BAB II

TEORI DASAR

2.1 Matriks

2.1.1 Pengertian

Matriks adalah himpunan skalar (bilangan riil atau kompleks) yang disusun / disejajarkan secara empat persegi panjang (menurut baris – baris dan kolom – kolom).

Skalar – skalar itu disebut elemen matriks. Untuk batasnya kita berikan :

[ ]   atau   ( )   atau   ‖ ‖

Notasi Matriks

Matriks kita beri nama dengan huruf besar A, B, P, C, dan lain – lain. Secara lengkap ditulis matriks A = (a_{ij}) artinya suatu matriks A yang elemen – elemennya a_{ij} dimana indeks i menyatakan baris ke-i dan indeks j menyatakan kolom ke-j dari elemen tersebut.

Secara Umum :

Pandang sebuah matriks A = (a_{ij}), i = 1, 2, …., m ; j = 1,2 ….,n yang berarti bahwa banyaknya baris m dan banyaknya kolom n.

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

Dapat pula kita tuliskan matriks $A_{(mxn)} = (a_{ij})$.

m x n disebut ukuran (ordo) dari matriks A

2.1.2 Operasi Pada Matriks

(a) Penjumlahan Matriks (berlaku untuk matriks – matriks berukuran sama).

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.

Atau $A + B = (a_{ij} + b_{ij})$.

(b) Perkalian Skalar Terhadap Matriks

Kalau λ suatu skalar (bilangan) dan $A = (a_{ij})$ maka matriks $\lambda A = (\lambda a_{ij})$; dengan perkataan lain, matriks λA diperoleh dengan mengalikan semua elemen matriks A dengan λ.

Catatan (1) :

Mengurangi matriks A dengan matriks B, yaitu A – B, adalah menjumlahkan matriks A dengan matriks –B.

Catatan (2) :

Beberapa hukum pada penjumlahan dan perkalian skalar : Kalau A, B, C matriks berukuran sama, dan λ skalar maka :

(1) $A + B = B + A$ (komutatif)

(2) $(A + B) + C = A + (B + C)$ (asosiatif)

(3) $\lambda(A + B) = \lambda A + \lambda B$ (distributif)

(4) Selalu ada matriks D sedemikian sehingga $A + D = B$

(c) Perkalian Matriks

Pada umumnya matriks tidak komutatif terhadap operasi perkalian : $AB \neq BA$. Pada perkalian matriks AB, matriks A kita sebut matriks pertama dan B matriks kedua.

Syarat Perkalian Matriks :

Jumlah banyaknya kolom matriks pertama = jumlah banyaknya baris matriks kedua.

Catatan (3) :

Beberapa hukum pada perkalian matriks : Jika A, B, C matriks – matriks yang memenuhi syarat – syarat perkalian matriks yang diperlukan, maka :

(1) $A(B + C) = AB + AC$, $(B + C)A = BA + CA$, memenuhi hukum distributif

(2) $A(BC) = (AB)C$, memenuhi hukum asosiatif.

(3) Perkalian tidak komutatif, $AB \neq BA$.

(4) Jika $AB = 0$ (matriks nol) yaitu matriks yang semua elemennya = 0, kemungkinan – kemungkinannya :

(i) $A = 0$ dan $B = 0$

(ii) $A = 0$ atau $B = 0$

(iii) $A \neq 0$ dan $B \neq 0$.

(5) Bila AB = AC belum tentu B = C.

2.1.3 Transpose Dari Suatu Matriks

Pandang suatu matriks $A = (a_{ij})$ berukuran (m x n) maka transpose dari A adalah matriks A^T berukuran (n x m) yang didapatkan dari A dengan menuliskan baris ke-i dari A, i = 1, 2, …., m, sebagai kolom ke-i dari A^T.

2.2 Program Linier

2.2.1 Pengantar

Program linier (Linier Programming yang disingkat LP) merupakan salah satu teknik operasi riset yang paling sering digunakan. LP merupakan metode matematik dalam mengalokasikan sumber daya untuk mencapai suatu tujuan seperti memaksimumkan keuntungan dan meminimumkan biaya. LP banyak diterapkan dalam masalah ekonomi, industri, militer, sosial dan lain – lain. LP berkaitan dengan suatu model matematik yang terdiri dari sebuah fungsi tujuan linier dan beberapa kendala linier.

