BAB II

MATERI PENUNJANG

Dalam pembahasan algoritma simpleks pada Bab III nanti, diperlukan beberapa materi penunjang agar pembahasannya lebih terarah. Materi penunjang yang dibahas pada bab ini adalah mengenai himpunan dan operasi himpunan, dasar-dasar teori graph, program linier serta metode simpleks dengan variabel terbatas.

2.1 Himpunan dan Operasi Himpunan

Definisi 2.1

Himpunan adalah kumpulan objek – objek yang di definisikan secara jelas dan mempunyai hubungan yang erat. Objek – obyek yang berada dalam himpunan disebut elemen atau anggota himpunan. Notasi untuk himpunan adalah dengan huruf besar A, B, C dan seterusnya atau jika anggota- anggotanya didaftar satu persatu digunakan notasi $A = \{a,b,c,...\}$

Misal a adalah anggota himpunan A, selanjutnya akan dituliskan dengan $a \in A$ dan jika a bukan anggota A ditulis dengan $a \notin A$

Definisi 2.2

Himpunan kosong, dinotasikan dengan ϕ adalah himpunan yang tidak mempunyai anggota.

Definisi 2.3

Sebuah himpunan A dikatakan subset dari himpunan B jika untuk setiap anggota himpunan A menjadi anggota himpunan B dan dinotasikan dengan $A \subseteq B$.

Definisi 2.4

Sebuah himpunan A dikatakan subset sejati dari himpunan B jika untuk setiap anggota himpunan A menjadi anggota himpunan B dan $A \neq B$, dinotasikan dengan $A \subset B$. Himpunan kosong merupakan himpunan bagian dari sebarang himpunan tetapi bukan himpunan bagian sejati.

Definisi 2.5

Dua buah himpunan A dan B dikatakan sama jika mempunyai anggota yang sama yaitu jika untuk sebarang $a \in A$ maka $a \in B$ dan juga jika untuk sebarang $a \in B$ maka juga berlaku $a \in A$, dinotasikan dengan $A = B$

Dengan kata laian $A=B$ jika dan hanya jika :

1. $A \subseteq B$ dan
2. $B \subseteq A$

Definisi 2.6

Misal A dan B adalah suatu himpunan. Gabungan atau union dari A dan B, dinotasikan dengan $A \cup B$ adalah suatu himpunan yang anggota – anggotanya

menjadi anggota himpunan A atau himpunan B . Dengan bentuk notasi matematik sebagai berikut :

$$A \cup B = \{x \mid x \in A \text{ atau } x \in B \}$$

Definisi 2.7

Misal A dan B adalah suatu himpunan. *Irisan* dari A dan B, dinotasikan dengan $A \cap B$ adalah suatu himpunan yang anggota – anggotanya menjadi anggota himpunan A dan sekaligus menjadi anggota himpunan B. Dengan bentuk notasi matematik sebagai berikut:

$$A \cap B = \{x \mid x \in A \text{ dan } x \in B \}$$

Definisi 2.8

Misal A dan B adalah suatu himpunan. Selisih *(deference*) dari A dan B, dinotasikan dengan $A - B$, adalah suatu himpunan yang anggota – anggotanya menjadi anggota himpunan A tetapi tidak menjadi anggota himpunan B . Dengan bentuk notsi matematik sebagai berikut :

$$A - B = \{x \mid x \in A \text{ dan } x \notin B \}$$

2.2 Dasar – dasar Teori Graph

Dalam hal ini, teori graph digunakan untuk menggambarkan bentuk dari jaringan *(network)* yang akan diselesaikan dengan algoritma simpleks.

2.2.1 Definisi graph, subgraph dan operasi dalam graph

Untuk memudahkan pemahaman gambar, terlebih dahulu diberikan makna gambar

i : lingkaran bernomor menggambarkan node

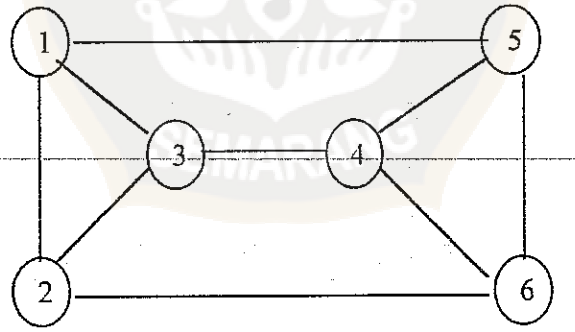
: garis menggambarkan arc.

