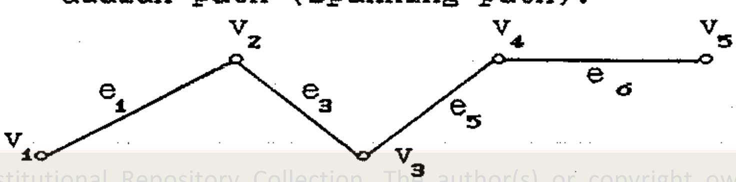
Definisi 2.1.12 :

Apabila pada suatu walk semua titiknya berlainan maka disebut *path*. Jika path tidak terputus dan memuat semua titik dalam graph G, disebut *spanning path*.

Jadi pada path titik-titik dan garis-garis tidak boleh diulang.

Contoh :

Pada Gb.7. deretan v_1 e_1 v_2 e_3 v_3 e_5 v_4 e_6 v_5 adalah path (spanning path).



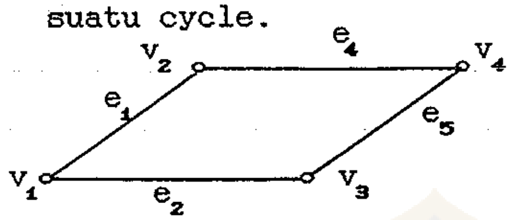
Gb. 8.

Definisi 2.1.13 :

Suatu path yang tertutup disebut *cycle*.

Contoh :

Pada Gb.7, path v_1 e_1 v_2 e_4 v_4 e_5 v_3 e_2 v_1 adalah suatu cycle.

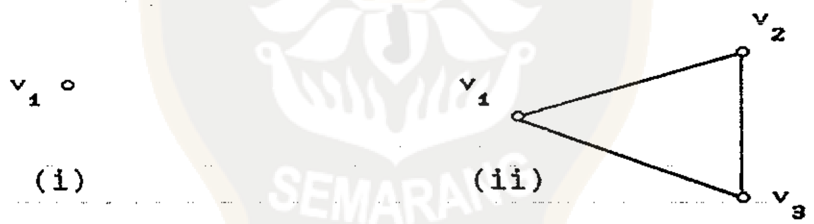


Gb. 9.

Definisi 2.1.14:

Suatu graph G disebut *graph trivial (trivial graph)* jika memiliki order 1, dan disebut *graph nontrivial* jika memiliki order paling sedikit 2.

Contoh :



Gb. 10.

(i). graph trivial (order=1)

(ii). graph non trivial (order=3)

Definisi 2.1.15 :

Suatu graph disebut *graph komplit (Complete graph)* jika setiap titik adjacent dengan setiap titik yang lainnya.

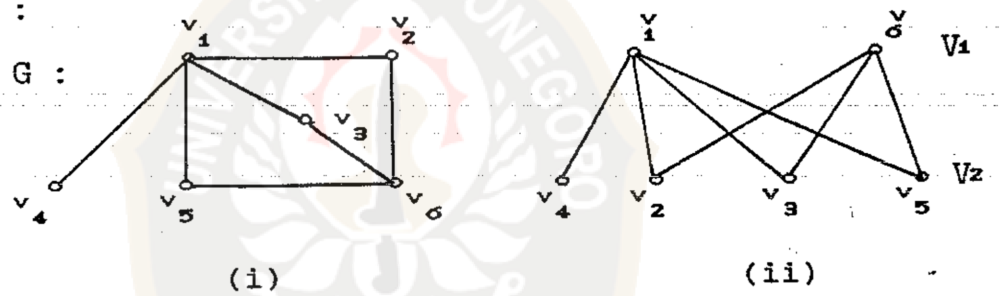
Contoh :

Graph G pada Gb.10 (ii) merupakan contoh graph komplit.

