BAB II

LANDASAN TEORI

A. Pemrograman Linear

Pemrograman linear (PL) ialah salah satu teknik dari riset operasi untuk memecahkan persoalan optimasi (maksimum atau minimum) dengan menggunakan persamaan dan pertidaksamaan linear dalam rangka untuk mencari pemecahan yang optimal dengan memperhatikan pembatasan-pembatasan yang ada (Johannes Supranto, 1991 : 43). Fungsi linear yang hendak dicari nilai optimum berbentuk sebuah persamaan yang disebut fungsi tujuan. Fungsi linear yang harus terpenuhi dalam optimisasi fungsi tujuan, dapat berbentuk persamaan maupun pertidaksamaan yang disebut fungsi kendala (Dumairy, 2012 : 344). Sebuah fungsi f adalah suatu aturan padanan yang menghubungkan setiap obyek x dalam satu himpunan, yang disebut daerah asal, dengan sebuah nilai tunggal $f(x)$ dari suatu himpunan kedua. Himpunan nilai yang diperoleh secara demikian disebut daerah hasil fungsi (Varberg & Purcell, 2011 : 57).

Siswanto (2007 : 26) menyebutkan definisi pemrograman linear yaitu sebagai metode metematis yang berbentuk linear untuk menentukan suatu penyelesaian optimal dengan cara memaksimumkan atau meminimumkan fungsi tujuan terhadap suatu susunan kendala. Secara keseluruhan, berdasarkan definisi maka tujuan pemrograman linear adalah memecahkan persoalan memaksimumkan atau meminimumkan untuk mendapatkan penyelesaian yang optimal.

Terdapat tiga unsur utama yang membangun suatu program linear yaitu (Siswanto, 2007 : 26):

1. Variabel keputusan

   Variabel keputusan adalah variabel yang mempengaruhi nilai tujuan yang hendak dicapai. Pada proses pembentukan suatu model, menentukan variabel keputusan merupakan langkah pertama sebelum menentukan fungsi tujuan dan fungsi kendala.

2. Fungsi tujuan

   Fungsi tujuan pada model pemrograman linear haruslah berbentuk linear. Selanjutnya, fungsi tujuan tersebut dimaksimalkan atau diminimalkan terhadap fungsi-fungsi kendala yang ada.

3. Fungsi kendala

   Fungsi kendala adalah suatu kendala yang dapat dikatakan sebagai suatu pembatas terhadap variabel-variabel keputusan yang dibuat. Fungsi kendala untuk model pemrograman linear juga harus berupa fungsi linear.

4. Fungsi non-negative

   Fungsi yang menyatakan bahwa setiap variabel yang terdapat di dalam model pemrograman linear tidak boleh negatif. Secara matematis ditulis sebagai $x_1, x_2, \cdots, x_j \geq 0$.

1. Asumsi-asumsi Dasar Pemrograman Linear

Asumsi-asumsi dasar pemrograman linear diuraikan agar penggunaan teknik pemrograman linear ini dapat memuaskan untuk berbagai masalah. Asumsi-asumsi dalam pemrograman linear akan dijelaskan secara implisit pada bentuk umum model pemrograman linear. Adapun asumsi-asumsi dasar pemrograman linear sebagai berikut (Pangestu Subagyo, 1995:14-15):

a. *Proportionality* (kesebandingan)

Asumsi ini mempunyai arti bahwa naik turunnya nilai fungsi tujuan dan penggunaan sumber atau fasilitas yang tersedia akan berubah secara sebanding (proportional) dengan perubahan tingkat kegiatan.

b. *Additivity* (penambahan)

Asumsi ini mempunyai arti bahwa nilai fungsi tujuan tiap kegiatan tidak saling mempengaruhi, atau dalam pemrograman linear dianggap bahwa kenaikan dari nilai tujuan yang diakibatkan oleh kenaikan suatu kegiatan dapat ditambahkan tanpa mempengaruhi bagian nilai tujuan yang diperoleh dari kegiatan lain.

c. *Divisibility* (dapat dibagi)

Asumsi ini menyatakan bahwa keluaran (*output*) yang dihasilkan oleh setiap kegiatan dapat berupa bilangan pecahan. Demikian pula dengan nilai tujuan yang dihasilkan.

d. *Deterministic* (kepastian)

Asumsi ini menyatakan bahwa semua parameter yang terdapat dalam model pemrograman linear (a_{ij}, b_i, c_j) dapat diperkirakan dengan pasti.