Definisi 2.9

Sebuah graph tak berarah $G=(N,A)$ terdiri dari himpunan tidak kosong yang berhingga dari node $N=\{1,2,3,\cdots,n\}$ dan himpunan arc A yang elemen elemennya merupakan pasangan dari node yang berbeda.

$A=\{(1,2),(2,3),\cdots,(i,j),\cdots,(n-1,n)\}$

Untuk memperjelas definisi di atas, berikut diberikan sebuah gambar graph:



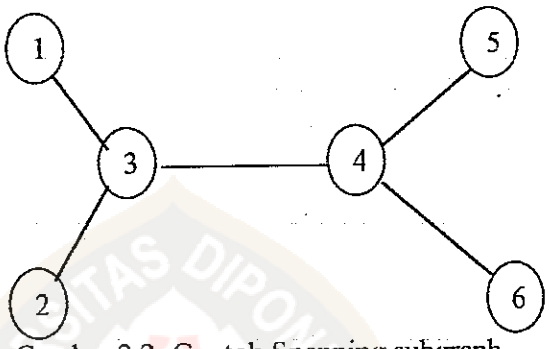
Gambar 2.1 Graph dengan 6 node dan 9 arc

Definisi 2.10

Dalam graph $G=(N,A)$, untuk *arc* $e_{ij}=(i,j)\in A$ dan $(i,j)\in N$ maka node i dan node j dikatakan *adjacent* dalam G, dan dimana *arc* e_{ij} menghubungkan node i dan j.

Definisi 2.11

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 $G=(N,A)$

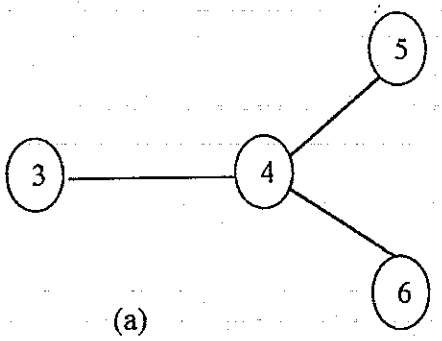


Gambar 2.2 Contoh Spanning subgraph dari gambar 2.1

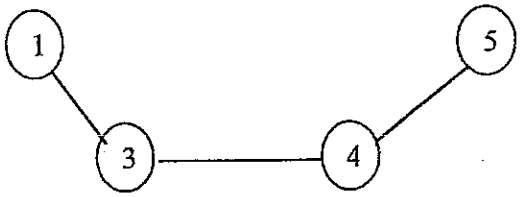
Definisi 2.12

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 $A_3=A_1^{'} \cup A_2^{'}$.

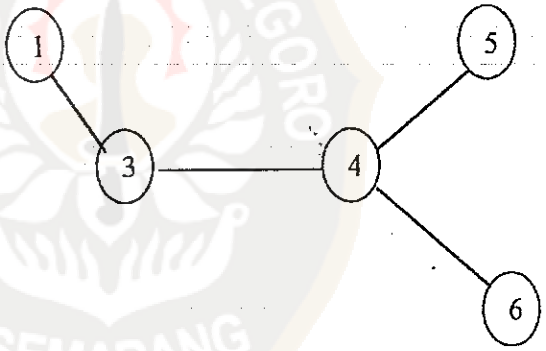
Sembarang subgraph dapat digabungkan. Sebagai contoh akan diberikan gambar berikut ini:



(a)



(b)



(c)

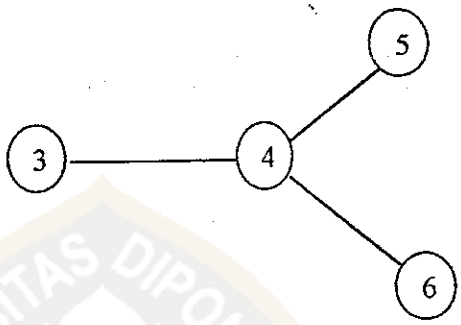
Gambar 2.3 (a) $G_1^{'} = (N_1^{'}, A_1^{'})$ dan (b) $G_2^{'} = (N_2^{'}, A_2^{'})$ subgraph dari graph pada gambar 2.1
(c) $G_3 = G_1^{'} \cup G_2^{'}$

Definisi 2.13

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 $G_2^{'} = (N_2^{'}, A_2^{'})$

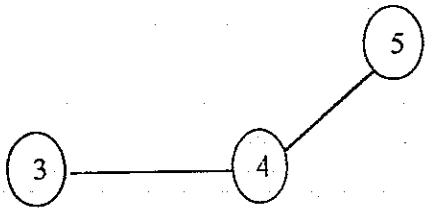
adalah suatu graph lain $G_4 = G_1' \cap G_2'$ dengan himpunan node $N_4 = N_1' \cap N_2'$ dan himpunan arc $A_4 = A_1' \cap A_2'$.

