BAB II

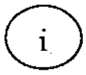
MATERI PENUNJANG

2.1 Dasar – dasar Teori Graph

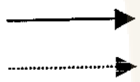
Pada bagian ini akan diberikan beberapa definisi dan teorema tentang graph yang berkaitan dengan metode network simpleks.

2.1.1 Definisi graph, subgraph dan operasi pada graph

Untuk memudahkan pemahaman gambar, terlebih dahulu diberikan makna gambar



: lingkaran bernomor menggambarkan node



: garis berarah menggambarkan arc.

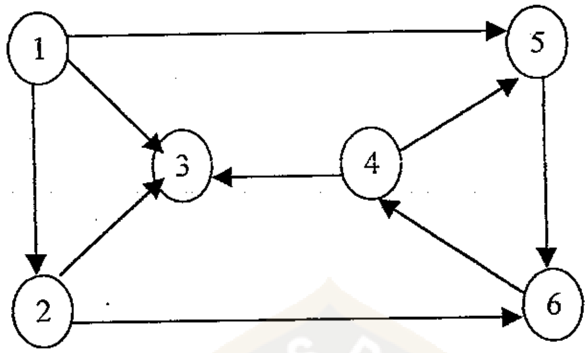
Definisi 2.1

Sebuah graph berarah $G=(N,A)$ terdiri dari himpunan berhingga node $N=\{1,2,3,\cdots,n\}$ dan himpunan arc A yang elemen elemennya merupakan pasangan berurutan dari node yang berbeda yaitu $A=\{(i,j)\}$ dimana i dan j adalah node yang berbeda

Definisi 2.2

Sebuah arc berarah (i,j) adalah sebuah garis berarah yang menghubungkan node awal i dan node akhir j. Jika *arc* $(i,j)\in A$ maka node j dikatakan *adjacent* dengan node i.

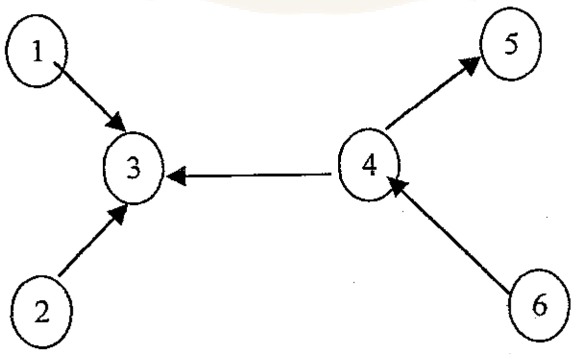
Untuk memperjelas definisi di atas, berikut diberikan gambar graph berarah



Gambar 2.1 Graph berarah dengan 6 node dan 9 arc

Definisi 2.3

Sebuah graph $G'=(N',A')$ merupakan subgraph dari graph $G=(N,A)$ jika $N'\subseteq N$ dan $A'\subseteq A$. Jika $N'=N$ dan $A'\subseteq A$ maka $G'=(N',A')$ dikatakan sebagai spanning subgraph dari $G=(N,A)$



Gambar 2.2 Contoh Spanning subgraph dari gambar 2.1

Definisi 2.4

Misal $G_1 = (N_1, A_1)$ dan $G_2 = (N_2, A_2)$ adalah subgraph dari graph $G = (N, A)$ union dari dua graph $G_1 = (N_1, A_1)$ dan graph $G_2 = (N_2, A_2)$ adalah graph $G_3 = G_1 \cup G_2$ dengan himpunan node $N_3 = N_1 \cup N_2$ dan dan himpunan arc $A_3 = A_1 \cup A_2$. Sebarang subgraph dapat digabungkan.

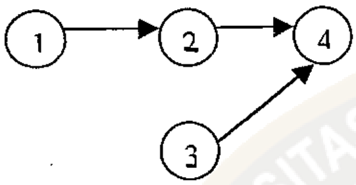
Definisi 2.5

Misal $G_1 = (N_1, A_1)$ dan $G_2 = (N_2, A_2)$ adalah subgraph dari graph $G = (N, A)$. Irisan dari dua graph $G_1 = (N_1, A_1)$ dan graph $G_2 = (N_2, A_2)$ adalah suatu graph lain $G_4 = G_1 \cap G_2$ dengan himpunan node $N_4 = N_1 \cap N_2$ dan himpunan arc $A_4 = A_1 \cap A_2$. Dua buah subgraph dapat diriskan jika mempunyai node yang berserikat.

