

BAB II

KAJIAN TEORI

A. Perencanaan Menu Diet

1. Pengertian Perencanaan Menu Diet.

Mengingat bahwa diet merupakan obat utama yang dapat menekan timbulnya *diabetes mellitus* (DM) dan dapat menekan kemungkinan komplikasi akut maupun kronik, penulis akan menguraikan perencanaan diet DM di RS PKU Muhammadiyah Yogyakarta.

Setiap macam diet diusahakan agar dapat memenuhi beberapa persyaratan sebagai berikut, yaitu bahwa diet tersebut diharapkan dapat (Askandar Tjokropawiro, 1997):

- a. memperbaiki kesehatan umum penderita,
- b. menyesuaikan berat badan penderita ke berat badan normal,
- c. menormalkan pertumbuhan DM anak atau *diabetes mellitus* dewasa muda (masa pertumbuhan),
- d. mempertahankan glukosa darah sekitar normal,
- e. menekan atau menunda timbulnya *angiopati diabetik*,
- f. memberikan modifikasi diet sesuai dengan keadaan penderita misalnya penderita DM yang hamil, DM dengan penyakit hati, TBC,
- g. menarik serta mudah diterima penderita.

Pola Makan Penderita *Diabetes Melitus*

Pola makan adalah makanan yang seimbang baik antara zat gizi karbohidrat, protein, lemak, vitamin dan mineral. Makanan yang seimbang adalah makanan

yang tidak mementingkan salah satu zat gizi tertentu dan dikonsumsi sesuai dengan kebutuhan seseorang. Dengan demikian pola makan dapat diartikan sebagai suatu cara untuk melakukan kegiatan makan secara sehat. Pola makan adalah suatu cara atau usaha dalam pengaturan jumlah dan jenis makanan dengan maksud tertentu seperti mempertahankan kesehatan, status nutrisi, mencegah atau membantu kesembuhan penyakit. Pola makan sehari-hari merupakan pola makan seseorang yang berhubungan dengan kebiasaan makan setiap harinya.

Diet yang baik merupakan kunci keberhasilan penatalaksanaan diabetes. Diet yang dianjurkan adalah makanan dengan komposisi yang seimbang dalam hal karbohidrat, protein, dan lemak. Tujuan pengobatan diet pada diabetes adalah:

- a. mencapai dan kemudian mempertahankan kadar glukosa darah mendekati kadar normal,
- b. mencapai dan mempertahankan *lipid* mendekati kadar yang optimal,
- c. mencegah komplikasi akut dan kronik,
- d. meningkatkan kualitas hidup.

Syarat kebutuhan kalori untuk penderita DM harus sesuai untuk mencapai kadar glukosa normal dan mempertahankan berat badan normal. Komposisi energi adalah 60-70 % dari karbohidrat, 10-15 % dari protein, 20-25 % dari lemak. Makan dengan aneka ragam makanan yang mengandung sumber zat tenaga, sumber zat pembangun serta zat pengatur.

- a. Makanan sumber zat tenaga mengandung zat gizi karbohidrat, lemak, dan protein yang bersumber dari nasi serta penggantinya seperti: roti, mie, kentang, dan lain-lain.

- b. Makanan sumber zat pembangun mengandung zat gizi protein dan mineral. Makanan sumber zat pembangun seperti kacang-kacangan, tempe, tahu, telur, ikan, ayam, daging, susu, keju, dan lain-lain.
- c. Makanan sumber zat pengatur mengandung vitamin dan mineral. Makanan sumber zat pengatur antara lain: sayuran dan buah-buahan.

Banyak yang beranggapan bahwa penderita DM harus makan makanan khusus, anggapan tersebut tidak selalu benar karena tujuan utamanya adalah menjaga kadar glukosa darah pada batas normal. Untuk itu sangat penting bagi kita terutama penderita DM untuk mengetahui efek dari makanan pada glukosa darah. Jenis makanan yang dianjurkan untuk penderita DM adalah makanan yang kaya serat seperti sayur-mayur dan buah-buahan segar. Hal yang terpenting adalah jangan terlalu mengurangi jumlah makanan karena akan mengakibatkan kadar gula darah yang sangat rendah (hypoglikemia) dan juga jangan terlalu banyak makan makanan yang memperparah penyakit DM.

Ada beberapa jenis makanan yang dianjurkan bagi penderita DM yaitu:

- a. sumber karbohidrat kompleks seperti nasi, roti, mie, kentang, singkong, ubi dan sagu,
- b. sumber protein rendah lemak seperti ikan, ayam tanpa kulitnya, susu skim, tempe, tahu dan kacang-kacangan,
- c. sumber lemak dalam jumlah terbatas yaitu bentuk makanan yang mudah dicerna terutama mudah diolah dengan cara dipanggang, dikukus, disetup, direbus, dan dibakar.

2. Jenis-jenis Diet *Diabetes Mellitus*.

Jenis-jenis diet DM telah terbagi sesuai dengan jumlah kalorinya, yaitu:

- a. *Diabetes Mellitus* I (1100 kalori)
- b. *Diabetes Mellitus* II (1300 kalori)
- c. *Diabetes Mellitus* III (1500 kalori)
- d. *Diabetes Mellitus* IIV (1700 kalori)
- e. *Diabetes Mellitus* V (1900 kalori)
- f. *Diabetes Mellitus* VI (2100 kalori)
- g. *Diabetes Mellitus* VII (2300 kalori)
- h. *Diabetes Mellitus* VIII (2500 kalori)

Penentuan gizi penderita dilakukan dengan menghitung Berat Badan Ideal (BBI) dengan rumus

$$BBI = (TB - 100) - (10\% \times (TB - 100)) \quad (2.1)$$

dengan:

TB = tinggi badan (cm).

Setelah menghitung BBI selanjutnya adalah menentukan energi (kalori) dengan menggunakan rumus

Pria : $BBI \times 30$ kalori sehari

Wanita : $BBI \times 25$ kalori sehari

3. Faktor-faktor Dalam Perencanaan Menu Diet *Diabetes Mellitus*.

Terdapat faktor-faktor lain yang dapat mempengaruhi penentuan total energi seorang penderita DM berdasarkan RS. PKU Muhammadiyah Yogyakarta, yaitu:

1. Faktor Usia

Pada rentan usia tertentu dapat mempengaruhi energi (kalori) sebagai berikut:

40 – 54 tahun	: -5% dari energi
60 – 69 tahun	: -10% dari energi
>70 tahun	: -20% dari energi

2. Faktor aktivitas

Aktivitas yang dilakukan oleh penderita sehari-hari juga mempengaruhi energi (kalori) sebagai berikut:

Istirahat	: +10% dari energi
Ringan	: +20% dari energi
Sedang	: +30% dari energi
Berat	: +50% dari energi

3. Faktor Berat Badan

Pada kasus diet DM faktor berat badan hanya dibedakan menjadi dua, yaitu kategori kurus dan gemuk. Untuk menentukannya dilakukan dengan menghitung *Body Mass Index* (BMI) atau Indeks Massa Tubuh (IMT) dengan rumus:

$$IMT/BMI = \frac{BB}{TB^2} \quad (2.2)$$

dengan:

BB : berat badan (kg),

TB : tinggi badan (m).

Seseorang dikatakan memiliki berat badan ideal jika nilai IMT antara 18,5 – 23,5. Untuk nilai IMT < 18,5 dikatakan bahwa orang tersebut memiliki berat badan kurang (kurus), sedangkan untuk nilai IMT > 23,5 dikatakan bahwa orang tersebut memiliki berat badan berlebih (gemuk).