George B. Dantzig diakui umum sebagai pemakai pertama LP, karena jasanya dalam menemukan metode mencari solusi masalah LP dengan banyak variabel keputusan. Dantzig bekerja pada penelitian teknik matematik untuk memecahkan masalah logistik militer ketika ia dipekerjakan oleh angkatan udara Amerika Serikat selama Perang Dunia II. Penelitiannya didukung oleh ahli – ahli lain seperti : J. Von Neuman, L.

Hurwicz dan T.C. Koopmans, yang bekerja pada subyek yang sama. Nama asli teknik ini adalah *program saling ketergantungan kegiatan – kegiatan dalam suatu linier* yang kemudian disingkat menjadi *Linier Programming.*

Setelah perang, banyak ahli bergabung dengan Dantzig dalam pengembangan konsep LP. Paper pertama yang berisi metode solusi yang sekarang dikenal sebagai *metode simpleks* dipublikasikan oleh Dantzig tahun 1947. Dantzig bekerja sama dengan Marshall Wood dan Alex Orden dalam pengembangan metode simpleks. Dalam pengembangan penerapan LP, banyak peneliti seperti : W.W Cooper, A. Henderson, dan W. Orchard bergabung dengan Dantzig. Pada awal, penerapan – penerapan LP banyak dijumpai pada masalah – masalah militer seperti logistik, transportasi dan perbekalan. Kemudian Program Linier segera diterapkan dalam masalah – masalah sektor pemerintah dan swasta. Hasilnya, LP disadari sebagai pendekatan penyelesaian masalah yang sangat ampuh untuk analisa keputusan dalam bidang bisnis. Disamping itu, analisa Input – Output dari Wassily Leontief memberikan suatu dasar untuk menerapkan LP pada analisa ekonomi antar industri. Akhir – akhir ini aplikasi LP telah meningkat dengan perkembangan yang cepat karena dukungan komputer elektronik.

2.2.2 Bentuk Umum Model Program Linier

Dari contoh – contoh yang dibicarakan secara mendalam terlihat adanya suatu pola yang khas untuk merumuskan secara umum suatu masalah LP. Pada setiap masalah, ditentukan variabel keputusan, fungsi tujuan, dan system kendala, yang sama – sama membentuk suatu model matematik dari dunia nyata. Bentuk umum model LP itu adalah :

Maksimumkan (minimumkan) $Z = \sum_{j=1}^{n} c_j x_j$

Dengan syarat : $a_{ij}\, x_j\, (\leq, =, \geq)\, b_i$, untuk semua i $(i = 1, 2, \ldots, m)$

semua $x_j \geq 0$

Keterangan :

x_j : banyaknya kegiatan j, dimana $j = 1, 2, \ldots n$.

berarti disini terdapat n variabel keputusan.

Z : nilai fungsi tujuan.

c_j : sumbangan per unit kegiatan j,

b_i : jumlah sumber daya ke i $(i = 1, 2, \ldots, m)$

berarti terdapat m jenis sumber daya.

a_{ij} : banyaknya sumber daya i yang dikonsumsi sumber daya j.

2.2.3 Bentuk Baku Model Program Linier

Dalam menggunakan metode simpleks untuk menyelesaikan masalah – masalah LP, model LP harus diubah kedalam suatu bentuk umum yang dinamakan " bentuk baku " *(standar form).* Ciri – ciri dari bentuk baku model LP adalah :

1. Semua kendala berupa persamaan dengan sisi kanan non negatif.
2. Semua variabel non negatif.

3. Fungsi tujuan dapat maksimum maupun minimum.

Untuk memudahkan melakukan transformasi ke bentuk baku ikuti contoh berikut :

Kendala

1. Suatu kendala jenis $\leq$ ($\geq$) dapat diubah menjadi suatu persamaan dengan menambahkan suatu variabel *slack* ke (mengurangkan suatu variabel *slack* dari) sisi kiri kendala.