Dua buah subgraph dapat diriskan jika mempunyai node yang berserikat.



(a)

(b)



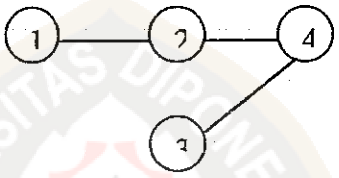
(c)

Gambar 2.4 (a) $G_1' = (N_1', A_1')$ dan (b) $G_2' = (N_2', A_2')$ subgraph dari graph pada gambar 2.1

2.2.2 Path, Cycle dan Tree

Definisi 2.14

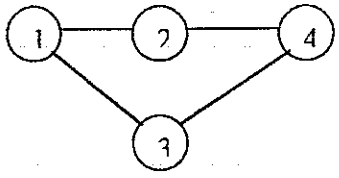
Path yang menghubungkan node i_0 dan node i_p adalah suatu barisan node dan arc, berurutan yaitu $P = \{i_0, (i_0, i_1), i_1, (i_1, i_2), \cdots, i_{p-1}, (i_{p-1}, i_p), i_p\}$ dimana $i_0, i_1, i_2, \cdots, i_p$ merupakan node yang berbeda .



Gambar 2.5 Contoh Path

Definisi 2.15

Sebuah cycle adalah sebuah path dengan sebuah arc tambahan yang menghubungkan node akhir dengan node awal.



Gambar 2.6 Contoh Cycle

Definisi 2.16

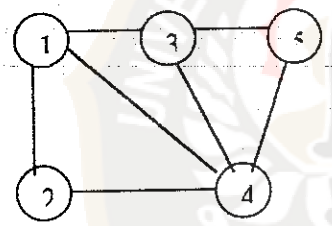
Sebuah graph dikatakan siklik jika memuat cycle.

Definisi 2.17

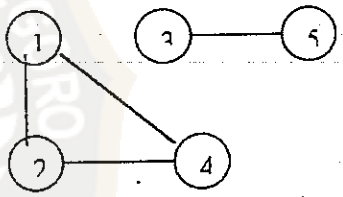
Sebuah graph dikatakan tidak siklik jika tidak memuat cycle.

Definisi 2.18

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. Masing-masing subgraph terhubung maksimal dari graph yang tak terhubung disebut *komponen*.



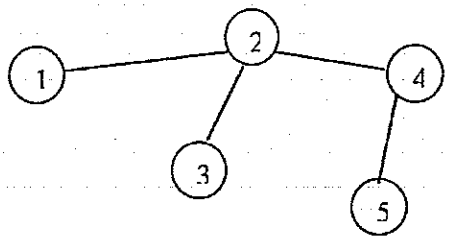
Gambar 2.7 Contoh graph terhubung



Gambar 2.8 Contoh graph tak terhubung

Definisi 2.19

Tree adalah sebuah graph terhubung yang tidak memuat cycle.



Gambar 2.7 Contoh tree

Teorema 2.1

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

Bukti:

Untuk membuktikan akan digunakan induksi matematik.

1. Akan dibuktikan benar untuk $n = 1$

Jika n = 1 maka haruslah arc = 0

Untuk n = 1 maka arc = n –1

= 1 – 1

= 0

2. Asumsi benar untuk n = k

Akan ditunjukkan benar untuk n = k + 1 sehingga jumlah arc = (k + 1) - 1

Untuk n = k maka jumlah arc = k – 1..... (a)

Dari (a) jika n = k + 1 maka jumlah arc = (k – 1) + 1

= (k + 1) – 1

Jadi terbukti bahwa jika suatu tree dengan n node mempunyai n – 1 arc. ■

Definisi 2.20

Derajat (*degree)* dari sebuah node i adalah banyaknya arc yang insiden pada node tersebut

Dari gambar 2.1 dapat ditentukan degree setiap titiknya sebagai berikut:

Misal degree suatu node ditulis dengan $d(i)$ sehingga

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

Teorema 2.2

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 .