Setelah mengetahui penderita tersebut termasuk kategori yang mana, dapat diterapkan:

Kurus : + (20-30%) dari energi

Gemuk : - (20-30%) dari energi

Selanjutnya, setelah semua diperoleh jumlahkan semua hasil perhitungan, sebagai berikut:

$$E = energi\ (kalori) + kalori\ umur + kalori\ aktivitas \\ +\ kalori\ berat\ badan$$

4. Penentuan Menu Diet Diabetes Mellitus

Penentuan menu diet dilakukan berdasarkan pada standar diet DM dengan metode bahan makanan penukar, yaitu penggolongan bahan makanan berdasarkan nilai gizi yang setara. Bahan makanan penukar ini dibagi menjadi 8 golongan bahan pangan, yaitu sumber karbohidrat, sumber protein hewani, sumber protein nabati, sayuran, buah dan gula, susu, minyak, dan makanan tanpa kalori. Satu satuan penukar untuk setiap jenis pangan pada bahan pangan tersebut masing-masing mempunyai kandungan karbohidrat, protein, dan lemak.

Tabel 2.1. Standar Diet *Diabetes Mellitus*

Pukul	Gol. Makanan	Energi (K kalori)							
		1100	1300	1500	1700	1900	2100	2300	2500
07.00	Nasi	½	1	1	1	1 ½	1 ½	1 ½	2
	Hewani	1	1	1	1	1	1	1	1
	Nabati	-	-	½	½	½	1	1	1
	Sayur A	S	S	S	S	s	S	S	S
	Minyak	1	1	1	1	2	2	2	2
10.00	Roti	-	-	-	-	-	-	½	½
	Buah	1	1	1	1	1	1	1	1
13.00	Nasi	1	1	2	2	2	2	2	2
	Hewani	1	1	1	1	1	1	1	1
	Nabati	1	1	1	1	1	1	1	2
	Sayur A	S	S	S	S	s	S	S	S
	Sayur B	1	1	1	1	1	1	1	1
	Buah	1	1	1	1	1	1	1	1
	Minyak	1	2	2	2	2	2	2	2
16.00	Roti	-	-	-	-	-	1	1	1
	Margarin	-	-	-	-	-	1	1	1
	Buah	1	1	1	1	1	1	1	1
19.00	Nasi	1	1	1	2	2	2	2	2
	Hewani	1	1	1	1	1	1	1	1
	Nabati	1	1	1	1	1	1	1	1
	Sayur A	S	S	S	S	s	S	S	S
	Sayur B	1	1	1	1	1	1	1	1
	Buah	1	1	1	1	1	1	1	1
	Minyak	1	1	1	1	2	2	2	2
21.00	Susu	-	-	-	-	-	-	1	1

Standar yang dianjurkan adalah makanan dengan komposisi yang seimbang dalam presentase karbohidrat, protein, dan lemak. Komposisi energi untuk mencapai dan mempertahankan Berat Badan Ideal (BBI) adalah 60%-70% dari karbohidrat, 10%-15% dari protein, dan 20%-25% dari lemak.

Misal:

KH : karbohidrat (gram),

P : protein (gram),

L : lemak (gram),

PK : presentase karbohidrat (60%-70%),

PP : presentase protein (10%-15%),

PL : presentase lemak (20%-25%),

E : energi (kalori).

Komposisi energi tersebut berdasarkan bagian Gizi di RS PKU Muhammadiyah Yogyakarta dapat dihitung dengan rumus:

$$KH = \frac{PK \times E}{4} \quad (2.3)$$

$$P = \frac{PP \times E}{4} \quad (2.4)$$

$$L = \frac{PL \times E}{9} \quad (2.5)$$

B. Optimisasi

Optimisasi adalah suatu pendekatan normatif untuk mengidentifikasi penyelesaian terbaik dalam pengambilan keputusan dari suatu permasalahan. Penyelesaian permasalahan dalam optimisasi ditujukan untuk memperoleh titik maksimum atau titik minimum dari fungsi yang dioptimumkan. Seperti permasalahan suatu perusahaan dalam menentukan jumlah produksi agar keuntungan maksimum dan biaya minimum dapat diperoleh (Rio Armino, 2006).

Komponen penting dari permasalahan optimum adalah fungsi tujuan, yang dalam beberapa hal sangat tergantung pada peubah. Penyelesaian masalah optimisasi dengan program matematika dapat dilakukan melalui program linear, program tak linear, program integer dan program dinamik. Fungsi tujuan secara umum merupakan langkah minimisasi biaya atau penggunaan bahan-bahan baku, dan sebagainya. Penentuan fungsi tujuan dikaitkan dengan permasalahan yang dihadapi. Penentuan kondisi optimum dikenal sebagai pemrograman teknik matematik. Tujuan dan kendala-kendala dalam pemrograman matematik diberikan

dalam bentuk fungsi-fungsi matematika dan hubungan fungsional (hubungan keterkaitan). Hubungan keterkaitan tersebut dapat diartikan sebagai hubungan yang saling mempengaruhi, hubungan interaksi, timbal-balik, dan saling menunjang (Rio Armino, 2006).

C. Pemrograman Linear

Pemrograman linear merupakan salah satu teknik penelitian operasional yang digunakan untuk menyelesaikan masalah optimisasi suatu model linier dengan keterbatasan-keterbatasan sumber daya yang tersedia. *Pemrograman linear* mengalami perkembangan pesat setelah masa perang dunia karena banyak industri yang menggunakannya. George B. Dantzig menemukan metode simpleks pada tahun 1947, sedangkan John Von Neumann menemukan teori dualitasnya ditahun yang sama. Selanjutnya berbagai alat dan metode dikembangkan untuk menyelesaikan masalah program linear bahkan sampai pada masalah penelitian operasional hingga tahun 1950-an seperti pemrograman dinamik, teori antrian, teori persandian, dan *goal programming*. *Pemrograman linear* merupakan dasar dari *goal programming* (Siswanto, 2007).

Secara umum, fungsi pada *pemrograman linear* ada dua macam yaitu fungsi tujuan dan fungsi kendala. Fungsi tujuan digunakan untuk menentukan nilai optimum dari fungsi tersebut yaitu nilai maksimal untuk masalah keuntungan dan nilai minimal untuk masalah biaya. Fungsi kendala diperlukan berkenaan dengan adanya keterbatasan sumber daya yang tersedia. Tujuan utama dari *linear programming* adalah menentukan nilai optimum (maksimal/minimal) dari fungsi

yang telah ditetapkan, untuk memecahkan permasalahannya yaitu dengan metode grafik, dan metode simpleks.

1. Formulasi Model Pemrograman Linear

Model *pemrograman linear* merupakan model matematis perumusan masalah umum pengalokasian sumber daya untuk berbagai kegiatan di mana bentuk dan susunan dalam penyajian masalah-masalahnya dipecahkan dengan teknik *pemrograman linear*. Model *pemrograman linear* mempunyai tiga unsur utama yaitu variabel keputusan, fungsi tujuan, dan fungsi kendala. Variabel keputusan merupakan variabel persoalan yang akan mempengaruhi nilai tujuan yang ingin dicapai. Penemuan variabel keputusan tersebut harus dilakukan terlebih dahulu sebelum merumuskan fungsi tujuan dan kendala-kendalanya. Fungsi tujuan adalah tujuan yang hendak dicapai dalam model *pemrograman linear* yang diwujudkan ke dalam sebuah fungsi matematika linear. Selanjutnya fungsi tersebut dimaksimumkan atau diminimumkan terhadap kendala-kendala yang ada. Sedangkan fungsi kendala merupakan fungsi matematis dari berbagai kendala untuk mewujudkan tujuan-tujuannya.