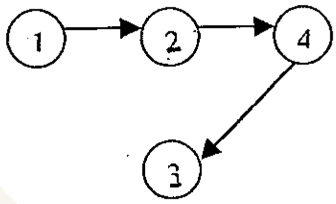
2.1.2 Path, Cycle dan Tree

Definisi 2.6

Path yang menghubungkan node i_0 dan node i_p adalah suatu barisan arc, yaitu $P = \{(i_0, i_1), (i_1, i_2), \cdots, (i_{p-1}, i_p)\}$ dimana node awal setiap arc sama dengan node akhir arc sebelumnya dalam barisan tersebut dan $i_0, i_1, i_2, \cdots, i_p$ merupakan node yang berbeda . Arc pada path P dapat dibedakan menjadi dua yaitu arc maju *(forward arc)* dan arc mundur (*backward arc*). Sebuah *arc* (i, j) pada path P dikatakan arc maju jika arc tersebut adalah arc yang keluar dari node i dan masuk ke node j. Dan dikatakan arc mundur jika berlawan arahnya dengan *arc* (i,j).Sebuah path berarah adalah sebuah path dimana semua arc dalam path mempunyai arah yang sama.



Gambar 2..3 Contoh Path

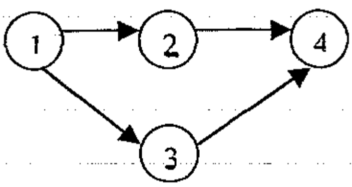


Gambar 2.4 Contoh Path Berarah

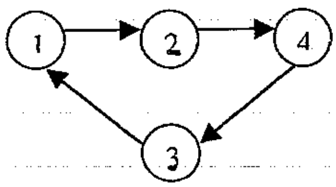
Definisi 2.7

Sebuah cycle adalah sebuah path dengan sebuah arc tambahan yang menghubungkan node akhir dan node awal. Cycle berarah adalah sebuah path berarah dengan node awal sama dengan node akhir $i_0 = i_p$. Arc maju pada cycle yaitu arc yang searah dengan orientasi yang ditentukan dan arc mundur adalah arc yang arahnya berlawanan dengan orientasi cycle.

Pada gambar 2.5 arc maju terdiri dari $\bar{C} = \{(1,2),(2,4)\}$ dan arc mundur $\underline{C} = \{(3,4),(1,3)\}$



Gambar 2.5 Contoh Cycle



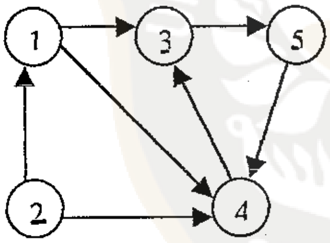
Gambar 2.6 Contoh Cycle Berarah

Definisi 2.8

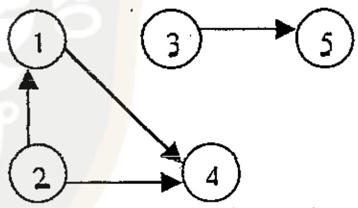
Sebuah graph dikatakan tidak siklik jika tidak memuat cycle.

Definisi 2.9

Dua node i dan j dikatakan terhubung jika graphnya paling sedikit memuat sebuah path yang menghubungkan node i dan j. Sebuah graph dikatakan terhubung jika setiap pasangan node terhubung dan dikatakan tak terhubung jika ada node yang tidak terhubung. Sebuah subgraph terhubung maksimal dari network yang tak terhubung disebut *komponen.*



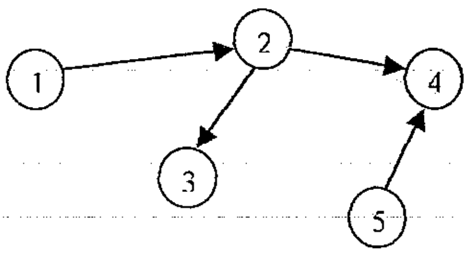
Gambar 2.7 Contoh graph terhubung



Gambar 2.8 Contoh graph tak terhubung

Definisi 2.10

Tree adalah sebuah graph terhubung yang tidak memuat cycle.