Sehingga terbukti bahwa jumlah derajat dari semua node adalah dua kali jumlah arc dalam graph. ■

Definisi 2.21

Node ujung adalah node yang mempunyai derajat satu atau dengan kata lain node yang terhubung hanya dengan satu arc.

Teorema 2.3

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 $i \in N, d(i) \geq 2$

(1). $\sum_{i \in N} d(i) \geq 2n$.

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

dari (1) dan (2) terjadi kontradiksi karena persamaan $2(n-1) \geq 2n$ adalah salah.

Kasus 2:

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

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

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

$= 2n - 3$

dari (1) dan (2) terjadi kontradiksi karena $2n-2 \neq 2n-3$

Dari kasus 1 dan 2 sebuah tree dengan n ≥ 2 mempunyai paling sedikit dua node ujung. ■

Teorema 2.4

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

Bukti:

Karena diketahui G tidak mempunyai cycle maka tinggal dibuktikan bahwa G terhubung. Andaikan G tidak terhubung, maka paling sedikit mempunyai $k \geq 2$ komponen terhubung.

(1) 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 $n_1 + n_2 + \cdots + n_k = n$.

(2) 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: $(n_1 - 1) + (n_2 - 1) + \cdots + (n_k - 1) = n - k$ dengan $k \geq 2$

Dari (1) jumlah arc = n – 1 dan dari (2) maksimal arc yang mungkin adalah n – 2 arc.

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

Definisi 2.22

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 2.5

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. ■

Teorema 2.6

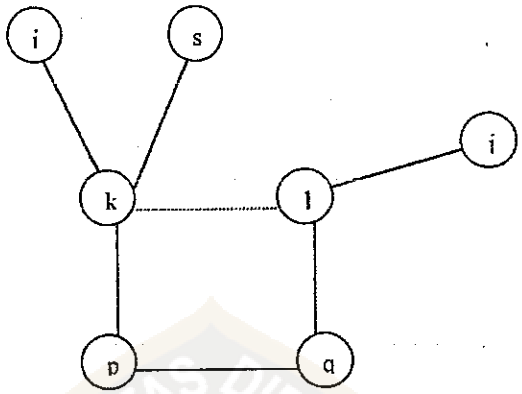
Misal T adalah spanning tree dari graph G dan arc (k,l) merupakan arc dari G yang bukan arc dari T dan arc (k,l) menghubungkan dua node k dan l pada G (dan juga pada T), maka hanya akan dapat dibuat sebuah cycle C pada T sedemikian sehingga semua arc pada C tersebut adalah arc dari T , kecuali arc (k,l). 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:

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 yang berarti bahwa untuk setiap pasangan node i, j dalam T' harus dapat dihubungkan oleh suatu path yang seluruhnya dalam T' atau harus ditunjukkan bahwa T' terhubung.

Misal i dan j adalah dua node sebarang dalam G dan P adalah path yang menghubungkan i dan j. Bila arc (p,q) bukan salah satu arc pada P maka P juga merupakan path pada T' sehingga i dan j terhubung pada T'. Bila arc (p,q) merupakan salah satu arc pada P maka path P dapat diubah dengan menggunakan arc dari C juga merupakan path untuk membentuk path baru P' yang menghubungkan i dan j sehingga arc (p,q) bukan arc pada P'. Jadi P' adalah path pada T' yang menghubungkan i dan j. Jadi jelaslah bahwa T' adalah suatu tree. Karena jumlah node T' sama dengan banyaknya node G maka T' merupakan spanning tree dari graph G. ■