Langkah-langkah memformulasikan masalah *pemrograman linear* menjadi model matematika adalah sebagai berikut:

1. menentukan variabel keputusan dan menyatakan dalam simbol matematik,
2. membentuk fungsi tujuan yang ditunjukkan sebagai suatu hubungan linear (bukan perkalian) dari variabel keputusan,

3. menentukan semua kendala masalah dalam persamaan atau pertidaksamaan yang merupakan hubungan linear dari variabel keputusan yang menunjukkan keterbatasan sumber daya masalah tersebut.

Misal:

Z : nilai fungsi tujuan,

c_j : sumbangan per unit kegiatan, dengan $j = 1, 2, 3, \dots, n$,

x_j : banyaknya kegiatan j , dengan $j = 1, 2, 3, \dots, n$,

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

b_i : jumlah sumber daya i ($i = 1, 2, 3, \dots, m$).

Secara umum, suatu masalah model *programan linear* dapat diformulasikan sebagai berikut.

Fungsi tujuan

Memaksimumkan atau meminimumkan

$$Z = c_1x_1 + c_2x_2 + c_3x_3 + \dots + c_nx_n \quad (2.6)$$

Dengan kendala,

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1 \quad (2.7)$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2$$

$\vdots$

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m$$

untuk $i = 1, 2, 3, 4, \dots, m$

$$x_j \geq 0 \quad \text{untuk } j = 1, 2, 3, 4, \dots, n$$

2. Penyelesaian Model Pemrograman Linear

Penyelesaian *pemrograman linear* dapat diselesaikan dengan metode grafik atau metode simpleks. Masalah *pemrograman linear* dengan dua variabel atau tiga variabel yang dapat disusutkan masih bisa diselesaikan dengan menggunakan metode grafik, sedangkan untuk masalah *pemrograman linear* yang memuat tiga atau lebih variabel dan tidak dapat disusutkan menjadi masalah yang mempunyai dua variabel saja harus menggunakan metode simpleks (Pangestu Subagyo, 1995).

a. Pengertian Metode Simpleks

Metode simpleks ini merupakan suatu metode untuk penyelesaian masalah program linear dengan menggunakan perhitungan ulang atau *iterasi* yang mana perhitungannya diulang berkali-kali dengan langkah-langkah yang sama sampai solusi yang optimal tercapai. Metode simpleks juga merupakan suatu algoritma yang dilakukan berulang untuk menemukan penyelesaian optimal dari suatu masalah *pemrograman linear* dengan cara menguji titik-titik sudutnya. Proses algoritma tersebut mencakup prosedur kapan mulai pemecahan dan kapan berakhirnya proses iterasi. Secara umum algoritma tersebut adalah sebagai berikut (Pangestu Subagyo, 1995):

1. tahap awal yaitu menyusun tabel awal sebagai pangkal proses iterasi,
2. proses iterasi dilakukan secara berulang hingga mencapai hasil optimal yang dikehendaki,
3. proses berhenti apabila hasil optimal tercapai atau bahkan tidak dapat dicapai sama sekali.

Pada suatu masalah *pemrograman linear*, kendala terlebih dahulu diubah ke bentuk kanonik. Bentuk kanonik adalah bentuk dari sistem persamaan linear dan memuat variabel basis (variabel yang berkoefisien 1). Sedangkan dalam hal membentuk kendala tersebut ke bentuk kanonik dibutuhkan penambahan variabel basis baru, yaitu:

1. *Slack Variable*, merupakan suatu variable yang berfungsi untuk menampung sisa kapasitas pada kendala berupa pembatas yang memuat hubungan kurang dari atau sama dengan ($\leq$) membuat nilai ruas kiri dan ruas kanan bernilai sama. Contoh:

Terdapat suatu fungsi kendala:

$$2x_1 + x_2 \leq 20$$

tambahkan *slack variable* pada kendala tersebut, sehingga menjadi:

$$2x_1 + x_2 + S_1 = 20$$

2. *Surplus Variable*, merupakan variabel yang berfungsi untuk menampung kelebihan nilai dari ruas kiri suatu pertidaksamaan yang memuat hubungan lebih dari atau sama dengan ($\geq$) pada kendala yang berupa syarat. Contoh:

Terdapat suatu fungsi kendala:

$$x_1 + 4x_2 \geq 12$$

Tambahkan *surplus variable* pada kendala tersebut, sehingga menjadi:

$$x_1 + 4x_2 - S_1 = 12$$

3. *Artificial variabel*, merupakan suatu variabel yang ditambahkan pada fungsi kendala dikarenakan fungsi kendala tersebut belum memuat variabel basis. Tidak seperti *slack variable* dan *surplus variable* yang tidak berpengaruh

terhadap fungsi tujuan dikarenakan memiliki koefisien nol, *artificial variabel* mempunyai parameter M yaitu berupa bilangan yang nilainya sangat besar.

Contoh:

$$x_1 + 4x_2 - S_1 = 12$$

Tambahkan *artificial variable* $d \geq 0$ sehingga menjadi:

$$x_1 + 4x_2 - S_1 + \mathbf{d}_1 = 12$$

b. Tabel Simpleks

Metode simpleks memerlukan suatu tabel simpleks pada pengujian suatu titik sudut untuk menentukan apakah variabel keputusan pada titik-titik sudut tersebut telah menghasilkan nilai tujuan yang ekstrem. Dikarenakan hal tersebut untuk mengetahui keekstreman nilai fungsi tujuan, padahal nilai ekstrem tersebut bisa minimum ataupun maksimum, maka arah keekstreman tersebut dapat mempengaruhi proses penyelesaian tabel.

Misal:

x_j : variabel-variabel keputusan,

$\bar{x}_i$: variabel yang menjadi basis dalam tabel yang ditinjau,

a_{ij} : koefisien teknis,

b_i : suku tetap (tak negatif),

c_j : koefisien ongkos,

$\bar{c}_i$: koefisien ongkos milik variabel basis $\bar{x}_i$,

z_j : $\sum_{i=1}^m \bar{c}_i a_{ij}$ (hasil kali dari $\bar{c}_i$ dengan kolom a_{ij}),

$z_j - c_j$: selisih z_j dengan c_j ,

Z : $\sum_{j=1}^m \bar{c}_i b_i$ (hasil kali dari $\bar{c}_i$ dengan b_i),

R_i : rasio antara b_i dengan c_j .

Model matematis masalah *pemrograman linear* yang mengandung matriks identitas, yaitu:

$$\begin{aligned}
 a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + S_1 &= b_1 \\
 a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n + S_2 &= b_2 \\
 \vdots + \vdots + \vdots + \vdots + \vdots & \\
 a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n + S_m &= b_m \\
 x_1, x_2, \dots, x_n &\geq 0
 \end{aligned}$$

Model matematis tersebut kemudian dimasukkan dalam tabel simpleks, akan diperoleh bentuk baku tabel simpleks atau tabel awal seperti pada tabel 2.2.