Definisi 2.1.16 :

Suatu graph G(V,E) disebut *Bipartite graph* apabila himpunan titik-titik V dapat dipisahkan atas dua himpunan V_1 dan V_2, dengan $V_1 \cap V_2 = \emptyset$ (V_1 dan V_2 saling asing), sedemikian sehingga jika ada garis e dalam G maka e pasti menghubungkan titik dalam V_1 dan V_2. (Sehingga tidak ada garis yang menghubungkan 2 titik di V_1 dan tidak ada garis yang menghubungkan 2 titik di V_2).

Contoh :



Gb.11

Contoh graph bipartite :

(i). Graph G

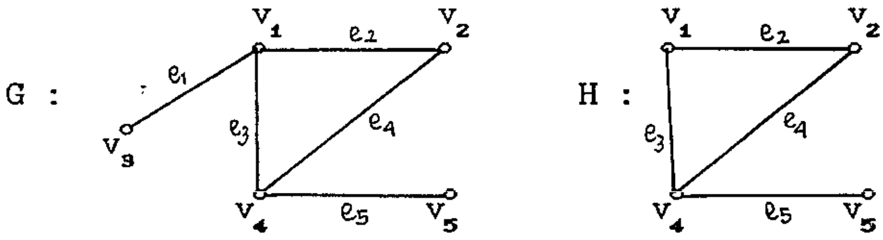
(ii). Graph G dipisah dalam V_1 dan V_2 ($V_1 \cap V_2 = \emptyset$).

II.2. SUBGRAPH

Definisi 2.2.17 :

Suatu graph H adalah sebuah *subgraph* dari graph G jika $V(H) \subseteq V(G)$ dan $E(H) \subseteq E(G)$.

Contoh :



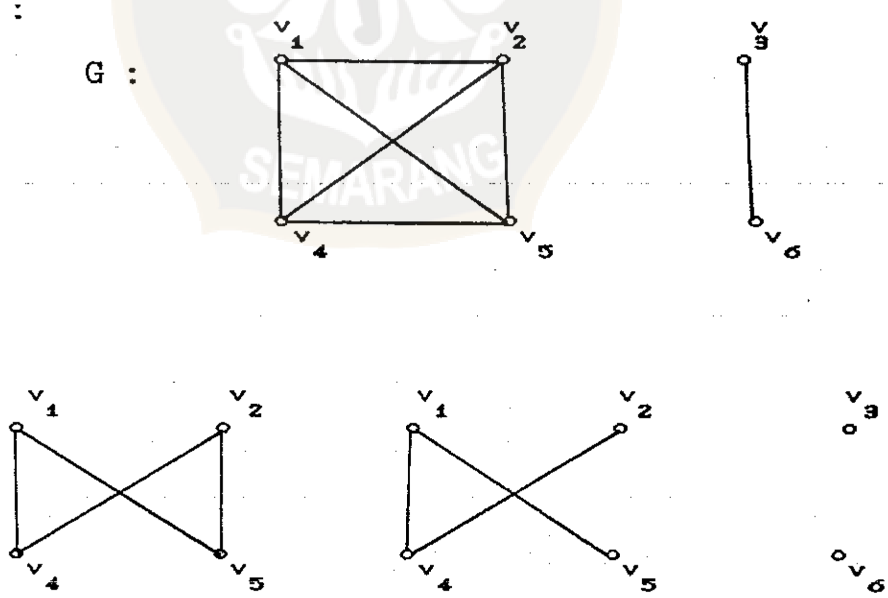
Gb. 12

Pada Gb.12, graph H merupakan subgraph dari graph G.

Definisi 2.2.18 :

Graph H disebut *subgraph murni (proper subgraph)* dari graph G, apabila V(H) atau E(H) merupakan himpunan bagian murni dari G.

Contoh :



Gb.13.

Pada Gb.13, (i) dan (ii) merupakan subgraph murni dari G.