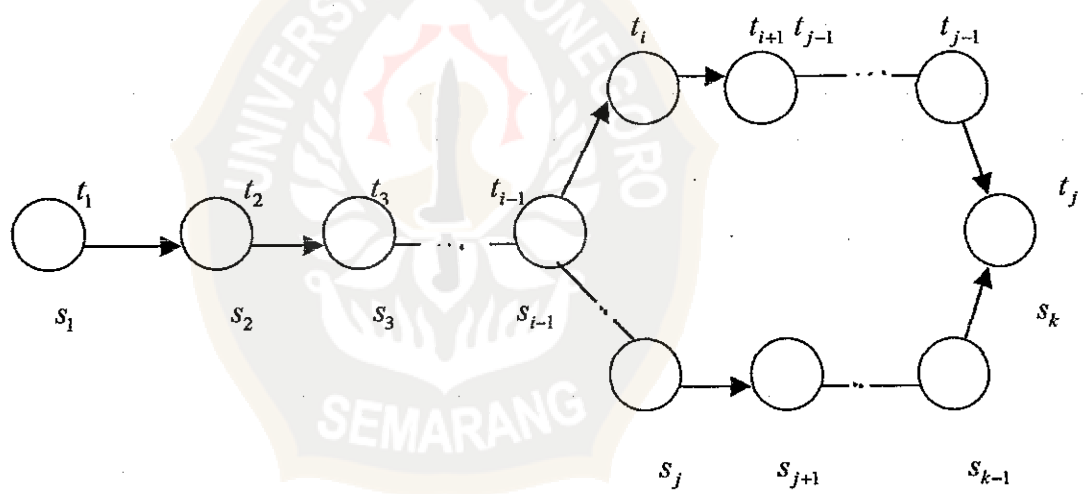
Gambar 2.9 Contoh tree

Teorema 2.1

Jika G adalah graph terhubung dan tidak mempunyai cycle maka G merupakan tree

Bukti:

Misal G adalah graph terhubung yang tidak mempunyai cycle dan andaikan G bukan tree. Sebagai ilustrasi diberikan pada gambar 2.10.



Gambar 2.10 Pembuktian tree

Selanjutnya akan ada dua path berbeda yang menghubungkan dua node yang berbeda dari G, misalkan node p dan node q. Kedua path tersebut yaitu $P = \{(t_1,t_2),(t_2,t_3),\cdots,(t_{j-1},t_j = q)\}$ dan $P = \{(s_1,s_2),(s_2,s_3),\cdots,(s_{k-1},s_k = q)\}$.

Karena kedua path itu berbeda maka harus ada suatu indeks i sehingga dipenuhi $t_1 = s_1, t_2 = s_2, \cdots, t_{i-1} = s_{i-1}$ dan $t_i \neq s_i$. Selain itu juga ada indeks lain $j > i$

sehingga $t_j = s_k$ untuk setiap nilai k tertentu. Tetapi hal ini akan mengakibatkan graph tersebut mempunyai cycle

$$C = \{(t_{i-1}, t_i), (t_i, t_{i+1}), \cdots, (t_{j-1}, t_j = s_k), (s_k, s_{k-1}), \cdots, (s_j, s_{j-1} = t_{i-1})\}.$$

Tetapi hal ini bertentangan dengan asumsi bahwa graph G tidak mempunyai cycle. Jadi pengandaian salah dan haruslah G merupakan tree.

Terbukti.

Teorema 2.2

Suatu tree dengan n node mempunyai $n-1$ arc.

Bukti:

Untuk membuktikan akan digunakan induksi matematik. Untuk $n=1$ jelas teorema berlaku. Selanjutnya misalkan G adalah tree dengan $n>1$ node. Karena path dalam tree tunggal maka penghapusan sebuah arc dari graph G akan membuat G menjadi dua komponen , misal G_1 dan G_2 dengan banyaknya node masing – masing n_1 dan n_2 dimana $n_1 + n_2 = n$. Dari hipotesis G_1 mempunyai $n_1 - 1$ arc dan G_2 mempunyai $n_2 - 1$ arc. Oleh karena itu banyaknya arc dari graph G adalah $(n_1 - 1) + (n_2 - 1) + 1 = n - 1$.