Gambar 2.10 Pembentukan spanning tree

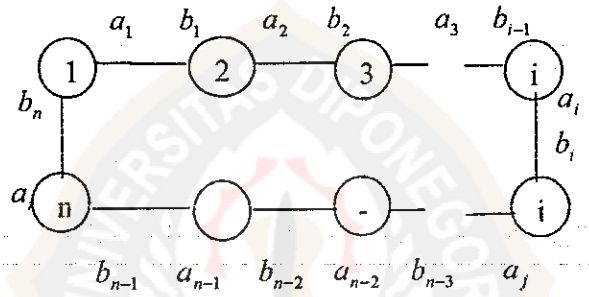
2.2.3 Matriks graph

Untuk memudahkan dalam perhitungan maka suatu graph $G=(N,A)$ dapat dituliskan dalam bentuk matriks. Salah satu notasi graph dalam matriks yaitu *matriks insidensi node - arc*. Matriks insidensi merupakan matriks pembatas atau kendala pada permasalahan yang akan dibahas. Representasi matriks insidensi menyimpan graph dalam matriks $\mathbf{A}_{mxn}$ yang memuat sebuah baris untuk setiap node dan sebuah kolom untuk setiap arc. Kolom yang bersesuaian dengan arc $e_j=(i,j)$ hanya mempunyai dua elemen tak nol karena setiap arc hanya insiden pada dua node.

Sebagai contoh matriks insidensi dari graph pada gambar 2.1 yaitu:

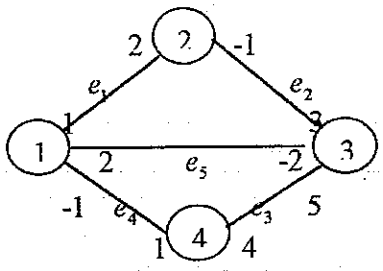
$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix}$$

Pada pembahasan masalah program linier akan digunakan representasi graph dengan pengali arc (*arc multiplier*) seperti gambar :



Gambar2.11 Graph dengan pengali arc

Sebagai contoh graph dengan pengali arc:



Gambar2.12 graph dengan pengali

Matriks insidensi node arcnya yaitu:

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

2.3 Program Linier

Program linier merupakan masalah optimasi dimana fungsi tujuan maupun kendala adalah fungsi linier. Model permasalahan program linier dalam bentuk standard adalah

$$\text{Min } z(x) = \sum_{j=1}^{p} c_j x_j \qquad (2.3.1)$$

$$\text{kendala } \sum a_{ij} x_j = d_i, i = 1,2,\cdots,m \qquad (2.3.2)$$

$$x_j \geq 0, j = 1,2,\cdots,m \qquad (2.3.3)$$

Metode yang paling banyak digunakan untuk menyelesaikan masalah diatas yaitu metode simpleks

Pada metode simpleks, masalah program linier yang diselesaikan harus dalam bentuk standard yaitu fungsi tujuan dalam meminimalkan, kendala persamaan dan ruas kanan nonnegatif. Suatu permasalahan yang tidak dalam bentuk standard, misal maksimalkan $z(x)$ equivalen dengan meminimalkan $-z(x)$. Kendala dalam bentuk pertidaksamaan dapat diubah dalam bentuk persamaan dengan menambahkan variabel slack nonnegatif s_i, dengan cost pada fungsi tujuan nol dan kendala ditulis

$$\sum a_{ij}x_j + s_i' = d_i, i = 1,2,\cdots,m. \qquad (2.3.4)$$

Untuk mendapatkan solusi basis fisibel dapat dilakukan sebagai berikut:

Lakukan isolasi sebuah variabel pada tiap kendala. Misal variabel x_i diisolasi pada kendala ke-i,misal B adalah himpunan indeks dari variabel basis dan misal L adalah himpunan indeks dari variabel non basis, untuk selanjutnya pasangan (B,L) dinamakan struktur basis dari program linier. Untuk struktur basis (B,L) yang diberikan dapat diperoleh partisi yang digabungkan dari kolom matriks pembatas yaitu **A.** Matriks $\mathbf{B}_{pxq}$ dinamakan matriks basis dan matrik $\mathbf{L}_{pxq}$ dinamakan matriks non basis. Misal $X^B = (x_i : i \in B)$ dan $X^L = (x_j : j \in L)$ adalah partisi variabel kendala subvektor yang bersesuaian dengan **B** dan **L** sehingga kendala $\mathbf{A}X = d$ dapat dituliskan

$$\mathbf{B}X^B + \mathbf{L}X^L = d \qquad (2.3.5)$$

Dengan mengalikan (2.3.5) dengan $\mathbf{B}^{-1}$ diperoleh

$$X^B + \mathbf{B}^{-1}\mathbf{L}X^L = \mathbf{B}^{-1}d \qquad (2.3.6)$$

Dari (2.3.6) dengan mengambil $X^L = 0$ diperoleh

$$X^B = \mathbf{B}^{-1}d \qquad (2.3.7)$$

Definisi 2.23

Sebuah solusi X dinamakan solusi fisibel basis jika nilai setiap variabel basis adalah nonnegatif yaitu $X^B \geq 0$. Struktur basis (B,L) dikatakan fisibel jika solusi basis yang bersesuaian adalah fisibel.

