**MODUL I**

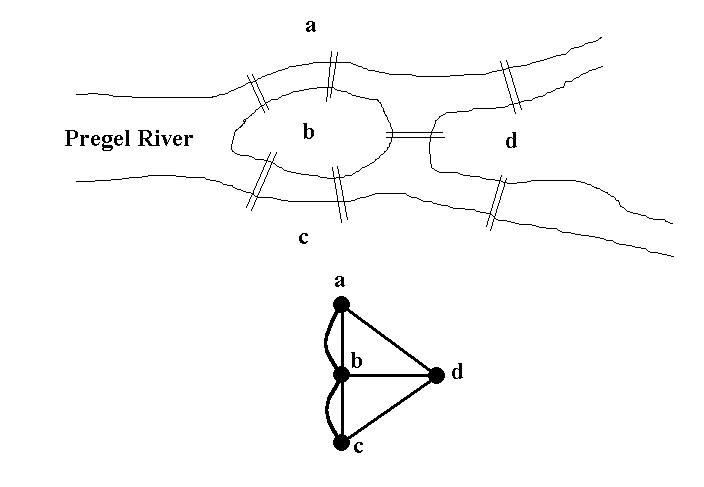
**DASAR-DASAR TEORI GRAPH**

Graph adalah kumpulan dari titik ( node ) dan garis dimana pasangan-pasangan titik ( node ) tersebut dihubungkan oleh segmen garis. Node ini biasa disebut simpul (**verteks**) dan segmen garis disebut ruas (**edge).**

Simpul dan ruas dalam graph dapat diperluas dengan penambahan informasi. Sebagai contoh, simpul bisa diberi nomor atau label dan ruas dapat diberi nilai juga. Perluasan dengan pemberian informasi ini sangat berguna dalam penggunaan graph untuk banyak aplikasi komputer. Contoh, graph dengan simpul yang merepresentasikan kota dan ruas merepresentasikan jarak yang ditempuh diantara kota-kota tsb. (atau harga tiket pesawat antara kota-kota tsb.) , dapat digunakan sebagai “transportation network” untuk mempelajari total jarak (atau harga) dari suatu perjalanan dengan banyak kota pemberhentian. Satu kemungkinan pertanyaan yang bisa muncul adalah “Jalur mana yang terpendek dengan satu atau lebih tempat pemberhentian, yang menghubungkan kota tertentu menuju kota tertentu lainnya dalam transportation network tersebut ?”.

Dalam kehidupan sehari-hari maupun dalam bidang akademis banyak persoalan yang dimodelkan dengan graph. Graph dipakai untuk membantu pemecahan masalah. Dari model graph yang dibuat, suatu masalah dapat dipahami menjadi lebih mudah. Untuk kemudian diturunkan metode pemecahannya.

* 1. **Kelahiran Teori Graph**



**Gbr1**. Jembatan Konigsberg

Menurut catatan sejarah, masalah jembatan Konigsberg adalah masalah yang pertama kali menggunakan graph (th. 1736). Masalah Jembatan Konigsberg adalah : “ **apakah mungkin melalui tujuh buah jembatan masing-masing tepat satu kali. Dan kembali lagi ke tempat semula ?” .** Tak seorangpun yang dapat memecahkan masalah ini. Barulah Euler yang pertama kali menemukan jawabannya. Ia memodelkan masalah dengan memodelkannya ke dalam graph. Daratan (titik-titik yang dihubungkan oleh jembatan) dinyatakannya sebagai simpul (vertex) dan jembatan sebagai sisi. Graph dibuat oleh Euler diperlihatkan pada gambar dibawah atas.

Jawabannya adalah : Orang tidak mungkin berjalan melalui ketujuh jembatan masing-masing satu kali dan kembali ke tempat asal keberangkatan. Singkatnya, tidak terdapat **siklus Euler** pada Graph tersebut.

Graph yang memenuhi kondisi diatas tersebut kemudian dikenal dengan nama **Graph Euler** dan perjalanannya disebut perjalanan euler.

Perjalanan Euler adalah :

Perjalanan dari suatu simpul kembali ke simpul tersebut dengan melalui setiap ruas tepat satu kali.

Perjalanan Euler akan terjadi, jika :

- Graf terhubung.

- Banyaknya ruas yang datang pada setiap simpul adalah genap

* 1. **Graph secara Formal**

Definisi Graf Graf *G* (*V*, *E*), adalah koleksi atau pasangan dua himpunan

(1) Himpunan *V* yang elemennya disebut *simpul* atau *titik*, atau *vertex*, atau *point*, atau *node*.