Terbukti.

Definisi 2.11

Derajat masuk *(in-degree)* dari sebuah node i yaitu banyaknya arc yang masuk pada node i dan derajat keluar (*out degree*) dari sebuah node i yaitu banyaknya arc yang keluar dari node i tersebut . Degree suatu node yaitu jumlah dari indegree dan out degree.

Dari gambar 2.1 dapat ditentukan degree setiap titiknya sebagai berikut:

misal indegree ditulis dengan $d^-(i)$, out degree dengan $d^+(i)$ dan degree dengan $d(i)$ sehingga

$$d^-(1)=0, d^-(2)=1, d^-(3)=3, d^-(4)=1, d^-(5)=2, d^-(6)=2$$

$$d^+(1)=3, d^+(2)=2, d^+(3)=0, d^+(4)=2, d^+(5)=1, d^+(6)=1$$

$$d(1)=3, d(2)=3, d(3)=3, d(4)=3, d(5)=3, d(6)=3$$

Teorema 2.3

Jumlah degree dari semua node dalam graph adalah dua kali jumlah arc dalam graph.

Bukti:

Karena setiap arc insiden pada dua node sehingga menyumbangkan 2 pada jumlah degree .

Terbukti

Teorema 2.4

Sebuah tree dengan banyaknya node $n \geq 2$ paling sedikit mempunya dua node ujung.

Bukti :

Misal $T = (N, A)$ adalah tree dengan $n \geq 2$ node. Setiap node T insiden pada paling sedikit satu arc, sebab jika tidak demikian T akan tidak terhubung. Sehingga degree dari setiap node paling sedikit satu. Dari teorema 2.2 tree T mempunyai $n-1$ arc. Andaikan T mempunyai node ujung lebih kecil dari 2.

Kasus 1:

Misal T tidak mempunyai node ujung maka untuk setiap node $i \in N, d(i) \geq 2$ sehingga $\sum_{i \in N} d(i) \geq 2n \neq 2(n-1)$ dan kontradiksi .

Kasus 2:

Misal T mempunyai satu node ujung misal j sehingga $d(j) = 1$ dan untuk setiap $i \in N - \{j\}$, $d(i) \geq 2$ sehingga berlaku

$$\sum_{i \in N} d(i) = d(j) + \sum_{i \in N - \{j\}} d(i) = 1 + 2(n-1) \neq 2(n-1)$$

Kontradiksi. Jadi i mempunyai paling sedikit dua node ujung.

Terbukti.

Teorema 2.5

Misal G adalah sebuah graph dengan n node dan mempunyai $n-1$ arc. Jika G G tidak mempunyai cycle maka G merupakan tree.

Bukti:

Diketahui G tidak mempunyai cycle maka tinggal dibuktikan bahwa G terhubung. Andaikan G tidak terhubung, maka paling sedikit mempunyai $k \geq 2$ komponen terhubung. Misal $G_1, G_2, \cdots, G_k$ adalah komponen terhubung dari G dengan masing masing komponen mempunyai $n_1, n_2, \cdots, n_k$ node dengan $n_1 + n_2 + \cdots + n_k = n$. Setiap komponen tidak mempunyai cycle, jadi masing – masing komponen merupakan tree. Jumlah arc pada tiap komponen adalah $n_i - 1, i = 1,2,...,k$. Dengan demikian jumlah arc dari graph G adalah: $(n_1 - 1) + (n_2 - 1) + \cdots + (n_k - 1) = n - k \leq n - 2 \neq n - 1$.

Jadi terjadi kontradiksi, sehingga pengandaian G tidak terhubung salah. Oleh karena G terhubung dan tidak mempunyai cycle maka G adalah tree.

Terbukti.

Definisi 2.12

Sebuah subgraph $T=(N',A')$ dari graph $G=(N,A)$ dikatakan spanning tree untuk G jika $T=(N',A')$ adalah spanning subgraph yang merupakan tree.