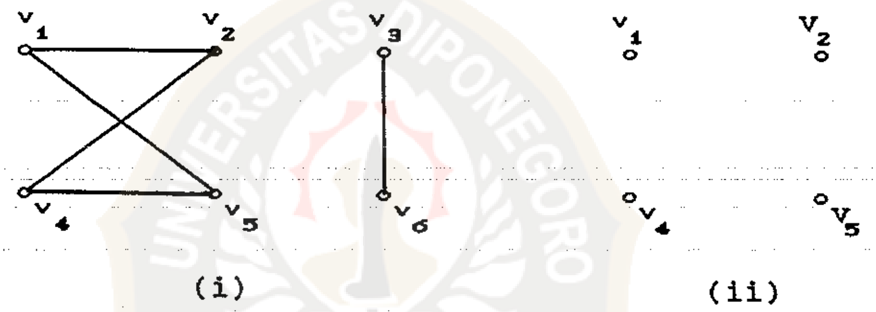
Definisi 2.2.19 :

Graph H disebut *spanning subgraph (subgraph bentangan)* dari graph G apabila V(H) = V(G).

Definisi 2.2.20 :

Graph H disebut *subgraph nol (Null subgraph)* dari graph G apabila himpunan E(H) kosong.

Contoh :



Gb. 14.

Pada Gb.14, (i) merupakan spanning subgraph dari graph G pada Gb.13, dan (ii) merupakan null subgraph.

II.3. GRAPH TERHUBUNG DAN GRAPH TIDAK TERHUBUNG *(CONNECTED AND DISCONNECTED GRAPH)*

Definisi 2.3.21:

Misal v_i dan v_j adalah titik-titik dalam graph G. Kita katakan bahwa v_i *terhubung (connected) ke* v_j jika G memuat path v_i-v_j (path dari titik v_i ke titik v_j).

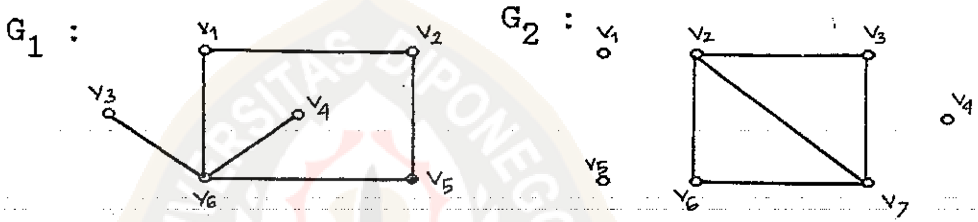
Contoh :

Pada Gb.12, titik $v_1$dan v_5 terhubung dalam G, karena terdapat path v_1 e_3 v_4 e_5 v_5.

Definisi 2.3.22 :

Suatu graph G disebut terhubung bila dan hanya bila setiap dua titik dalam G dihubungkan oleh sekurang kurangnya satu path. Sebaliknya jika ada 2 buah titik atau lebih yang tidak terhubung, maka G disebut graph tidak terhubung (*disconnected*). Graph trivial adalah terhubung.

Contoh :



Gb.15

Pada Gb.15, graph G_1 merupakan graph terhubung sedang graph G_2 adalah tidak terhubung.

Definisi 2.3.23 :

Suatu subgraph H dari graph G adalah *komponen* (*component*) *dari G*, jika H merupakan suatu subgraph terhubung maksimal dari G.

Jumlah komponen dari graph G dinotasikan dengan k(G). Suatu graph G adalah terhubung jika dan hanya jika k(G)=1.

Contoh :

Pada Gb.15 di atas, G_1 adalah terhubung [k(G)=1], sedang G_2 tidak terhubung dengan k(G)=4.