Tabel 2.2. Tabel Simpleks dalam Bentuk Simbol

	c_j	c_1	c_2	$\cdots$	c_n	0	0	$\cdots$	0		
$\bar{c}_i$	$\bar{x}_i/x_j$	x_1	x_2	$\cdots$	x_n	S_1	S_2	$\cdots$	S_n	b_i	R_i
0	S_1	a_{11}	a_{12}	$\cdots$	a_{1n}	1	0	$\cdots$	0	b_1	R_1
0	S_2	a_{21}	a_{22}	$\cdots$	a_{2n}	0	1	$\cdots$	0	b_2	R_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$
0	S_m	a_{m1}	a_{m2}	$\cdots$	a_{mn}	0	0	$\cdots$	1	b_m	R_m
	z_j	z_1	z_2	$\cdots$	z_n	0	0	$\cdots$	0	Z	
	$z_j - c_j$	$z_1 - c_1$	$z_2 - c_2$	$\cdots$	$z_n - c_n$	0	0	$\cdots$	0	Z	

Langkah-langkah penyelesaian *pemrograman linear* dengan menggunakan metode simpleks, yaitu:

1. Mengubah masalah *pemrograman linear* ke bentuk kanonik.
2. Susun ke dalam tabel simpleks.
3. Menentukan variabel masuk (*entering variable*).

Memilih kolom kunci yang memiliki nilai pada garis fungsi tujuan yang bernilai negatif terbesar.

4. Menentukan variabel keluar (*leaving variable*).

Memilih baris kunci dengan mencari indeks tiap-tiap baris dengan membagi nilai-nilai pada kolom b_i dengan nilai sebaris pada kolom kunci.

5. Menentukan persamaan *pivot* baru.

Mengubah nilai pada baris kunci dengan cara membaginya dengan angka kunci.

6. Menentukan persamaan-persamaan baru selain persamaan *pivot* baru.
7. Merevisi tabel simpleks mulai dari langkah 3 sampai langkah 6 hingga nilai optimum tercapai.

Berikut akan diberikan suatu contoh penggunaan model *pemrograman linear*:

Contoh 2.1:

Memaksimumkan: $z = 40x + 30y$

Dengan kendala,

$$2x + 3y \leq 60$$

$$2y \leq 30$$

$$2x + y \leq 40$$

$$x, y \geq 0$$

Berdasarkan langkah- langkah penyelesaian *pemrograman linear*, masalah di atas terlebih dahulu diubah menjadi bentuk kanonik, berikut model kanonik:

$$\text{Maksimumkan: } z = 40x + 30y + 0S_1 + 0S_2 + 0S_3$$

dengan kendala,

$$2x + 3y + S_1 = 60$$

$$2y + S_2 = 30$$

$$2x + y + S_3 = 40$$

$$x, y, S_1, S_2, S_3 \geq 0$$

Model yang sudah berbentuk kanonik ini dapat langsung diubah menjadi tabel simpleks pertama, dengan menempatkan variabel- variabel semu atau slack variable S_1, S_2 , dan S_3 sebagai variabel dasar. Langkah awal disajikan pada Tabel 2.3.

Tabel 2.3. Tabel Simpleks I

	C_j	40	30	0	0	0		
$\bar{C}_i$	$\bar{x}_i/x_j$	x	y	S_1	S_2	S_3	b_i	R_i
0	S_1	2	3	1	0	0	60	30
0	S_2	0	2	0	1	0	30	-
0	S_3	2	1	0	0	1	40	20
	z_j	0	0	0	0	0	0	
	$z_j - C_j$	-40	-30	0	0	0		

Pada tabel di atas terlihat masih terdapat nilai $z_j - c_j$ yang bernilai negatif, maka tabel belum optimal sehingga nilai $z_j - c_j$ negatif terkecil yang terdapat pada kolom variabel x merupakan variabel baru yang masuk. Nilai R_i terkecil adalah 20 yang terdapat pada variabel S_3 sehingga variabel S_3 keluar digantikan

oleh variabel x . Elemen pivotnya adalah 2 yang terletak pada perpotongan kolom x dan baris S_3 yang akan menjadi acuan Operasi Baris Elementer (OBE) untuk mengubah 2 menjadi 1. Elemen pivot yang sebelumnya bernilai 2 diubah menjadi bernilai 1 dengan cara perhitungan baris pivot tersebut dikalikan $\frac{1}{2}$ sedangkan untuk elemen selain elemen pivot diubah menjadi bernilai nol, baris kedua tetap karena nilainya adalah 0, sedangkan baris pertama diperoleh nilai 0 dengan cara mengurangi baris pertama dengan 2 kali lipat baris pivot (baris ketiga yang baru). Diperoleh tabel simpleks baru seperti pada tabel 2.4.

Tabel 2.4. Tabel Simpleks Iterasi I

	C_j	40	30	0	0	0		
$\bar{C}_i$	$\bar{x}_i/x_j$	x	y	S_1	S_2	S_3	b_i	R_i
0	S_1	0	2	1	0	-1	20	10
0	S_2	0	2	0	1	0	30	15
40	x	1	0,5	0	0	0,5	20	40
	z_j	40	20	0	0	20	800	
	$z_j - C_j$	0	-10	0	0	20		

Pada tabel 2.3 terdapat nilai $z_j - c_j$ yang bernilai negatif maka tabel belum optimal, sehingga nilai $z_j - c_j$ negatif terkecil yang terdapat pada kolom variabel y merupakan variabel baru yang masuk. Nilai R_i terkecil adalah 10 yang terdapat pada variabel S_1 sehingga variabel S_1 keluar digantikan oleh variabel y . Elemen pivotnya adalah 2 yang terletak pada perpotongan kolom y dan baris S_1 yang akan menjadi acuan Operasi Baris Elementer (OBE) untuk mengubah 2 menjadi 1. Elemen pivot yang sebelumnya bernilai 2 diubah menjadi bernilai 1 dengan cara

perhitungan baris pivot tersebut dikalikan $\frac{1}{2}$ sedangkan untuk elemen selain elemen pivot diubah menjadi bernilai nol, baris kedua dikurangkan dengan 2 kali lipat baris pivot (baris pertama yang baru), sedangkan baris ketiga diperoleh nilai 0 dengan cara mengurangkan baris ketiga dengan $\frac{1}{2}$ kali lipat baris pivot (baris pertama yang baru). Diperoleh tabel simpleks baru seperti pada tabel 2.5.

Tabel 2.5. Tabel Simpleks Iterasi II

	C_j	40	30	0	0	0		
$\bar{C}_i$	$\bar{x}_i/x_j$	x	y	S_1	S_2	S_3	b_i	R_i
30	Y	0	1	0,5	0	-0,5	10	
0	S_2	0	0	-1	1	0	10	
40	X	1	0	-0,25	0	0,75	15	
	z_j	40	30	5	0	15	900	
	$z_j - C_j$	0	0	5	0	15		

Dari tabel 2.5 tidak terdapat lagi nilai $z_j - c_j < 0$. Dengan demikian telah diperoleh penyelesaian optimal

$$\{x, y, S_1, S_2, S_3\} = \{15, 10, 0, 10, 0\}$$

dengan nilai maksimum

$$f\{15, 10, 0, 10, 0\}$$

$$\begin{aligned}
 z &= 40x + 30y + 0S_1 + 0S_2 + 0S_3 \\
 &= 40(15) + 30(10) + 0(0) + 0(10) + 0(0) \\
 &= 900
 \end{aligned}$$

3. Solusi *Linear Programming*

Solusi dari suatu masalah *pemrograman linear* merupakan hasil akhir dari suatu pemecahan masalah tersebut. Terdapat dua jenis solusi pada *pemrograman*

linear yaitu solusi layak (*feasible solution*) dan solusi tidak layak (*no feasible solution*). Solusi layak merupakan penyelesaian yang memenuhi semua kendala yang ada pada masalah *pemrograman linear*, sedangkan solusi tidak layak merupakan penyelesaian yang tidak memenuhi salah satu atau beberapa kendala yang ada.