Teorema 2.6

Jika G suatu graph terhubung maka G akan mempunyai spaning tree. Spanning tree dari graph tersebut dapat diperoleh dengan melakukan penghapusan arc dari graph G sedemikian sehingga G tetap terhubung dan tidak mempunyai cycle.

Bukti:

Jika G tidak mempunyai cycle dan G terhubung maka hal ini akan mengakibatkan bahwa G adalah suatu tree. Selanjutnya misalkan bahwa G mempunyai paling sedikit satu cycle. Dengan menghapus salah satu arc pada cycle akan diperoleh graph baru G_1 dengan banyaknya node sama dengan node G dan tetap terhubung. Bila G masih mempunyai suatu cycle maka salah satu arc yang membentuk cycle dihapus dan akan terbentuk graph baru G_2 yang mempunyai node sama dengan node G_1 dan tetap terhubung. Jika proses ini dilakukan sampai tidak ada cycle pada graph G dan karena graph yang terbentuk masih tetap terhubung maka graph yang diperoleh terakhir merupakan tree.

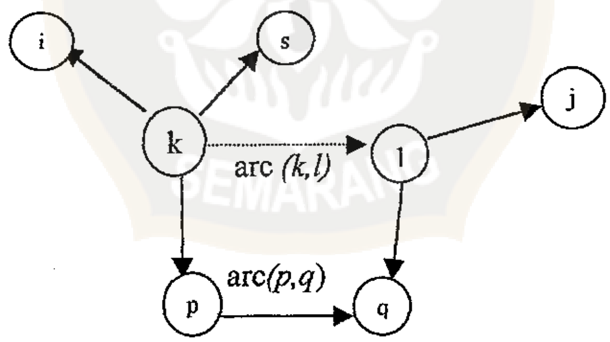
Terbukti.

Teorema 2.7

Misal T adalah spanning tree dari graph G dan arc (k,l) merupakan arc dari G yang bukan arc dari T dan arc (k,l) menghubungkan dua node k dan l pada G (dan juga pada T), maka hanya akan dapat dibuat sebuah cycle C pada T sedemikian sehingga semua arc pada C tersebut adalah arc dari T, kecuali arc (k,l). Bila arc (p,q) adalah arc yang lain dalam C yang tidak sama dengan arc (k,l) dan arc (p,q) dihapus kemudian arc (k,l) ditambahkan pada T maka akan diperoleh suatu tree baru T' yang juga merupakan spanning tree dari G

Bukti:

Sebagai ilustrasi diberikan pada gambar 2.11



Gambar 2.11 Pembentukan spanning tree

Dalam setiap tree T hanya dapat dibuat satu path yang menghubungkan dua node k dan l pada T. Path ini jika ditambah dengan arc (k,l) akan membentuk sebuah cycle C. Jadi cycle C adalah cycle yang memuat arc (k,l) di luar T dan arc yang lain yang merupakan arc dari T. Misalkan arc (p,q) adalah arc sebarang pada C yang tidak sama dengan arc(k,l). Bila arc(p,q) dihapus dari C dan arc(k,l) ditambahkan pada T akan diperoleh graph T'. Tetapi dengan menghapus arc (p,q) dari T maka cycle C akan terputus dan graph T' yang terbentuk merupakan graph yang tidak mempunyai cycle. Harus ditunjukkan bahwa T' adalah tree .

Karena ke dalam spanning tree T ditambahkan sebuah arc dan kemudian sebuah arc dihapus dari spanning tree maka pada T' mempunyai arc yang sama banyaknya dengan T . Demikian juga banyaknya node dalam T' akan sama dengan banyaknya node dalam T. Oleh karena itu T' merupakan spanning tree.

Terbukti.

2.2 Formulasi Masalah Aliran Biaya Minimal dalam Network

Definisi 2.13

Sebuah network merupakan sebuah graph berarah $G=(N,A)$ dimana setiap node atau arcnya dihubungkan dengan sebuah nilai numerik

Definisi 2.14

Aliran dalam network merupakan suatu bobot nonnegatif yang dihubungkan dengan arc dalam network $G=(N,A)$. Nilai aliran merupakan suatu fungsi dari himpunan arc A ke suatu himpunan bilangan riil nonnegatif.