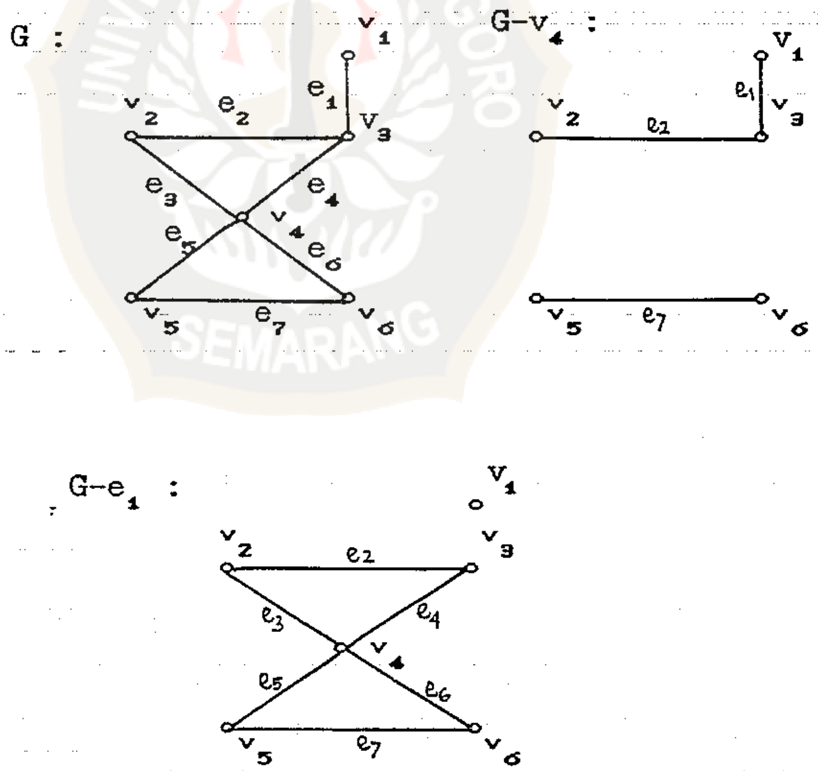
II.4. KETERHUBUNGAN DAN GARIS-KETERHUBUNGAN

(CONNECTIVITY AND EDGE-CONNECTIVITY).

Definisi 2.4.24 :

Misal G suatu graph terhubung. Jika G memuat suatu garis e_i sedemikian hingga $G-e_i$ tidak terhubung, maka e_i disebut *garis perpotongan (bridge) dari G,* dan jika G memiliki titik v_j sedemikian hingga $G-v_j$ tidak terhubung, maka v_j disebut *titik perpotongan (cut-vertex) dari G.*

Contoh :



Gb. 16.

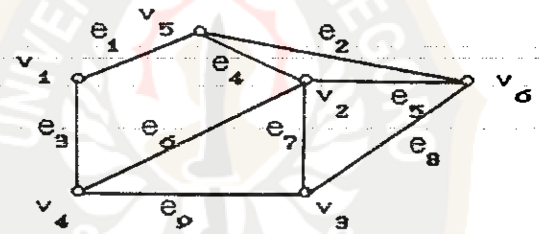
Pada Gb.16 diatas, $G-v_4$ menjadi tidak terhubung, jadi v_4 merupakan titik-perpotongan, dan $G-e_1$ juga tidak terhubung sehingga e_1 merupakan garis-perpotongan .

Definisi 2.4.25 :

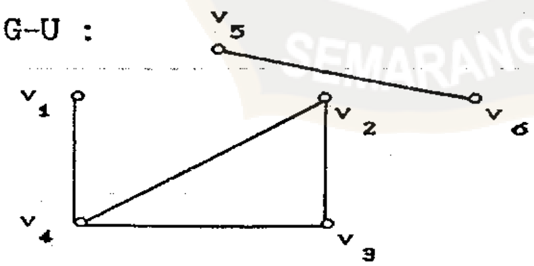
Jika U adalah himpunan bagian (subset) dari himpunan garis pada graph terhubung G sehingga G-U menjadi tidak terhubung, maka U disebut *himpunan garis perpotongan* (*edge cutset*), dan jika S himpunan bagian dari himpunan titik dari G sedemikian hingga G-S tidak terhubung maka S disebut *himpunan titik perpotongan (vertex cutset)* dari G.