4. Analisis Sensitivitas

Analisis sensitivitas merupakan suatu analisa yang berkaitan dengan perubahan parameter pada model pemrograman linear sampai sejauh mana, yaitu pada koefisien tujuan dan nilai ruas kanan kendala boleh berubah tanpa mempengaruhi penyelesaian optimal. Analisa ini dikembangkan dari penyelesaian optimal suatu permasalahan.

Perubahan-perubahan yang dapat mempengaruhi penyelesaian optimal, yaitu (B. Susanta, 1994):

1. perubahan suku tetap (ruas kanan) (b_i),
2. perubahan koefisien ongkos (c_j),
3. perubahan koefisien teknis (a_{ij}),
4. penambahan kendala baru,
5. penambahan perubah/variabel baru.

Pada kasus penelitian tentang menu diet Diabetes Mellitus ini analisis sensitivitas yang digunakan adalah pada perubahan suku tetap (ruas kanan).

Penyelesaian suatu masalah pemrograman linear menggunakan metode simpleks diketahui bahwa suatu penyelesaian layak basis akan menjadi penyelesaian optimal jika $z_j - c_j \leq 0$ untuk semua j (pola minimum) atau dengan

nilai $z_j - c_j \geq 0$ untuk semua j (pola maksimum). Nilai $z_j - c_j$ ini tidak berhubungan dengan b_i tetapi bergantung pada basis a_{ij} dan c_j . Apabila suatu penyelesaian untuk permasalahan tertentu telah diperoleh penyelesaian optimal dan nilai b_i diubah menjadi $b_i + \Delta b_i$, maka perubahan tersebut mempunyai kemungkinan mempengaruhi nilai variabel basis dan penyelesaian optimal. Kemudian jika perubahan basis baru tetap layak untuk $b_i + \Delta b_i$ maka penyelesaian layak basis optimal untuk soal asli misalkan $\bar{X}$ akan tetap optimal untuk masalah yang baru pada perubahan basis baru, maka

$$\bar{X}^* = (\bar{x}_1^*, \bar{x}_2^*, \dots, \bar{x}_m^*) \text{ harus memenuhi } \bar{X}^* = D^{-1}(B + \Delta B) \geq 0$$

Jika penyelesaian layak basis soal asli adalah

$$\bar{X}^* = (\bar{x}_1^*, \bar{x}_2^*, \dots, \bar{x}_m^*) = D^{-1}B$$

maka

$$\bar{x}_i^* = \bar{x}_i + \sum_{j=1}^m d_{ij} \Delta b_j \geq 0 \quad (2.8)$$

untuk $i = 1, 2, \dots, m$

dengan $D^{-1} = d_{ij}$

perubahan nilai variabel basis ke- i (Δx_i) yang bersesuaian dengan perubahan pada b_i adalah

$$\bar{X}^* - \bar{X} = \Delta \bar{X} = D^{-1} \Delta B$$

atau

$$\Delta x_i = \sum_{j=1}^m d_{ij} \Delta b_j \quad (2.9)$$

untuk $i = 1, 2, \dots, m$

untuk nilai pada fungsi tujuan yang diakibatkan oleh Δb_i dapat diperoleh sebagai berikut:

$$\Delta f = \bar{C}\Delta\bar{X} = \bar{C}D^{-1}\Delta B \quad (2.10)$$

Misalkan bahwa Δb_i menyebabkan tidak terpenuhinya $\bar{X}^* - \bar{X} = \Delta\bar{X} = D^{-1}\Delta B \geq 0$ sehingga mengakibatkan $x_i + \Delta x_i < 0$, maka perlu dilakukan perhitungan penyelesaian ulang dan rumus-rumus sebelumnya tidak berlaku (B. Susanta, 1994).

Contoh 2.2: Analisis sensitivitas pada perubahan nilai ruas kanan

Soal asli masalah pemrograman linear.

Meminimumkan: $-45x_1 - 100x_2 - 30x_3 - 50x_4$

dengan kendala, $7x_1 + 10x_2 + 4x_3 + 9x_4 \leq 1200$

$$3x_1 + 40x_2 + x_3 + x_4 \leq 800$$

$$x_i \geq 0, i = 1, 2, 3, 4$$

Sesudah penyelesaian optimal ditemukan, tambahkan $\Delta b_1 = 300$ dan $\Delta b_2 = 200$ kepada suku tetap dan selidiki pengaruhnya terhadap penyelesaian optimal asli.

Penyelesaian:

Model kanonik masalah pemrograman linear

Meminimumkan: $-45x_1 - 100x_2 - 30x_3 - 50x_4 + S_1 + S_2$

dengan kendala, $7x_1 + 10x_2 + 4x_3 + 9x_4 + S_1 = 1200$

$$3x_1 + 40x_2 + x_3 + x_4 + S_2 = 800$$

$$x_i, S_1, S_2 \geq 0, i = 1, 2, 3, 4$$

Kemudian diselesaikan dengan menggunakan metode simpleks, sehingga diperoleh perhitungan dengan menggunakan tabel simpleks seperti pada tabel 2.6.

Tabel 2.6. Tabel Simpleks Contoh 2.2

I		c_j	-45	-100	-30	-50	0	0		
	c_i	x_i/x_j	x_1	x_2	x_3	x_4	S_1	S_2	b_i	R_i
	0	S_1	7	10	4	9	1	0	1200	120
	0	S_2	3	40	1	1	0	1	800	20
		z_j	0	0	0	0	0	0	0	
II		$z_j - c_j$	45	100	30	50	0	0	0	
	0	S_1	25/4	0	15/4	35/4	1	-1/4	1000	800/7
	-100	x_2	3/40	1	1/40	1/40	0	1/40	20	800
		z_j	-15/2	-100	-5/2	-5/2	0	-5/2	-2000	
		$z_j - c_j$	75/2	0	55/2	75/2	0	-5/2	-2000	
III	-50	x_4	5/7	0	3/7	1	4/35	-1/35	800/7	800/3
	-100	x_2	2/35	1	1/70	0	-1/350	9/350	120/7	1200
		z_j	-290/7	-100	-160/7	-50	-38/7	-8/7	-4000	
		$z_j - c_j$	25/7	0	50/7	0	-38/7	-8/7	-4000	
	-30	x_3	5/3	0	1	7/3	4/15	-1/15	800/3	
IV	-100	x_2	1/30	1	0	-1/30	-1/150	2/75	40/3	
		z_j	-160/3	-100	-30	-200/3	-22/3	-2/3	-28000/3	
		$z_j - c_j$	-25/3	0	0	-50/3	-22/3	-2/3	-28000/3	

Berdasarkan tabel IV pada tabel 2.6 di atas telah diperoleh penyelesaian optimalnya, yaitu: $(x_1, x_2, x_3, x_4, S_1, S_2) = (0, \frac{40}{3}, \frac{800}{3}, 0, 0, 0)$ dengan $f_{min} = -\frac{28000}{3}$ dalam peubah keputusan penyelesaian optimalnya adalah $(x_1, x_2, x_3, x_4) = (0, \frac{40}{3}, \frac{800}{3}, 0)$ sehingga untuk penyelesaian optimal tersebut diperoleh:

$$\bar{X} = \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \end{pmatrix} = \begin{pmatrix} x_3 \\ x_2 \end{pmatrix} = \begin{pmatrix} 800/3 \\ 40/3 \end{pmatrix} \quad \bar{C} = (-30, -100)$$

$$D = \begin{pmatrix} 4 & 10 \\ 1 & 40 \end{pmatrix} \quad D^{-1} = \begin{pmatrix} 4/15 & -1/15 \\ -1/150 & 2/75 \end{pmatrix}$$