Untuk mengkonversikan program linier ke dalam bentuk kanonik mengendaki invers matriks basis **B** yang mana hanya biasa dilakukan jika kolom yang berkaitan dengan variabel basis bebas linier. Jadi basis merupakan subset dengan p variabel yang bersesuaian dengan kolom yang bebas linier.

Dalam proses iterasi metode simpleks diperlukan informasi tentang $\mathbf{B}^{-1}\mathbf{A}_j$ kolom yang diperbaharui bersesuaian dengan variabel nonbasis x_j. Dengan mengambil $\bar{\mathbf{A}}_j = \mathbf{B}^{-1}\mathbf{A}_j$. Selanjutnya vektor kolom ini dinamakan representasi $\mathbf{A}_j$ terhadap matriks basis B karena dengan mengalikan B pada kedua ruas persamaan diperoleh $\mathbf{B}\bar{\mathbf{A}}_j = \mathbf{A}_j$ yang mana dapat diinterpretasikan dalam $\mathbf{A}_j$ sebagai bobot yang digunakan untuk mengalikan kolom matriks basis **B** untuk memperoleh kolom $\mathbf{A}_j$. Misal $\bar{\mathbf{A}}^L = \mathbf{B}^{-1}L$ adalah matriks kolom representasi dari variabel non basis dan misal $\bar{d} = \mathbf{B}^{-1}d$ merupakan ruas kanan termodifikasi dan $\mathbf{C}^B = \begin{bmatrix} c_1 & c_2 & \cdots & c_p \end{bmatrix}$ dan $\mathbf{C}^L = \begin{bmatrix} c_{p+1} & c_{p+2} & \cdots & c_q \end{bmatrix}$ adalah vektor cost yang berkaitan dengan variabel basis dan nonbasis. Dalam bentuk kanonik sebuah program linier setiap variabel basis mempunyai koefisien nol dalam fungsi tujuan. Bentuk khusus fungsi tujuan ini dapat diperoleh dengan melakukan serangkaian

operasi elementer baris yaitu dengan mengalikan setiap kendala i dengan π_i dan mengurangkan dari fungsi tuajuan. Operasi ini memberkan fungsi tujuan yang ekuivalen :

$$\sum_{j=1}^{q} c_j x_j - \left(\sum_{i=1}^{p} \pi_i \left(\sum_{j=1}^{q} a_{ij} x_j - d_i \right) \right) \tag{2.3.8}$$

$$z(x) = \sum_{j=1}^{p} \left(c_j - \sum_{i=1}^{p} \pi_i a_{ij} \right) x_j + \sum_{j=p+1}^{q} \left(c_j - \sum_{i=1}^{p} \pi_i a_{ij} \right) x_j + \sum_{i=1}^{p} \pi_i d_i$$

Untuk mendapatkan bentuk kanonik ini dipilih vektor π sedemikian sehingga

$$c_j - \sum \pi_i a_{ij} = 0 \;\; j \in B \text{ atau } \boldsymbol{\pi}\mathbf{B} = \mathbf{C}^B \tag{2.3.9}$$

dengan $\mathbf{C}^B = \begin{bmatrix} c_1 & c_2 & \cdots & c_p \end{bmatrix}$.Untuk selanjutnya $\boldsymbol{\pi} = \mathbf{B}^{-1}\mathbf{C}^B$ disebut pengali simpleks atau variabel dual yang berkaitan dengan basis B dan $c_j^{\pi} = c_j - \sum \pi_i a_{ij}$ disebut *cost tereduksi* untuk variabel x_j.