Contoh :

G :

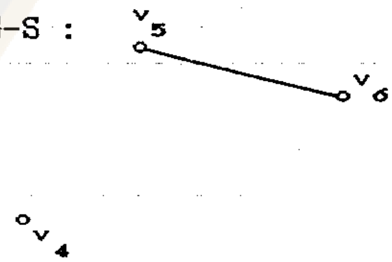


G-U :



(i)

G-S :



(ii)

G-U dengan himpunan garis perpotongan U,

$U = \{e_1, e_4, e_5, e_8\}$

G-S dengan himpunan titik perpotongan S,

$S = \{v_1, v_2, v_3\}$

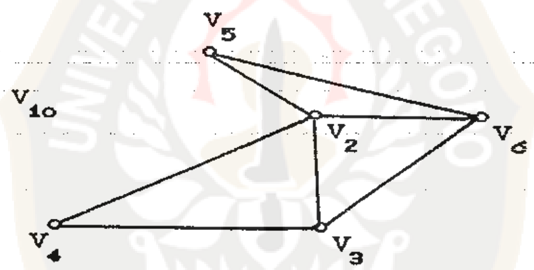
Gb. 17.

Definisi 2.4.26 :

Garis keterhubungan (Edge-connectivity) $\lambda(G)$ dari suatu graph G adalah jumlah minimum dari suatu himpunan garis U dalam graph G sedemikian hingga G-U menjadi tidak terhubung atau trivial. Jika G tidak terhubung atau trivial, maka $\lambda(G) = 0$.

Contoh :

Graph G pada Gb.17 mempunyai $\lambda(G) = 2$, karena dengan penghapusan garis e_1 dan garis e_3 akan diperoleh graph tidak terhubung sebagai berikut :



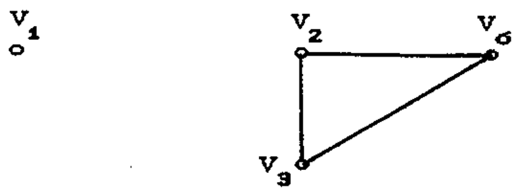
Gb. 18.

Definisi 2.4.27 :

Keterhubungan (Connectivity) K(G) dari suatu graph G adalah jumlah minimum dari titik-titik yang dihapus dari graph G untuk menghasilkan suatu graph tidak terhubung atau trivial. Jika G merupakan graph tidak terhubung atau trivial, maka K(G) = 0.

Contoh :

Graph G pada Gb.17 mempunyai K(G) = 2, karena dengan penghapusan titik v_5 dan titik v_4 akan diperoleh graph tidak terhubung sebagai berikut :



Gb. 19.

Definisi 2.4.28 :

Suatu graph G disebut *n-garis-terhubung (n-edge-connected)* dengan $n \geq 1$ jika $\lambda(G) \geq n$ dan disebut *n-terhubung (n-connected)* jika $K(G) \geq n$.

Contoh :

Pada Gb.17, graph G merupakan graph 2-garis-terhubung, sekaligus graph 2-terhubung.

Theorema 2.4.3 :

Untuk setiap graph G berlaku $K(G) \leq \lambda(G) \leq \delta(G)$

Bukti :

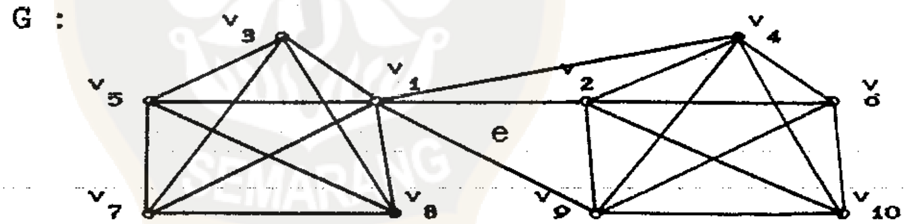
Pertama, akan dibuktikan pertidaksamaan yang kedua. Jika G tidak memiliki garis, maka $\lambda(G)=0$. Sebaliknya, apabila G suatu graph dengan derajat minimal $\delta(G)$, jika semua garis yang insiden pada titik dengan derajat minimal itu dihapus, maka akan dihasilkan suatu graph tidak terhubung. Dalam hal ini $\lambda(G) \leq \delta(G)$.