2. Bentuk Umum Model Pemrograman Linear

Masalah pemrograman linear adalah masalah optimisasi bersyarat yakni pencarian nilai maksimum atau pencarian nilai minimum sesuatu fungsi tujuan berkenaan dengan keterbatasan-keterbatasan atau kendala yang harus dipenuhi.

Masalah-masalah tersebut secara umum dapat dirumuskan sebagai berikut (Johannes Supranto, 1991 : 44):

Fungsi tujuan memaksimumkan dinotasikan dengan Z dan relasi dalam kendala berbentuk ($\leq$) sehingga bentuknya dapat dilihat pada persamaan (2.1).

Maksimumkan fungsi tujuan

$$Z = c_1x_1 + c_2x_2 + \cdots + c_jx_j \tag{2.1}$$

terhadap kendala-kendala

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1j}x_j &\leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2j}x_j &\leq b_2 \\ \vdots \qquad \vdots \qquad\quad \vdots \quad &\quad \vdots \\ a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{ij}x_j &\leq b_i \end{aligned} \tag{2.2}$$

Kendala non negatif

$$x_j \geq 0 \ (j = 1, 2, \cdots, n)$$

Fungsi tujuan meminimumkan dinotasikan dengan W dan relasi dalam kendala berbentuk ($\geq$) sehingga menjadi:

Meminimumkan fungsi tujuan

$$W = c_1x_1 + c_2x_2 + \cdots + c_jx_j$$

terhadap kendala-kendala

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1j}x_j &\geq b_1 \\ a_{21}x_1 + a_{22}x_1 + \cdots + a_{2j}x_j &\geq b_2 \\ \vdots \qquad \vdots \qquad\quad \vdots \quad &\quad \vdots \\ a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{ij}x_j &\geq b_i \end{aligned}$$

Kendala non negatif

$$x_j \geq 0 \ (j = 1, 2, \cdots, n)$$

dengan:

x_j : variabel keputusan ke$-j$/ banyaknya produk ke$-j$ ($j = 1, 2, \cdots, n$)

b_i : suku tetap / bahan mentah jenis ke$-i$ yang tersedia ($i = 1, 2, \cdots, m$)

a_{ij} : koefisien kendala/ bahan mentah ke$-i$ yang digunakan untuk memproduksi satu unit produk j

c_j : koefisien ongkos /harga jual satu unit j

Secara keseluruhan model matematis yang digunakan untuk menyelesaikan suatu permasalahan pemrograman linear dapat disusun ke dalam bentuk tabel, seperti tampak pada Tabel 2.1

Tabel 2.1 Bentuk umum tabel model pemrograman linear

	c_j	c_1	c_2	$\cdots$	c_n	
$\bar{c}_i$	$\bar{x}_i / x_j$	x_1	x_2	$\cdots$	x_n	b_i
$\bar{c}_1$	$\bar{x}_1$	a_{11}	a_{12}	$\cdots$	a_{1j}	b_1
$\bar{c}_2$ $\vdots$	$\bar{x}_2$ $\vdots$	a_{21} $\vdots$	a_{22}	$\cdots$	a_{2j}	b_2
$\bar{c}_m$	$\bar{x}_m$	a_{i1}	a_{i2}	$\cdots$	a_{ij}	b_m
	z_j	z_1	z_2	$\cdots$	z_n	z
	$z_j - c_j$	$z_1 - c_1$	$z_2 - c_1$	$\cdots$	$z_n - c_n$	

(Sumber: Program Linear, B.Susanta, 1994)

Keterangan:

x_j : variabel keputusan ke$-j$/ banyaknya produk ke$-j$ $(j = 1, 2, \cdots, n)$

$\bar{x}_i$: variabel basis ke$-i$ $(i = 1, 2, \cdots, m)$

c_j : koefisien ongkos / harga jual satu unit j

$\bar{c}_i$: koefisien ongkos milik variabel basis ke$-i$ $(i = 1, 2, \cdots, m)$

a_{ij} : koefisien kendala/ bahan mentah ke$-i$ yang digunakan untuk memproduksi satu unit produk j

b_i : suku tetap / bahan mentah jenis ke$-i$ yang tersedia

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

z : $\sum_{i=1}^{m} \bar{c}_i b_i$ (jumlah hasil kali $\bar{c}_i$ dengan kolom b_i)