   Contoh :

   - Pada kendala $x_1 + x_2 \leq 15$ ditambahkan suatu *slack* $S_1 \geq 0$ pada sisi kiri untuk mendapatkan persamaan $x_1 + x_2 + S_1 = 15$. Jika kendala menunjukkan keterbatasan penggunaan suatu sumber daya, S_1 akan menunjukkan *slack* atau jumlah sumber daya yang tak digunakan.
   - Pada kendala $3x_1 + 2x_2 - 3x_3 \geq 5$ dikurangkan $S_2 \geq 0$ pada sisi kiri untuk memperoleh persamaan

     $3x_1 + 2x_2 - 3x_3 - S_2 = 5$

2. Sisi kanan suatu persamaan dapat selalu dibuat non negatif dengan mengalikan kedua sisi dengan -1.

   Contoh :

   $-5x_1 + x_2 = -25$ adalah ekuivalen secara matematik dengan

   $5x_1 - x_2 = 25$.

3. Arah pertidaksamaan dibalik jika kedua sisi dikalikan -1.

   Contoh :

$-5x_1 + x_2 \leq -25$ dapat diganti dengan $5x_1 - x_2 \geq 25$.

Fungsi Tujuan

Meskipun model LP dapat berjenis maksimisasi maupun minimisasi, kadang – kadang bermanfaat untuk mengubah salah satu bentuk ke bentuk lain. Maksimisasi dari suatu fungsi adalah ekuivalen dengan minimisasi dari negatif fungsi yang sama, dan sebaliknya.

Contoh :

Maks. $Z = 50x_1 + 80x_2 + 60x_3$ ekuivalen secara matematik dengan

Min. $(-Z) = -50x_1 - 80x_2 - 60x_3$

Ekuivalen berarti bahwa untuk seperangkat kendala yang sama, nilai optimum x_1, x_2, x_3 , adalah sama pada kedua kasus. Perbedaannya pada nilai fungsi tujuan, meski besarnya angka sama, tetapi tandanya berlawanan.

2.3 Metode Simpleks

Metode simpleks pertama kali diperkenalkan oleh George B. dantzig pada tahun 1947 dan telah diperbaiki oleh beberapa ahli lain. Metode ini menyelesaikan masalah LP melalui perhitungan ulang *(iteration)* dimana langkah – langkah perhitungan yang sama diulang berkali – kali sebelum solusi optimum dicapai.

Dalam penyelesaian masalah LP dengan grafik, telah dinyatakan bahwa solusi optimum selalu terletak pada titik pojok ruang solusi. Metode simpleks didasarkan pada gagasan ini.

1. Dimulai pada suatu titik pojok yang layak, biasanya titik asal (yang disebut sebagai solusi awal).
2. Bergerak dari satu titik pojok layak ke titik pojok layak lain yang berdekatan. Pergerakan ini akan menghasilkan nilai fungsi tujuan yang lebih baik (meningkat untuk masalah maksimisasi dan menurun untuk masalah minimisasi). Jika solusi yang lebih baik telah diperoleh, prosedur simpleks dengan sendirinya akan menghilangkan semua solusi – solusi lain yang kurang baik.
3. Proses ini diulang – ulang sampai suatu solusi yang lebih baik tak dapat ditemukan. Proses simpleks kemudian berhenti dan solusi optimum diperoleh.

Karena kesulitan menggambarkan grafik berdimensi banyak, maka penyelesaian masalah LP yang melibatkan lebih dari dua variabel menjadi tak praktis atau tidak mungkin. Dalam keadaan ini kebutuhan metode solusi yang lebih umum menjadi nyata. Metode umum itu dikenal dengan nama Algoritma Simpleks yang dirancang untuk menyelesaikan seluruh masalah LP, baik yang melibatkan dua variabel atau lebih dua variabel.

Mengubah bentuk baku model LP ke dalam bentuk tabel akan memudahkan proses perhitungan simpleks. Langkah – langkah perhitungan dalam algoritma simpleks adalah :

1. Berdasar bentuk baku, tentukan solusi awal (*initial basic feasible solution*) dengan menetapkan n-m variabel non basis sama dengan nol. Dimana n jumlah variabel dan m banyaknya kendala.
2. Pilih sebuah *entering variable* diantara yang sedang menjadi variabel non basis, yang jika dinaikkan diatas nol, dapat memperbaiki nilai fungsi tujuan. Jika tak ada, berarti solusi sudah optimal. Jika tidak, menuju ke langkah 3.
3. Pilih sebuah *leaving variable* diantara yang sedang menjadi variabel basis yang harus menjadi non basis (nilainya menjadi nol) ketika *entering variable* menjadi variabel basis.
4. Menentukan solusi yang baru dengan membuat *entering variable* dan *leaving variable* menjadi non basis. Kembali ke tahap 2.