(2) Himpunan E yang merupakan pasangan tak terurut dari simpul, disebut *ruas* atau *rusuk*, atau *sisi*, atau *edge*, atau *line*.

Banyaknya simpul (anggota V) disebut ***order***Graph G, sedangkan banyaknya ruas (anggota E) disebut ***ukuran*** *(size)* Graph G

e4

e3  1 e1 2 e2 3 e5 4 9

e10 e11 e12 e13

e6 e7 e8 e9 8

e14

5 6 7 e16

e15 10

**Gbr 2.** Graph *G*

Pada Gbr 2, *G* (V,E)adalah graph dengan

*V* = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

*E* = {e1, e2, e3, e4, e5, e6, e7, e8, e9, e10, e11, e12, e13, e14, e15, e16 }

Ruas

e1 = (1,2)

e2 = (2,3)

e3 = (1,1)

e4 = (2,4)

e5 = (3,4)

e6 = (1,5)

e7 = (1,6)

e8 = (2,6)

e9 = (2,7)

e10 = (3,8)

e11 = (4,8)

e12 = (4,7)

e13 = (4,7)

e14 = (7,8)

e15 = (5,6)

e16 = (7,10)

Pada Graph G (gbr.2) , Order = 10 dan Ukuran = 16

Pada Graph *G* (gbr.2), ruas e12 = (4,7) dan ruas e13 = (4,7) dinamakan **ruas berganda** atau **ruas sejajar** (*multiple edges* atau *paralel edges*), karena kedua sisi ini menghubungi dua buah simpul yang sama, yaitu simpul 4 dan simpul 7.

Pada Graph *G* (gbr.2), ruas e3 = (1,1) dinamakan **gelung** atau **self-loop** karena ia berawal dan berakhir pada simpul yang sama.

**Derajat (*Degree*)** suatu simpul *d*(*v*) adalah banyaknya ruas yang menghubungi simpul tersebut.

Sedangkan ***Derajat* Graph** G adalah jumlah derajat semua simpul pada Graph G.

Pada Gbr 2, derajat simpul

*d*(1) = 5

*d*(2) = 5

*d*(3) = 3

*d*(4) = 5

*d*(5) = 2

*d*(6) = 3

*d*(7) = 5

*d*(8) = 3

*d*(9) = 0

*d*(10) = 1 +

= 30

Maka **derajat Graph** G adalah 32 dan jumlah derajat semua simpul Graph (derajat Graph) = dua kali banyaknya ruas Graph (size/ukuran Graph).

Simpul 9 dikatakan **simpul terpencil / simpul terisolasi,** karena *d*(9) = 0 dan simpul 10 dikatakan **simpul bergantung / simpul akhir** karena *d*(10) = 1

* 1. **Jenis Graph**

Berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graph, maka graph digolongkan menjadi dua jenis:

1. **Graph sederhana** (*simple graph*).

Graph yang tidak mengandung gelung maupun sisi-ganda dinamakan graph sederhana.

2. **Graph tak-sederhana** (*unsimple-graph/****multigraph*)**.

Graph yang mengandung ruas ganda atau gelung dinamakan graph tak-sederhana (*unsimple graph* atau *multigrapf*).

Berdasarkan jumlah simpul pada suatu graph, maka secara umum graph dapat digolongkan menjadi dua jenis:

1. **Graph berhingga** (*limited graph*)

Graph berhingga adalah graph yang jumlah simpulnya, *n*, berhingga.

2. **Graph tak-berhingga** (*unlimited graph*)

Graph yang jumlah simpulnya, *n*, tidak berhingga banyaknya disebut **graph tak-berhingga**.

Berdasarkan orientasi arah pada sisi, maka secara umum graph dibedakan atas 2 jenis:

1. **Graph tak-berarah** (*undirected graph*)

Graph yang sisinya tidak mempunyai orientasi arah disebut graph tak-berarah.

2. **Graph berarah** (*directed graph* atau *digraph*)

Graph yang setiap sisinya diberikan orientasi arah disebut sebagai graph berarah.

e4

e3  1 e1 2 e2 3 e5 4

e10 e11

e6 e7 e8 e9

5 6

e15

**Gbr 3**  graph berarah

* 1. **Subgraph**

Misalkan G = (V, E) adalah sebuah graph. G1 = (V1, E1), jika V1 ⊆ V dan E1 ⊆ E. maka *G*1 adalah **subgraph** (*subgraph*) dari *G .*

G = (V, E)

Dengan V = { A, B, C, D }

E = { e1, e2, e3, e4, e5 }

Dan

G1 = (V1, E1)

Dengan V1 = { A, B, D } ⊆ V

E1 = { e1, e5 } ⊆ E

D

C

B

A

e5

e1

e3

e25

e4

G

B

AD

D

e1

e5

G1

D

C

B

e3

e25

e4

G2

**Komplemen** dari subgraph *G*1 terhadap graph *G* adalah graph *G*2 = (*V*2, *E*2) sedemikian sehingga *E*2 = *E* - *E*1 dan *V*2 adalah himpunan simpul yang anggota-anggota *E*2 bersisian dengannya.