$z_j - c_j$: selisih z_j dengan c_j

3. Penyelesaian Masalah Pemrograman Linear

Dalam penyelesaian model pemrograman linear, dikenal metode simpleks. Metode simpleks adalah suatu metode yang secara sistematis dimulai dari suatu pemecahan dasar ke pemecahan dasar yang layak lainnya dilakukan berulang-ulang (dengan jumlah ulangan yang terbatas) sehingga akhirnya tercapai suatu pemecahan dasar yang optimal. Setiap langkah menghasilkan suatu nilai dan fungsi tujuan yang selalu lebih besar (lebih kecil) atau sama dari langkah-langkah sebelumnya (Johannes Supranto, 1991 : 73). Metode simpleks lebih efisien serta dilengkapi dengan suatu test kriteria yang dapat memberitahukan kapan hitungan harus dihentikan dan kapan harus dilanjutkan sampai diperoleh suatu penyelesaian yang optimal. Pada umumnya dipergunakan tabel-tabel, dari tabel pertama yang memberikan pemecahan dasar permulaan yang fisibel sampai pada pemecahan terakhir yang memberikan solusi optimal (Johannes Supranto, 1991 : 75).

Pada prinsipnya, proses pemecahan masalah pemrograman linear dengan menggunakan metode simpleks terjadi melalui algoritma, yaitu suatu urutan kerja secara teratur dan berulang sehingga tercapai hasil optimal yang dikehendaki. Metode ini paling efisien karena proses penyelesaian dapat digunakan program komputer yang sudah tentu akan menghabiskan waktu singkat bila dibandingkan secara manual. Dalam masalah pemrograman linear dengan kendala terlebih dahulu diubah menjadi bentuk kanonik. Bentuk kanonik adalah bentuk sistem persamaan linear dan memuat variabel basis (variabel yang memiliki koefisien 1).

Untuk membentuk kendala menjadi bentuk kanonik diperlukan penambahan variabel basis baru . Variabel basis baru tersebut adalah

a. Variabel *slack*¸ yaitu variabel yang dibutuhkan pada fungsi kendala yang memuat hubungan kurang dari atau sama dengan (≤).

   Contoh:

   $3x + 5y \leq 15$ diubah menjadi $3x + 5y + s_1 = 15$

   Sehingga s_1 menjadi variabel basis baru

b. Variabel *surplus*, yaitu variabel yang ditambahkan pada fungsi kendala yang memuat hubungan lebih dari atau sama dengan (≥).

   Contoh:

   $3x + 5y \geq 15$ diubah menjadi $3x + 5y - t_1 = 15$. Variabel t_1 bukan variabel basis (ketika di ruas kiri koefisiennya bukan +1)

c. Variabel *artificial*, yaitu variabel yang ditambahkan pada fungsi kendala yang belum memuat variabel basis pada poin b.

   Contoh:

   $3x + 5y - t_1 = 15$ perlu ditambahkan variabel *artificial* $q \geq 0$ sehingga menjadi

   $3x + 5y - t_1 + q_1 = 15$

Misal masalah pemrograman linear (2.1) dan (2.2) diubah ke bentuk kanonik, dengan menambahkan variabel *slack* di setiap kendala. Variabel *slack* atau sering disebut perubah pengetat pada fungsi tujuan memaksimumkan merupakan variabel yang berperan untuk membuat ruas yang semula longgar menjadi ketat, sehingga sama nilai dengan ruas yang lainnya (B. Susanta, 1994 :

69). Bentuk umum formulasi tersebut disajikan pada persamaan (2.3) sebagai berikut:

Memaksimumkan:

$$Z = c_1x_1 + c_2x_2 + \cdots + c_jx_j \tag{2.3}$$

Kendala:

$$\begin{aligned}
&a_{11}x_1 + a_{12}x_2 + \cdots + a_{1j}x_j + s_1 = b_1 \\
&a_{21}x_1 + a_{22}x_2 + \cdots + a_{2j}x_j + s_2 = b_2 \\
&\;\;\vdots \qquad\quad \vdots \qquad\qquad\quad \vdots \qquad \vdots \\
&a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{ij}x_j + s_m = b_i
\end{aligned}$$

Setelah bentuk umum tersebut diubah menjadi seperti (2.3) kemudian disusun ke dalam tabel. Diperoleh bentuk umum tabel simpleks 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_m	b_i	r_i
0	s_1	a_{11}	a_{12}	$\cdots$	a_{1j}	1	0	$\cdots$	0	b_1	r_1
0 $\vdots$	s_2 $\vdots$	a_{21} $\vdots$	a_{22}	$\cdots$	a_{2j}	0	1	$\cdots$	0	b_2 $\vdots$	r_2 $\vdots$
0	s_m	a_{i1}	a_{i2}	$\cdots$	a_{ij}	0	0	$\cdots$	1	b_m	r_m