Untuk menunjukkan pertidaksamaan pertama, berbagai kasus dipertimbangkan. Pertama, jika G tidak terhubung atau trivial, maka $K(G) = \lambda(G) = 0$. Kedua, jika G terhubung dan memiliki suatu garis perpotongan e, maka $\lambda(G) = 1$. Dalam hal ini $K(G) = 1$, karena G memiliki suatu titik perpotongan yang insiden dengan e atau G adalah K_2.

Yang terakhir, anggap G memiliki $\lambda(G) \geq 2$ garis yang harus dihapus agar G tidak terhubung. Jelas, penghapusan $\lambda(G)-1$ garis-garis ini menghasilkan suatu graph dengan bridge $e = v_1v_2$. Selanjutnya, pilih suatu titik yang insiden, yang berbeda dari v_1 atau v_2. Penghapusan titik ini juga menghapus $\lambda(G)-1$ garis dan mungkin lebih banyak. Jika graph hasil adalah tidak terhubung, maka $K(G) < \lambda(G)$. Jika bukan, e adalah suatu garis perpotongan, oleh karenanya, penghapusan v_1 atau v_2 akan menghasilkan suatu graph tidak terhubung atau trivial, sehingga $K(G) \le \lambda(G)$ dalam setiap kasus.

Karena $\lambda(G) \le \delta(G)$ dan $K(G) \le \lambda(G)$ maka $K(G) \le \delta(G)$. Jadi terbukti bahwa $K(G) \le \lambda(G) \le \delta(G)$.□

Contoh :



Gb. 20.

Graph G dengan $K(G)=1$, $\lambda(G)=3$ dan $\delta(G)=4$

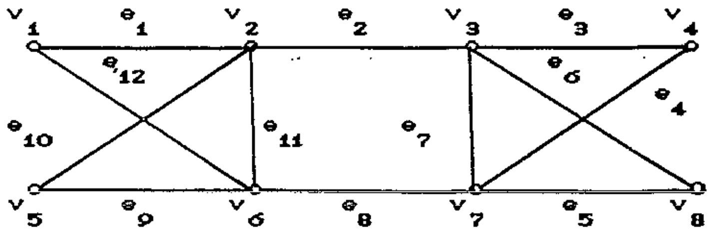
II.5. GRAPH EULER DAN GRAPH HAMILTONIAN

Definisi 2.5.29 :

Suatu graph G disebut *graph euler (Eulerian graph)* jika G memiliki *trail euler (eulerian trail)*. *Trail euler* adalah suatu trail tertutup yang memuat semua titik dari graph.

Contoh :

G :



Gb. 21

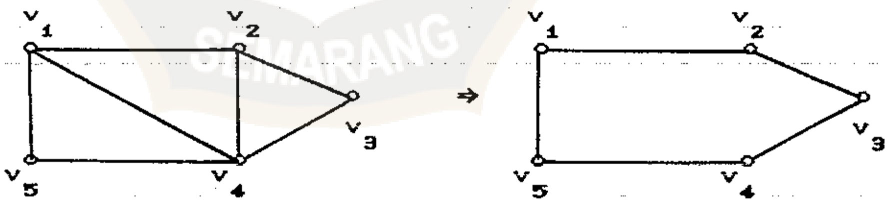
Graph G pada Gb.21 merupakan graph euler, dengan trail euler : $v_1e_1v_2e_2v_3e_3v_4e_4v_7e_5v_8e_6v_3e_7v_7e_8v_6e_9v_5e_{10}v_2e_{11}v_6e_{12}v_1$.