G = (V, E)

Dengan V = { A, B, C, D }

E = { e1, e2, e3, e4, e5 }

G1 = (V1, E1) subgraph dari G

Dengan V1 = { A, B, D } ⊆ V

E1 = { e1, e5 } ⊆ E

G2 = (V1, E1) subgraph dari G

Dengan V2 = { B, C, D } ⊆ V

E2 = { e2, e3, e4 } ⊆ E

Dan terlihat *E*2 = *E* - *E*1

**Subgraph yang Direntang (*Spanning Subgraph*)** Apabila E‘ mengandung semua ruas di E yang kedua ujungnya di V‘ , maka G‘ adalah **Spanning Subgraph** dari G yang dibentuk oleh V‘ .

G2 = (V1, E1) Spanning subgraph dari G

Dengan V2 = { B, C, D } ⊆ V

E2 = { e2, e3, e4 } ⊆ E

Anggota E2 adalah semua ruas dari G yang menghubungkan simpul di V2.

* 1. **Keterhubungan Graph**

Dua buah simpul dikatakan ***bertetangga* (*Adjacent*)** bila keduanya terhubung langsung oleh sebuah ruas.

D

C

B

A

e5

e1

e3

e25

e4

G

Pada graph *G* disamping,

simpul A bertetangga dengan simpul B dan D,

simpul A tidak bertetangga dengan simpul C

**Bersisian (*Incidency*)**

Untuk sembarang ruas *e* = (*vj*, *vk*) dikatakan :

*e* bersisian dengan simpul *vj* , atau *e* bersisian dengan simpul *vk*

Pada graph *G*

ruas e5 bersisian dengan simpul A dan simpul D, tetapi

ruas e5 tidak bersisian dengan simpul B

***Simpul terpencil* (*Isolated Vertex*)** ialah simpul yang tidak mempunyai sisi yang bersisian dengannya atau simpul bererajat 0 (nol).

##### ***Graf Kosong (*null graf *atau* empty graf***)* adalah graf yang himpunan sisinya merupakan himpunan kosong (*Nn*).

D

A

B

C

Graph N4

* 1. **Operasi Graph**

*G1 = (E1,V1) , G2 = (E2,V2)*

1. Gabungan G1 ∪ G2 adalah graf dgn himpunan ruasnya E1 ∪ E2.

2. Irisan G1 ∩ G2 adalah graf dgn himpunan ruasnya E1 ∩ E2.

3. Selisih G1 - G2 adalah graf dgn himpunan ruasnya E1 - E2.

4. Selisih G2 – G1 adalah graf dgn himpunan ruasnya E2 - E1.

5. Penjumlahan ring G1 ⊕ G2 adalah graf dgn himpunan ruasnya (E1 ∪ E2) - (E1 ∩ E2) atau (E1 - E2) ∪ (E2 - E1).

D

C

B

A

e5

e1

e3

e25

e4

G1

D

C

G2 – G1

e6

e8

e7

D

C

A

e5

e4

G2

e6

e8

e7

D

C

A

e5

e4

G1 G2

D

C

B

A

e1

e3

e25

G1 - G2

D

C

B

A

e5

e1

e3

e25

e4

G1 G2

e6

e8

e7

D

C

B

A

e1

e3

e25

e6

e8

e7

G2

G1

Graph K dan L merupakan **dekomposisi** Suatu graf G, bila K ∪ L = G dan K ∩ L = ∅

Contoh :

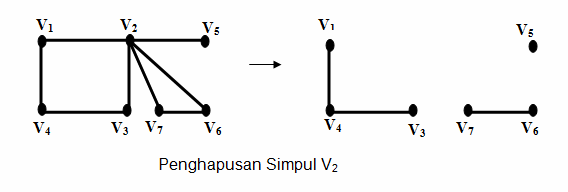


**Penghapusan ( deletion** ) dapat dilakukan pada simpul ataupun ruas.

1) Penghapusan Simpul .