Penyelesaian soal terubah

Tambahkan $\Delta b_1 = 300$ dan $\Delta b_2 = 200$, maka dengan menggunakan rumus (2.12).

$$\begin{aligned}
\bar{x}_1^* &= \bar{x}_1 + \sum_{j=1}^2 d_{1j} \Delta b_j = x_1 + d_{11} \Delta b_1 + d_{12} \Delta b_2 \\
&= \frac{800}{3} + \frac{4}{15} 300 - \frac{1}{15} 200 \\
&= \frac{1000}{3}
\end{aligned}$$

$$\begin{aligned}
\bar{x}_2^* &= \bar{x}_2 + \sum_{j=1}^2 d_{2j} \Delta b_j = x_2 + d_{21} \Delta b_1 + d_{22} \Delta b_2 \\
&= \frac{40}{3} - \frac{1}{150} 300 + \frac{2}{75} 200 \\
&= \frac{50}{3}
\end{aligned}$$

Diperoleh bahwa x_1^* dan x_2^* bernilai positif, maka variabel basis optimal soal asli masih menjadi variabel basis optimal soal baru. Nilai-nilai variabel basis baru akan menjadi $x_1^* = \frac{1000}{3}$ dan $x_2^* = \frac{50}{3}$. Dengan demikian berarti penyelesaian optimal baru menjadi $(0, \frac{50}{3}, \frac{1000}{3}, 0, 0, 0)$ dengan

$$\begin{aligned}
f_{min}^* &= f_{min} + \Delta f \\
&= f_{min} + \bar{C} D^{-1} \Delta B \\
&= -\frac{28000}{3} + (-30 \quad -100) \begin{bmatrix} 4/15 & -1/15 \\ -1/150 & 2/75 \end{bmatrix} \begin{bmatrix} 300 \\ 200 \end{bmatrix} \\
&= -\frac{28000}{3} + (-30 \quad -100) \begin{bmatrix} \frac{200}{3} \\ \frac{10}{3} \\ \frac{10}{3} \end{bmatrix} \\
&= -\frac{35000}{3}
\end{aligned}$$

D. Goal Programming

Goal Programming merupakan salah satu model matematis yang dapat digunakan sebagai dasar dalam pengambilan keputusan untuk menganalisis dan membuat solusi persoalan yang melibatkan banyak sasaran sehingga diperoleh solusi yang optimal. Pendekatan dasar *goal programming* adalah untuk menetapkan suatu tujuan yang dinyatakan dengan angka tertentu untuk setiap tujuan, merumuskan suatu fungsi tujuan untuk setiap tujuan, kemudian mencari penyelesaian yang meminimumkan jumlah penyimpangan-penyimpangan pada fungsi tujuan. Pada model *goal programming* ini berusaha untuk meminimumkan deviasi diantara berbagai tujuan atau sasaran yang telah ditentukan sebagai targetnya, maksudnya adalah nilai dari ruas kiri persamaan kendala sebisa mungkin mendekati nilai dari ruas kanannya.

Model *goal programming* merupakan perluasan dari model pemrograman linear yang dikembangkan oleh A. Charnes dan W. M. Cooper pada tahun 1955 dan mulai dipopulerkan pada tahun 1961 sehingga asumsi-asumsi, notasi, formulasi model matematis, prosedur perumusan model, dan penyelesaiannya tidak berbeda. Perbedaannya hanya terletak pada kehadiran sepasang variabel deviasional yang akan muncul di fungsi tujuan dan fungsi-fungsi kendala (Siswanto, 2007).

Beberapa asumsi dasar yang diperlukan dalam *goal programming* adalah sebagai berikut:

a. Linieritas

Asumsi ini menunjukkan perbandingan antara input yang satu dengan input yang lain atau suatu input dengan output besarnya tetap dan terlepas pada tingkat produksi. Hubungannya bersifat linear.

b. Proporsionalitas

Asumsi ini menyatakan bahwa jika peubah pengambil keputusan berubah, maka dampak perubahannya akan menyebar dalam proporsi yang sebanding dengan fungsi tujuan dan juga fungsi kendalanya. Jadi tidak berlaku hukum kenaikan hasil yang semakin berkurang.

c. Aditivitas

Asumsi ini menyatakan nilai parameter suatu kriteria optimisasi merupakan jumlah dari nilai individu-individu. Dampak total terhadap kendala ke-i merupakan jumlah dampak individu terhadap peubah pengambilan keputusan.

d. Disibilitas

Asumsi ini menyatakan bahwa peubah pengambilan keputusan jika diperlukan dapat dibagi ke dalam pecahan-pecahan.

e. Deterministik

Asumsi ini menghendaki agar semua parameter tetap dan diketahui atau ditentukan secara pasti.

Ada beberapa istilah yang dipergunakan dalam *goal programming*, yaitu (Siswanto, 2007):

1. variabel keputusan (*decision variables*), adalah seperangkat variabel yang tidak diketahui yang berada di bawah kontrol pengambilan keputusan, yang

berpengaruh terhadap solusi permasalahan dan keputusan yang akan diambil. Biasanya dilambangkan dengan X_j ($j=1, 2, 3, \dots, n$),

2. nilai sisi kanan (*right hand sides values*), merupakan nilai-nilai yang biasanya menunjukkan ketersediaan sumber daya (dilambangkan dengan b_i) yang akan ditentukan kekurangan atau penggunaannya,
3. koefisien teknologi (*technology coefficient*), merupakan nilai-nilai numerik yang dilambangkan dengan yang akan dikombinasikan dengan variabel keputusan, dimana akan menunjukkan penggunaan terhadap pemenuhan nilai kanan,
4. variabel deviasional (*penyimpangan*), adalah variabel yang menunjukkan kemungkinan penyimpangan-penyimpangan negatif dan positif dari nilai sisi kanan fungsi tujuan. Variabel penyimpangan negatif berfungsi untuk menampung penyimpangan yang berada di bawah sasaran yang dikehendaki, sedangkan variabel penyimpangan positif berfungsi untuk menampung penyimpangan yang berada di atas sasaran. Dalam *Goal Programming* dilambangkan dengan d_i^- penyimpangan negatif dan d_i^+ untuk penyimpangan positif dari nilai sisi kanan tujuan,
5. fungsi tujuan, adalah fungsi matematis dari variabel-variabel keputusan yang menunjukkan hubungan dengan nilai sisi kanannya, fungsi tujuan dalam *Goal Programming* adalah meminimumkan variabel devisional,
6. fungsi pencapaian, adalah fungsi matematis dari variabel-variabel simpangan yang menyatakan kombinasi sebuah objektif,

7. fungsi tujuan mutlak (non negatif), merupakan tujuan yang tidak boleh dilanggar dengan pengertian mempunyai penyimpangan positif dan atau negatif bernilai nol. Prioritas pencapaian dari fungsi tujuan ini berada pada urutan pertama, solusi yang dapat dihasilkan adalah terpenuhi atau tidak terpenuhi,
8. prioritas, adalah suatu sistem urutan dari banyaknya tujuan pada model yang memungkinkan tujuan-tujuan tersebut disusun secara ordinal dalam *goal programming*. Sistem urutan tersebut menempatkan sasaran-sasaran tersebut dalam susunan dengan seri,
9. pembobotan, merupakan timbangan matematis yang dinyatakan dengan angka ordinal yang digunakan untuk membedakan variabel simpangan i dalam suatu tingkat prioritas k .