2.4 Program Non Linier

2.4.1 Pengantar

Pemrograman non linier adalah suatu bentuk pemrograman yang memiliki fungsi tujuan non linier dan fungsi kendala linier atau non linier.

Hal ini didasarkan pada penemuan beberapa contoh penerapan dalam dunia usaha bisnis yang menggunakan asumsi ketaklinieran dalam membuat perencanaan. Secara umum masalah pemrograman non linier adalah menentukan $x = (x_1, x_2, ... x_n)$ sehingga $f(x)$ maksimum / minimum, dengan kendala $g_i(x) \le b_i$, untuk setiap i = 1,2,....n dan $x_i \ge 0$ dimana fungsi $f(x)$ non linier dan $g_i(x)$ linier atau non linier .

2.4.2 Matriks Hessian dan Fungsi Konveks

Beberapa konsep dalam matriks dan aljabar seperti matriks Hessian dan fungsi konveks merupakan pengetahuan minimum yang diharapkan untuk mempelajari program non linier

Matriks Hessian

Diberikan f($x_1, x_2, \ldots, x_n$) adalah sebuah fungsi dengan n variabel, $(x_1, x_2, \ldots\ldots x_n)$. Matriks Hessian adalah matriks yang merupakan turunan parsial kedua dari fungsi tersebut dengan susunan seperti berikut :

$$(H) = \begin{bmatrix} f_{11} & f_{12} & \ldots & f_{1n} \\ f_{21} & f_{22} & \ldots & f_{2n} \\ . & . & \ldots & . \\ . & . & \ldots & . \\ f_{n1} & f_{n2} & \ldots & f_{nn} \end{bmatrix}$$

$$f_{11} = \frac{\partial^2 f}{(\partial x_1)^2} \qquad f_{1n} = \frac{\partial^2 f}{\partial x_1 \partial x_n}$$

$$f_{2n} = \frac{\partial^2 f}{\partial x_2 \partial x_n} \qquad f_{nn} = \frac{\partial^2 f}{(\partial x_n)^2}$$

Definisi :

Jika terdapat suatu matriks berukuran (n x n), maka *principal minor ke k* *($k \leq n$)* adalah suatu sub matriks dengan ukuran (k x k) yang diperoleh dengan menghapus (n – k) baris dan kolom yang bersesuaian dari matriks tersebut.

Contoh :

$$(Q) = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Principal minor ke-1 adalah elemen – elemen yang diagonal yaitu 1, 5, 9.

Principal minor ke-2 adalah matriks – matriks (2 x 2) berikut :

$$\begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix} \qquad \begin{bmatrix} 1 & 3 \\ 7 & 9 \end{bmatrix} \qquad \begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix}$$

Principal minor ke-3 adalah matriks Q itu sendiri.

Determinan dari suatu *principal minor* dinamakan *principal determinant.*

Leading principal minor ke k dari suatu matriks (n x n) diperoleh dengan menghapus (n – k) baris terakhir dan kolom yang bersesuaian. Dengan matriks Q diatas *leading principal minor* ke-1 adalah 1 (hapus dua baris terakhir dan dua kolom terakhir). *Leading principal minor* ke-2 adalah :

$$\begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix}$$

Sementara yang ke-3 adalah matriks Q itu sendiri. Banyaknya *leading principal determinant* dari suatu matriks (n x n) adalah n. Determinan dari *leading principal minor* dinamakan *leading principal determinant.*

Ada cara pengujian yang sederhana untuk menentukan apakah suatu matriks adalah *definit positif, semidefinit positif, definit negatif,* *semidefinit negatif* atau *indefinit.* Semua pengujian ini berlaku hanya jika matriksnya simetris.

- Ketentuan uji bagi matriks *definit positif* adalah :
  (1) Semua elemen diagonal harus positif.
  (2) Semua *leading principal determinant* harus positif.
- Ketentuan uji untuk matriks *semidefinit positif* adalah :
  (1) Semua elemen diagonal ini negatif.
  (2) Semua *leading principal determinant* non negatif.

Untuk membuktikan bahwa suatu matriks *definit negatif (semidefinit negatif)*, uji negatif dari matriks itu untuk *definit positif (semidefinit positif)*. Suatu uji cukup bagi suatu matriks menjadi *indefinit* adalah bahwa sekurang – kurangnya dua elemen diagonalnya memiliki tanda berlawanan.