Notasinya : G – {V}

Contoh :



2) Penghapusan Ruas .

Notasinya : G – {e}

**Pemendekan/Shorting** adalah menghapus simpul yang dihubungkan oleh 2 ruas (simpul berderajat 2), lalu menghubungkan titik-titik ujung yang lain dari kedua ruas tersebut.

Contoh :



* 1. **Keterhubungan**

**Perjalanan atau walk** pada suatu Graf G adalah barisan simpul dan ruas berganti-ganti

v1, e1, v2, e2, …,en-1,

dengan vn simpul

 ei ruas menghubungkan vi dan vi+1

dapat hanya ditulis barisan ruas atau barisan simpul saja.

e1, e2, …,en-1 atau v1, v2, …, vn-1, vn

Dalam hal ini, v1 disebut simpul awal, dan vn disebut simpul akhir.

Perjalanan disebut ***perjalanan tertutup***bila v1 = vn, sedangkan Perjalanan disebut ***perjalanan tebuka***yang menghubungkan v1 dan vn. **Panjang Perjalanan** adalah banyaknya ruas dalam barisan tersebut

**Lintasan (*Trail*)** Lintasan adalah Walk dengan *semua ruas* dalam barisan adalah berbeda.

**Jalur** adalah Walk yang semua simpul dalam barisan adalah berbeda.

Lintasan yang berawal dan berakhir pada simpul yang sama disebut **sirkuit** atau **siklus**. **Panjang sirkuit** adalah jumlah ruas dalam sirkuit tersebut.

Graf yang tidak mengandung sirkuit disebut ***acyclic.***

Contoh Pohon :



***Suatu graf G disebut terhubung*** jika untuk setiap simpul dari graf terdapat jalur yang menghubungkan kedua simpul tersebut.

Subgraf terhubung suatu graf disebut ***komponen dari G*** bila subgraf tersebut tidak terkandung dalam subgraf terhubung lain yang lebih besar.

**Contoh :**



**Rank (G) = n – K**

**Nullity (G) = e – (n – k)**

Dimana : n : Order graf G

e : Size graf G

K : banyaknya komponen graf G

**Jarak antara 2 simpul dalam graf G** adalah panjang jalur terpendek antara ke 2 simpul tersebut.

**Diameter suatu graf terhubung G** adalah maksimum jarak antara simpul G.



Jarak maksimum dalam graf G adalah 3 (yaitu antara A – G atau B - G ataupun C - G), jadi diameter = 3

# Himpunan-Potong ( *Cut-set )* dari graf terhubung *G* adalah himpunan ruas yang bila dibuang dari *G,* menyebabkan *G* tidak terhubung. Jadi Himpunan-Potong selalu menghasilkan dua buah komponen.

D

C

B

A

e3

e1

e25

e4

G

E

e5

# Himpunan {e1 , e4} adalah cut-set, juga {e2,e4}, tetapi {e3 , e5} bukan, karena {e5} adalah cut-set

* 1. **Penyajian Graph**

Terdapat 2 cara standard untuk menyatakan graph yaitu :

* **Link list**
* **Matriks**

Matriks-matriks yang dapat menyajikan model graf tersebut antara lain :

1. Matriks Ruas

Matriks ukuran (2 X M) atau (M X 2) yang menyatakan ruas dari Graf. Matriks ini tidak dapat mendeteksi adanya simpul terpencil, kecuali jumlah simpul yang terdapat dalam Graf disebutkan.

1. Matriks Adjacency

Notasi

Matriks adjacency merupakan matriks simetri. Elemen yang tidak bernilai nol pada diagonal utama menyatakan suatu ***loop. Simpul terpencil*** dapat dideteksi bila ada baris yang semua elemennya bernilai nol.

c) Matriks Insidensi

Notasi :

Jumlah elemen tidak nol pada suatu baris menunjukkan derajat dari simpul.

Setiap kolom mempunyai tepat dua elemen yang tidak nol. Suatu kolom yang hanya mempunyai satu elemen tidak nol menunjukkan suatu ***loop***.

1. Matriks Connection

Notasi :

Graf terhubung jika dan hanya jika matriks tidak mengandung elemen nol. Tidak dapat mendeteksi adanya ruas sejajar dan loop.

4

3

2

1

e5

e1

e3

e25

e4

5

e71

e65

Contoh representasi Graph :

Adjacency List

2

4

1

1

4

2

3

2

3

5

4

1

4

2

5

3

3

4

5

Matriks ruas

atau

Matriks Ajasensi

Matriks Insidensi

Matriks koneksi