Definisi 2.5.30 :

Suatu graph G disebut *graph hamiltonian (Hamiltonian graph)*, jika graph tersebut mempunyai *spanning cycle*. Spanning cycle adalah suatu cycle yang memuat semua titik, tidak harus semua garis.

Contoh :

G :



Gb. 22

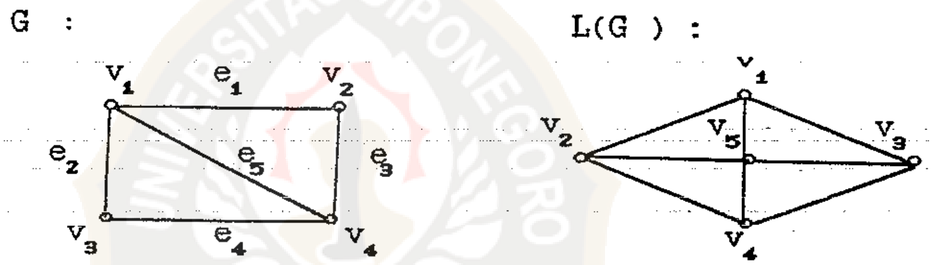
Contoh graph hamiltonian

II.6. LINE GRAPH

Definisi 2.6.31 :

Line graph L(G) dari suatu graph sederhana G adalah graph dimana titik-titiknya berkorespondensi satu-satu dengan garis-garis dari G, dan setiap 2 titik $\in$ L(G) adjacent jika dan hanya jika garis-garis $\in$ G yang bersesuaian adjacent juga.

Contoh :



Gb. 23

Untuk memudahkan dalam menentukan line graph, perlu diketahui banyaknya titik dan garis pada graph G, yang dapat ditentukan sesuai dengan theorema berikut :

Theorema 2.6.3 :

Jika G adalah suatu (p,q) graph yang mempunyai p titik dan q garis, dan setiap titiknya mempunyai derajat masing-masing d_i, maka line graphnya yaitu L(G) mempunyai q titik dan q_L garis, dengan $q_L = -q + 1/2\ \sum d_i^2$.

Bukti :

Karena titik-titik dari L(G) berkorespondensi satu-satu dengan garis-garis dari G, maka apabila G mempunyai q garis, L(G) mempunyai q titik, yang berarti $q = p_L$.

Selanjutnya karena garis-garis yang insiden dengan titik v_i memberikan $\binom{d_i}{2}$ garis kepada q_L, maka :

$$
\begin{aligned}
q_L &= \sum_{i=1}^{p} \binom{d_i}{2} \\
&= \sum_{i=1}^{p} \frac{d_i!}{(d_i-2)!2!} \\
&= 1/2 \sum_{i=1}^{p} \frac{d_i(d_i-1)(d_i-2)\ldots 1}{(d_i-2)(d_i-3)\ldots\ldots 1} \\
&= 1/2 \sum_{i=1}^{p} d_i(d_i-1) \\
&= 1/2 \sum_{i=1}^{p} d_i^2 - 1/2 \sum_{i=1}^{p} d_i \\
&= 1/2 \sum_{i=1}^{p} d_i^2 - 1/2\ .2q \\
&= -q + 1/2 \sum_{i=1}^{p} d_i^2
\end{aligned}
$$

$\therefore$ Jadi jumlah garis dalam $L(G) = -q + 1/2 \sum d_i^2$. □

Catatan :

Untuk setiap graph G, garis-garis yang insiden dengan titik v_i memberikan $\binom{d_i}{2}$ garis pada q_L (line graphnya).

Contoh :

Pada Graph G Gb.23, ada 3 garis yang insiden dengan titik v_1 yaitu garis e_1, e_2 dan e_5, sehingga pada line graph L(G) titik-titik v_1, v_2 dan v_5 memberikan $\binom{3}{2} = 3$ garis.