Contoh 1:

$$f(x) = 7x_1^2 + 10x_2^2 + 7x_3^2 - 4x_1x_2 + 2x_1x_3 - 4x_2x_3$$

$$\text{maka } (H) = \begin{bmatrix} 14 & -4 & 2 \\ -4 & 20 & -4 \\ 2 & -4 & 14 \end{bmatrix}$$

dengan *leading principal determinant* $H_1 = 14, H_2 = 264, H_3 = 3456$

sehingga (H) *definit positif.*

Contoh 2 :

$$f(x) = -x_1^2 - x_2^2 - x_3^2 + 3x_1x_2 - 3x_1x_3 + 4x_2x_3$$

$$\text{maka } (H) = \begin{bmatrix} -2 & 3 & -3 \\ 3 & -2 & 4 \\ -3 & 4 & -2 \end{bmatrix}$$

dengan *leading principal determinant* $H_1 = -2, H_2 = -5, H_3 = -12$

sehingga (H) *definit negatif.*

Fungsi Konveks dan Fungsi Konkaf

Semua fungsi f(x) adalah suatu fungsi konkaf jika dan hanya jika - f(x) adalah suatu fungsi konveks. Untuk mengetahui apakah suatu fungsi konveks atau konkaf digunakan pengujian sebagai berikut :

Suatu fungsi f adalah fungsi konveks jika matriks Hessian dari fungsi f adalah *definit positif* atau *semidefinit positif.*

Suatu fungsi adalah konkaf jika matriks Hessian dari fungsi f adalah *definit negatif* atau *semidefinit negatif.*

Contoh : $f(x_1, x_2, x_3) = 3x_1^2 + 2x_2^2 + x_3^2 - 2x_1x_2 - 2x_1x_3 + 2x_2x_3$

$$(H) = \begin{bmatrix} 6 & -2 & -2 \\ -2 & 4 & 2 \\ -2 & 2 & 2 \end{bmatrix}$$

dengan *leading principal determinant* adalah

$H_1 = 6, H_2 = 20$ dan $H_3 = 16$. Sehingga (H) adalah suatu matriks *definit positif* yang berarti f adalah fungsi konveks.

2.5 Syarat Kuhn Tucker

Kuhn tucker (1951) telah memperluas teori untuk menyelesaikan masalah program non linier umum baik dengan kendala persamaan maupun pertidaksamaan. Sekarang kita tinjau beberapa persamaan untuk menentukan penyelesaian optimal suatu masalah pemrograman non linier (dengan fungsi-fungsi yang dapat diturunkan). Suatu syarat perlu untuk optimasi adalah bahwa u_i non negatif untuk semua masalah maksimasi dan non positif untuk semua masalah minimasi. Adapun syarat perlu untuk masalah optimasi yang berkendala diperlukan syarat Karush Kuhn Tucker yaitu: $f(x)$, $g_i(x)$ adalah fungsi-fungsi yang dapat diturunkan maka $x = (x_1, x_2, \ldots\ldots x_n)$ merupakan penyelesaian optimal apabila terdapat $(u_1, u_2, \ldots u_m)$ sehingga semua syarat berikut terpenuhi :

1. $\dfrac{\partial f(x)}{\partial x_j} - \sum_{i=1}^{m} u_i \dfrac{\partial g_i(x)}{\partial x_j} \leq 0$
2. $x_j \left[\dfrac{\partial f(x)}{\partial x_j} - \sum_{i=1}^{m} u_i \dfrac{\partial g_i(x)}{\partial x_j} \right] = 0$
3. $g_i(x) - b_i \leq 0$
4. $u_i(g_i(x) - b_i) = 0$
5. $u_i \geq 0$
6. $x_j \geq 0$

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

Sedangkan syarat cukupnya yaitu , misalkan f(x) fungsi konkaf dan $g_1(x), g_2(x), \ldots\ldots\ldots g_m(x)$ fungsi-fungsi yang konveks maka

$x=(x_1,x_2,.......x_n)$ merupakan penyelesaian optimal jika dan hanya jika semua kondisi diatas terpenuhi. Syarat perlu Kuhn Tucker juga merupakan syarat cukup jika fungsi tujuan dan ruang solusi memenuhi syarat-syarat tertentu yang berkaitan dengan fungsi konveks dan fungsi konkaf.