Dalam *goal programming* terdapat tiga unsur utama yaitu fungsi tujuan, kendala tujuan, dan kendala non negatif.

1. Fungsi Tujuan

Fungsi tujuan dalam *goal programming* pada umumnya adalah masalah minimisasi, karena dalam fungsi tujuan terdapat variabel simpangan yang harus diminimumkan. Fungsi tujuan dalam *goal programming* adalah meminimumkan total penyimpangan tujuan yang ingin dicapai.

2. Kendala Non Negatif

Kendala non negatif dalam *goal programming* adalah semua variabel-variabel bernilai positif atau samadengan nol. Jadi variabel keputusan dan

variabel deviasi dalam masalah *goal programming* bernilai positif atau samadengan nol. Pernyataan non negatif dilambangkan $x_j, d_i^-, d_i^+ \geq 0$.

3. Kendala Tujuan

Menurut Rio Armindo (2006), dalam *goal programming* ada enam jenis kendala tujuan yang berlainan. Tujuan dari setiap jenis kendala itu ditentukan oleh hubungannya dengan fungsi tujuan. Berikut adalah enam jenis kendala tersebut:

Tabel 2.7. Tabel Jenis Kendala dalam *Goal Programming*

No	Kendala Tujuan	Variabel Deviasi dalam Fungsi Tujuan	Kemungkinan Simpangan	Penggunaan Nilai RHS yang Diinginkan
1	$C_{ij}x_{ij} + d_i^- = b_i$	d_i^-	Negatif	$= b_i$
2	$C_{ij}x_{ij} - d_i^+ = b_i$	d_i^+	Positif	$= b_i$
3	$C_{ij}x_{ij} + d_i^- - d_i^+ = b_i$	d_i^- dan d_i^+	Negatif atau Positif	b_i atau lebih
4	$C_{ij}x_{ij} + d_i^- - d_i^+ = b_i$	d_i^- dan d_i^+	Negatif atau Positif	b_i atau kurang
5	$C_{ij}x_{ij} + d_i^- - d_i^+ = b_i$	d_i^- dan d_i^+	Negatif atau Positif	$= b_i$
6	$C_{ij}x_{ij} - d_i^+ = b_i$	d_i^+ (artifisial)	Tidak ada	$= b_i$

Pada tabel di atas terlihat bahwa setiap kendala tujuan memiliki satu atau dua variabel simpangan yang keduanya atau salah satunya ditempatkan pada fungsi tujuan.

1. Formulasi Model Umum *Goal Programming*

Misal:

C_{ij} : koefisien teknologi kendala tujuan yang berhubungan dengan tujuan peubah pengambilan keputusan (x_j),

x_j : peubah pengambilan keputusan atau kegiatan yang dinamakan sebagai sub tujuan,

d_m^+ : jumlah unit deviasi yang berlebihan (+) terhadap tujuan (b_m),

d_m^- : jumlah unit deviasi yang kekurangan (-) terhadap tujuan (b_m),

b_i : tujuan yang ingin dicapai,

P_k : faktor prioritas pada tujuan ke- k .

Model umum *goal programming* tanpa faktor prioritas pada strukturnya adalah

meminimumkan:

$$Z = \sum_{i=1}^m (d_i^+ + d_i^-) \quad (2.11)$$

dengan kendala,

$$\begin{aligned} C_{11}x_1 + C_{12}x_2 + \cdots + C_{1n}x_n + d_1^- - d_1^+ &= b_1 \\ C_{21}x_1 + C_{22}x_2 + \cdots + C_{2n}x_n + d_2^- - d_2^+ &= b_2 \\ &\vdots \\ C_{m1}x_1 + C_{m2}x_2 + \cdots + C_{mn}x_n + d_m^- - d_m^+ &= b_m \end{aligned} \quad (2.12)$$

kendala non negatif:

$$x_j, \quad d_i^-, \quad d_i^+ \geq 0$$

untuk $i = 1, 2, \dots, m$ dan $j = 1, 2, \dots, n$

Model untuk permasalahan tujuan ganda dengan struktur timbangan prioritas (*pre-emptive weights*) adalah sebagai berikut:

Meminimumkan:

$$Z = P_1 d_i^- + \dots + P_l d_i^- + P_{l+1} d_i^+ + \dots + P_k d_i^+ \quad (2.13)$$

Dengan kendala,

$$\begin{aligned} C_{11}x_1 + C_{12}x_2 + \dots + C_{1n}x_n + d_1^- - d_1^+ &= b_1 \\ C_{21}x_1 + C_{22}x_2 + \dots + C_{2n}x_n + d_2^- - d_2^+ &= b_2 \\ &\vdots \\ C_{m1}x_1 + C_{m2}x_2 + \dots + C_{mn}x_n + d_m^- - d_m^+ &= b_m \end{aligned} \quad (2.14)$$

Kendala non negatif:

$$\begin{aligned} x_j, \quad d_i^-, \quad d_i^+ &\geq 0 \\ \text{untuk } i &= 1, 2, \dots, m \text{ dan } j = 1, 2, \dots, n \end{aligned}$$

Berdasarkan perumusan model *goal programming*, pencapaian tingkat sasaran atau target dilakukan dengan cara meminimumkan peubah deviasi. Ada dua tipe program sasaran, yaitu program sasaran yang setiap sasarnya memiliki prioritas yang sama dan program sasaran yang mengurutkan sasarnya menurut tingkat prioritas dari sasarnya. Untuk sasaran yang diurutkan berdasarkan tingkat prioritasnya diberikan faktor pembobot. Faktor pembobot adalah suatu nilai numerik yang tidak berdimensi dan digunakan untuk menunjukan tingkat prioritas relatif dari suatu sasaran. Besar kecilnya nilai faktor pembobot dari setiap sasaran diperoleh dari hasil manipulasi pendapat para ahli atau pengambil keputusan (Rio Armindo, 2006).

Jika faktor pembobot fungsi sasaran prioritas ke- i dilambangkan dengan W_i , maka secara matematis dapat bersifat

$$0 < W_i < 1 \text{ dan}$$

$$\sum_{i=1}^k W_i = 1$$

apabila terdapat pernyataan $W_c > W_y$ menunjukkan bahwa sasaran ke-c lebih penting daripada sasaran ke-y dan jika $W_c = W_y$ maka sasaran ke-c dan sasaran ke-y mempunyai prioritas yang sama.

2. Penyelesaian model *Goal Programming*

Langkah-langkah perumusan masalah pada *goal programming* adalah:

- a. Penentuan variabel keputusan, merupakan dasar dalam pembuatan model keputusan untuk mendapatkan solusi yang dicari. Semakin tepat penentuan variabel keputusan akan mempermudah pengambilan keputusan yang dicari.
- b. Penentuan fungsi tujuan, yaitu tujuan-tujuan yang ingin dicapai oleh perusahaan.
- c. Perumusan fungsi tujuan, dimana setiap sasaran pada sisi kirinya ditambahkan dengan variabel simpangan, baik simpangan positif maupun simpangan negatif. Dengan ditambahkan variabel simpangan, maka bentuk dari fungsi sasaran menjadi

$$f_i(x_i) + d_i^- - d_i^+ = b_i$$

- d. Penentuan prioritas utama. Pada langkah ini dibuat urutan dari sasaran-sasaran. Penentuan sasaran ini tergantung pada hal-hal berikut :
 - i. keinginan dari pengambil keputusan,
 - ii. keterbatasan sumber-sumber yang ada.
- e. Penentuan pembobotan. Pada tahap ini merupakan kunci dalam menentukan

urutan dalam suatu tujuan dibandingkan dengan tujuan yang lain.