Masalah utama dalam masalah aliran biaya minimal yaitu bagaimana menentukan pemindahan atau pendistribusian suatu barang atau komoditas dalam suatu network dengan biaya seminimal mungkin untuk memenuhi permintaan konsumen (*demand*) dari node tertentu yaitu node suplai.

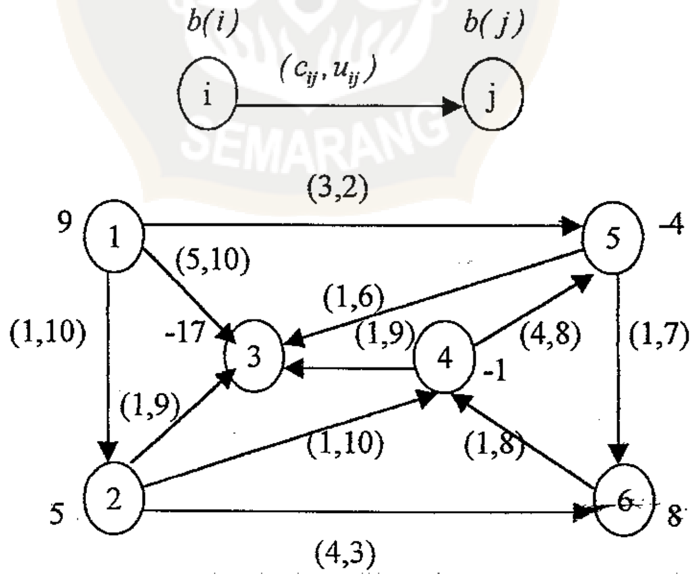
Misal $G = (N, A)$ suatu network berarah yang didefinisikan oleh sebuah himpunan n node N dan himpunan m arc A. Setiap arc$(i, j) \in A$ mempunyai biaya (*cost*) c_{ij} yang menggambarkan cost per unit aliran pada arc tersebut. Diasumsikan cost aliran bervariasi secara linier dengan jumlah aliran. Untuk setiap$(i, j) \in A$ dihubungkan dengan sebuah kapasitas u_{ij} yang menggambarkan jumlah maksimum aliran pada arc tersebut serta batas bawah l_{ij} yang menggambarkan jumlah minimum aliran pada arc$(i, j) \in A$. Setiap node $i \in N$ dihubungkan dengan sebuah bilangan bulat $b(i)$ yang menunjukkan suatu persediaan (*suplai*) atau permintaan (*demand*). Jika nilai $b(i) > 0$ maka node $i \in N$ merupakan node suplai, jika $b(i) < 0$ maka node $i \in N$ merupakan node demand dan jika $b(i) = 0$ maka node $i \in N$ merupakan node perantara (*transhippment node*). Secara umum masalah aliran biaya minimal diformulasikan sebagai berikut :

$$Min\ z(x) = \sum_{(i,j)\in A} c_{ij} x_{ij} \qquad (2.1)$$

$$kendala \quad \sum_{:(i,j)\in A} x_{ij} - \sum_{:(j,i)\in A} x_{ji} = b(i) \text{ untuk semua } i \in N \qquad (2.2)$$

$$l_{ij} \le x_{ij} \le u_{ij} \text{ untuk semua } (i, j) \in A \qquad (2.3)$$

Kendala (2.2) dinamakan konservasi aliran (*flow conservation*) atau kendala keseimbangan nodal (*nodal balance constrain*). Suku pertama pada (2.2) menyatakan total aliran yang keluar dari node i dan suku kedua menyatakan total aliran yang masuk (*inflow*) ke node j. Konservasi aliran menyatakan bahwa aliran yang keluar (*outflow*) dari satu node dikurangi aliran yang masuk harus sama dengan suplai atau demand dari node tersebut. Jika node i adalah node suplai maka aliran yang keluar melebihi aliran yang masuk dan jika i node demand maka aliran yang masuk melebihi aliran yang keluar . Jika node i adalah node perantara maka aliran yang masuk sama dengan aliran yang keluar . Aliran pada setiap arc juga harus memenuhi batas bawah aliran dan batas atas atau kapasitas aliran..Untuk memberi gambaran tentang network dapat dilihat pada gambar 2.12



Gambar 2.12 Contoh aliran dalam network