2.4 Metode Simpleks Dengan Variabel Terbatas

Variabel keputusan pada program linier pada umumnya hanya mempunyai batasan tanda nonnegatif yaitu $X \geq 0$. Pada beberapa kasus dimana sumber- sumber yang dialokasikan sangat terbatas dapat terjadi batasan $X \leq \mathbf{u}$ dimana u merupakan batas atas dari variabel keputusan. Program linier dengan tambahan kendala $X \leq \mathbf{u}$ dinamakan *masalah program linier dengan variabel terbatas.*

Secara umum masalah program linier dengan variabel terbatas diformulasikan sebagai:

$$Min \quad z(x) = \sum_{j=1}^{p} c_j x_j \tag{2.4.1}$$

kendala $\sum a_{ij} x_j = d_i ; i = 1,2,\cdots,q$ (2.4.2)

$$0 \le x_j \le u_j \text{ untuk semua j} \tag{2.4.3}$$

Metode untuk menyelesaikan masalah diatas disebut *metode simpleks dengan variabel terbatas.*

Selanjutnya didefinisikan solusi fisibel basis dalam metode simpleks dengan variabel terbatas sebagai tripel (B,L,U) dimana B adalah himpunan variabel basis , L himpunan variabel non basis dengan nilai nol dan U adalah himpunan variabel non basis dengan nilai pada batas atas $X = u$ .Misal **B,L,U** adalah partisi gabungan dari matriks kendala **A** yang berasesuaian dengan himpunan B, L, U maka :

$$\mathbf{B}X^B + \mathbf{L}X^L + \mathbf{U}X^U = d \tag{2.4.4}$$

Dalam solusi basis fisibel nilai setiap variabel non basis diambil pada batas bawah atau batas atasnya. Misal u^U adalah batas atas untuk variabel **U** maka $X^L = 0$ dan $X^U = u^U$ dan kemudian dicari :

$$\mathbf{B}X^B = d - \mathbf{U}u^U \text{ atau } X^B = \mathbf{B}^{-1}d - \mathbf{B}^{-1}\mathbf{U}u^U \tag{2.4.5}$$

dimana $\mathbf{B}^{-1}$ adalah invers dari matriks **B.**

Kriteria optimalitas

Dalam bentuk kanonik, fungsi tujuannya dapat ditulis sebagai :

$$z(x) = \sum_{j \in L} c_j^{\pi} x_j + \sum_{j \in U} c_j^{\pi} x_j + z_0 \qquad (2.4.6)$$

Jika $c_j^{\pi} \geq 0; j \in L$ dan $c_j^{\pi} \leq 0; j \in U$ maka $z_0 + \sum_{j \in U} c_j^{\pi} x_j$ merupakan batas bawah dari nilai optimal fungsi tujuan. Kondisi ini disebut *kondisi optimalitas* untuk program linier dengan variabel terbatas. Jika solusi fisibel basis memenuhi kondisi ini maka fungsi tujuannya mencapai batas bawah sehingga nilainya optimal.

Kriteria entering variabel

Pada metode simpleks dengan variabel terbatas dipilih suatu variabel non basis yang melanggar kondisi optimalitas sebagai entering variabel yaitu dengan memilih sebuah variabel non basis x_j yang memenuhi :

a) $c_j^{\pi} < 0$ untuk $j \in L$ (2.4.7)

b) $c_j^{\pi} > 0$ untuk $j \in U$ (2.4.8)

Kriteria Leaving variabel

Jika entering variabel x_s adalah variabel nonbasis dengan nilai batas bawah maka nilai x_s dinaikkan sebesar mungkin dan jika x_s sama dengan batas atas maka nilainya diturunkan sebesar mungkin. Perubahan maksimum ditentukan dengan tetap memperhatikan batasan bahwa nilai seetiap variabel basis dan

variabel non basis x_j tetap berada diantara batas bawah dan batas atas. Jika $\bar{a}_{is} < 0$ dengan menaikkan sebesar θ maka akan diperoleh nialai variabel basis x_j yang meningkat . Pada kasus ini perubahan maksimum yang diperbolehkan oleh batas atas dai x_i yaitu $\theta_i = \frac{u_i - a_i}{-a_{is}}$ .Jika $\bar{a}_{is} > 0$ dengan menaikkan sebesar θ maka akan diperoleh nialai variabel basis x_j yang menurun. Pada kasus ini perubahan maksimum yang diperbolehkan oleh batas bawah dari x_i yaitu $\theta_i = \frac{a_i}{a_{is}}$. Selanjutnya jika $\bar{a}_{is} = 0$ maka diambil $\theta_i = \infty$. Dari semua kasus maka perubahan maksimum yang diijinkan oleh batas atas dan batas bawah x_i yaitu $|u_s|$ Nilai maksimum θ_i yang dipasangkan dengan x_s yaitu nilai minimum dari $|u_s|$ dan $min\{\theta_i ; i = 1,2,\cdots, p\}$.