- f. Penentuan fungsi pencapaian. Dalam hal ini, yang menjadi kuncinya adalah memilih variabel simpangan yang benar untuk dimasukkan dalam fungsi pencapaian. Dalam memformulasikan fungsi pencapaian adalah menggabungkan setiap tujuan yang berbentuk minimisasi *variabel* penyimpangan sesuai prioritasnya.
- g. Penyelesaian model *Goal Programming*.

3. Metode Pemecahan Masalah *Goal Programming*

Algoritma simpleks dapat juga digunakan untuk menyelesaikan masalah *goal programming* dengan menggunakan variabel keputusan lebih dari dua. Langkah-langkah penyelesaian *goal programming* dengan metode algoritma simpleks tersebut adalah:

- a. Membentuk tabel simpleks awal
- b. Pilih kolom kunci dimana $C_j - Z_j$ memiliki negatif terbesar. Kolom kunci ini disebut kolom pivot.
- c. Pilih baris kunci yang berpedoman pada b_i/a_{ij} dengan rasio terkecil dimana b_i adalah nilai sisi kanan dari setiap persamaan. Baris kunci ini disebut baris pivot.
- d. Mencari sistem kanonik yaitu sistem dimana nilai elemen pivot bernilai 1 dan elemen lain bernilai nol dengan cara mengalikan baris pivot dengan -1 lalu menambahkannya dengan semua elemen di baris pertama. Dengan demikian, diperoleh table simpleks iterasi I.
- e. Pemeriksaan optimasi, yaitu melihat apakah solusi layak atau tidak. Solusi

dikatakan layak bila variabel adalah positif atau nol.

Misal:

$\bar{x}_i$: variabel basis,

$\bar{c}_i$: koefisien dari $\bar{x}_i$,

z_j : $\sum_{i=1}^m \bar{c}_i a_{ij}$,

z : $\sum_{i=1}^m \bar{c}_i b_i$ nilai fungsi tujuan,

R_i : rasio antara b_i dan a_{ik} jika x_k terpilih menjadi variabel basis.

Tabel awal dari model *goal programming* dapat dilihat pada tabel 2.8.

Tabel 2.8. Tabel Awal *Goal Programming*

	c_j	0	0	...	0	$\omega_1 P_1$	$\omega_1 P_1$	...	$\omega_m P_m$	$\omega_m P_m$		
$\bar{c}_i$	$\bar{x}_i/x_j$	x_1	x_2	...	x_m	d_1^-	d_1^+	...	d_m^-	d_m^+	b_i	R_i
$\omega_1 P_1$	d_1^-	a_{11}	a_{12}	...	a_{1m}	1	-1	...	0	0	b_1	R_1
$\omega_1 P_1$	d_2^-	a_{21}	a_{22}	...	a_{2m}	0	0	...	0	0	b_2	R_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$			$\vdots$	$\vdots$	
$\omega_m P_m$	d_m^-	a_{m1}	a_{m2}	...	a_{mm}	0	0	...	1	-1	b_m	R_m
	z_j	...	...	...	...	...	...	...	...	...	Z	
	$z_j - c_j$	...	...	...	...	...	...	...	...	...	Z	

Setelah model *goal programming* tersebut diselesaikan dengan metode simpleks maka diperoleh nilai dari variabel $x_1, \dots, x_n$ yang mengoptimalkan fungsi tujuan. Selain itu, juga diperoleh nilai variabel-variabel simpangan yang diartikan sebagai besarnya penyimpangan dari tujuan, tetapi dijamin simpangan yang diperoleh tetap paling minimal. Untuk bab selanjutnya d_i^- akan disimbolkan dengan DA_i dan d_i^+ disimbolkan dengan DB_i agar memudahkan penulisan pada LINGO.

E. LINGO

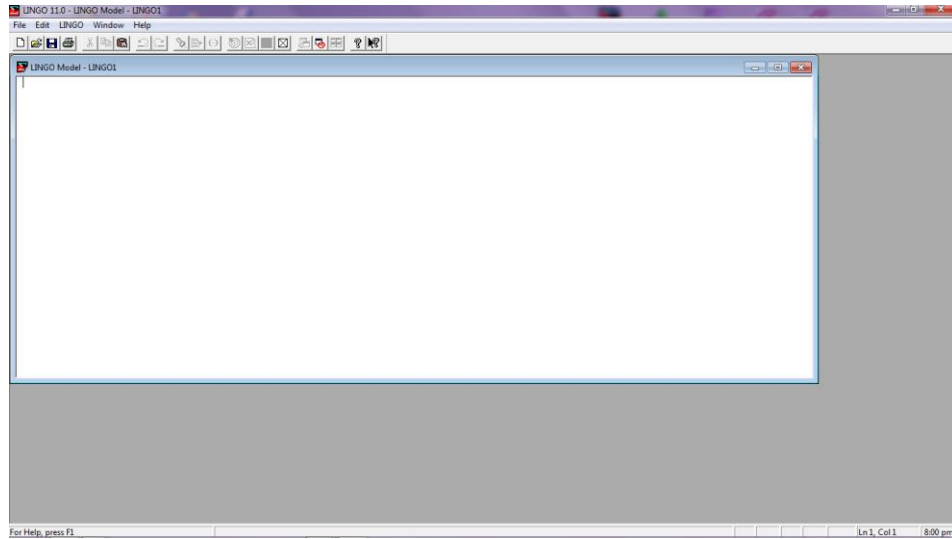
LINGO adalah suatu program komputer yang dapat digunakan untuk aplikasi *pemrograman linear*. Aplikasi *pemrograman linear* adalah suatu pemodelan matematika yang digunakan untuk mendapatkan penyelesaian optimal dengan kendala yang ada.

LINGO adalah perangkat lunak yang dapat digunakan untuk menyelesaikan masalah-masalah pemrograman linear, non-linear dan integer. LINGO sudah banyak digunakan oleh perusahaan-perusahaan untuk membantu membuat perencanaan produksi yang bertujuan untuk mendapatkan keuntungan yang optimum dan biaya yang minimum.

LINGO telah menjadi *software* optimasi selama lebih dari 20 tahun. Sistem LINGO telah menjadi pilihan utama dalam penyelesaian yang cepat dan mudah, terutama dalam masalah optimasi persamaan matematika. Selain itu struktur bahasa yang digunakan dalam memformulasikan masalahnya lebih sederhana, yaitu persamaan linier.

Untuk menggunakan *software* LINGO ada beberapa tahapan yang perlu dilakukan, yaitu:

1. merumuskan masalah dalam kerangka program linier. Dapat dituliskan pada *software* Ms. Word,
2. menuliskan dalam persamaan matematika,
3. merumuskan rumusan ke dalam LINGO dan mengeksekusinya,
4. interpretasi keluaran LINGO.



Gambar 2.1. Tampilan Awal Program LINGO 11.0

Cara untuk menginput *script* dilakukan seperti mengetik tulisan biasa, untuk mengeksekusi perintah dilakukan dengan cara klik LINGO pada sub menu kemudian pilih Solve pada sub menu tersebut maka hasilnya akan diperoleh. Sedangkan untuk melakukan uji sensitivitas, perintah yang dilakukan adalah dengan cara pilih Range pada sub menu LINGO.