(Sumber: B. Susanta, 1994 : 74)

Apabila suatu tabel belum optimum dan dipilih x_j sebagai baris baru maka disusun kolom r_i yang diperoleh dengan:

$r_i = \frac{b_i}{a_{ij}}$ hanya untuk $a_{ij} > 0$ dan $s_1, s_2, \cdots, s_m$ adalah variabel *slack* yang menunjukkan kapasitas sumber daya yang tidak dipergunakan (B. Susanta, 1994 : 74). Kasus dimana semua fungsi kendalanya berupa pertidaksamaan satu jenis disebut sebagai kasus minimum atau maksimum baku (B.Susanta, 1994 : 70).

Pada kasus memaksimumkan, tabel simpleks dinyatakan telah mencapai optimal jika $z_j - c_j \geq 0$ untuk semua nilai j. Jika tabel belum optimal maka dilakukan perbaikan tabel (iterasi). Pada kasus memaksimumkan, x_j terpilih adalah yang memiliki $z_j - c_j < 0$ yang paling kecil sehingga x_j terpilih untuk masuk menjadi basis baru. Kolom yang terpilih dinamakan kolom kunci. Variabel yang terpilih keluar dari basis adalah variabel dengan nilai r_i terkecil sehingga $\bar{x}_i$ terpilih untuk keluar dari basis. Baris $\bar{x}_i$ disebut baris kunci dan unsur pada baris kunci yang juga pada kolom kunci disebut unsur kunci (B. Susanta, 1994 : 77-78).

Pada kasus meminimumkan, tabel simpleks dinyatakan telah mencapai optimal jika $z_j - c_j \leq 0$ untuk semua nilai j. Jika masih ada nilai $z_j - c_j$ yang positif maka dilakukan perbaikan tabel (iterasi). Memilih x_j yang masuk menjadi basis baru dengan $z_j - c_j > 0$ yang paling besar sehingga x_j terpilih untuk masuk menjadi basis. Variabel yang terpilih keluar dari basis adalah variabe dengan nilai r_i terkecil sehingga $\bar{x}_i$ terpilih untuk keluar dari basis (B. Susanta, 1994 : 94-95).

Untuk menambah penjelasan terkait penggunaan metode simpleks, maka diberikan ilustrasi dalam Contoh 2.1 berikut ini:

Contoh 2.1 :

Seorang pedagang beras mempunyai persediaan beras A, beras B dan beras C masing-masing sebanyak 10 kg, 24 kg dan 16 kg. Jika pedagang menjual beras tersebut dalam 2 jenis karung yaitu karung X berisi campuran beras A, beras B dan beras C masing-masing sebanyak 1 kg, 3 kg, 1 kg. Karung Y berisi campuran beras A sebanyak 1 kg, beras B sebanyak 2 kg dan beras C sebanyak 2 kg. Jika keuntungan karung X \$28 dan keuntungan karung Y \$24 maka akan ditentukan besar keuntungan maksimum penjualan beras tersebut.

Masalah tersebut akan diselesaikan menggunakan metode simpleks dengan fungsi tujuan memaksimumkan keuntungan. Modelnya sebagai berikut:

Fungsi tujuan : Memaksimumkan $Z = 28x + 24y$ (2.4)

Fungsi Kendala : $x + y \leq 10$ (2.5)

$3x + 2y \leq 24$ (2.6)

$x + 2y \leq 16$ (2.7)

$x, y \geq 0$

Berdasarkan langkah-langkah penyelesaian pemrograman linear sederhana, masalah (2.4) diubah menjadi bentuk kanonik. Menambahkan variabel *slack* $s_1 \geq 0$ pada kendala (2.5), s_2 pada kendala (2.6) dan $s_3 \geq 0$ pada kendala (2.7) sehingga bentuk bakunya menjadi

Memaksimumkan $Z = 28x + 24y + 0s_1 + 0s_2 + 0s_3$

Kendala : $x + y + s_1 = 10$

$3x + 2y + s_2 = 24$

$x + 2y + s_3 = 16$

$x, y, s_1, s_2, s_3 \geq 0$

Penyelesaian akan dilakukan iterasi-iterasi menggunakan tabel simpleks sehingga diperoleh nilai yang optimal. Langkah awal disajikan pada Tabel 2.3 sebagai berikut:

Tabel 2.3 Tabel simpleks awal

	c_j	28	24	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	1	1	1	0	0	10	10
0	s_2	3	2	0	1	0	24	8
0	s_3	1	2	0	0	1	16	16
	z_j	0	0	0	0	0		
	$z_j - c_j$	-28	-24	0	0	0		

Berdasarkan Tabel 2.3 dapat dilihat bahwa nilai dari $z_j - c_j < 0$ yang artinya masih belum optimal. Nilai $z_j - c_j$ negatif terkecil ada pada kolom variabel x sehingga x merupakan variabel baru yang masuk dalam kolom basis (x_i). Nilai r_i terkecil adalah 8 pada variabel basis s_2 sehingga s_2 keluar digantikan oleh variabel x. Elemen yang terletak pada perpotongan kolom x dan baris s_2 merupakan elemen pivot. Untuk mengubah 3 menjadi 1 dilakukan Operasi Baris Elementer (OBE). Elemen pada kolom x lainnya, yaitu 1 diubah menjadi 0 dengan melakukan OBE mengurangi baris ke 1 dengan 1 kali baris ke 1 yang baru. Diperoleh tabel simpleks baru sebagai berikut:

Tabel 2.4 Tabel simpleks iterasi I

	c_j	28	24	0	0	0		
c_i	x_i/x_j	x	y	s_1	s_2	s_3	b_i	r_i
0	s_1	0	$1/3$	1	$-1/3$	0	2	$2/3$
28	x	1	$2/3$	0	$1/3$	0	8	$16/3$

0	s_3	0	$4/3$	0	$-1/3$	1	8	~
	z_j	28	$56/3$	0	$28/3$	0		
	$z_j - c_j$	0	$-16/3$	0	$28/3$	0		

Berdasarkan Tabel 2.4 dapat dilihat bahwa nilai dari $z_j - c_j < 0$ masih belum optimal. Nilai $z_j - c_j$ negatif terkecil ada pada kolom variabel y sehingga y merupakan variabel baru yang masuk dalam kolom basis. Nilai r_i terkecil adalah $2/3$ yang terletak pada variabel s_1 sehingga s_1 keluar digantikan oleh variabel y. Elemen yang terletak pada perpotongan kolom y dan baris s_1 merupakan elemen pivot. Untuk mengubah $1/3$ menjadi 1 dilakukan Operasi Baris Elementer (OBE). Elemen pada kolom y lainnya, yaitu $2/3$ diubah menjadi 0 dengan melakukan OBE mengurangi baris ke 2 dengan $2/3$ kali baris ke 1 yang baru. Diperoleh tabel simpleks baru sebagai berikut:

Tabel 2.5 Tabel simpleks iterasi akhir

	c_j	28	24	0	0	0		
c_i	x_i/x_j	x	y	s_1	s_2	s_3	b_i	r_i
24	y	0	1	3	-1	0	6	
28	x	1	0	-2	1	0	4	
0	s_3	0	0	-4	1	1	0	
	z_j	28	24	16	4	0		
	$z_j - c_j$	0	0	16	4	0		

Dari Tabel 2.5 terlihat bahwa tidak ada lagi $z_j - c_j < 0$ dengan demikian telah dicapai penyelesaian optimal

$\{x, y, s_1, s_2, s_3\} = \{4, 6, 0, 0, 0\}$

Kesimpulan dari penyelesaian masalah (2.3) adalah keuntungan maksimal dapat dicapai sebesar $256 dengan menjual 4 karung beras jenis A dan 6 karung beras jenis B.

B. Pemrograman Linear Tujuan Ganda

1. Arti Pemrograman Linear Tujuan Ganda

Pemrograman linear tujuan ganda (PLTG) merupakan program linear yang fungsi tujuannya lebih dari satu. Menurut Hiller dan Lieberman (1995 : 273) terdapat dua macam kasus yang harus ditetapkan dalam masalah pemrograman linear tujuan ganda. Kasus yang pertama disebut sebagai *non-preemtive multiobjective programming* dimana semua tujuan mempunyai kepentingan yang sama. PLTG yang termasuk *non-preemtive multiobjective programming* salah satunya adalah model *De Novo Programming*. Kasus yang kedua adalah *preemtive multiobjective programming*, dimana masing-masing tujuan mempunyai urutan (ranking) menurut prioritasnya. *Goal Programming, Lexicographic Goal Programming* merupakan PLTG yang termasuk dalam *preemtive multiobjective programming*. Masalah PLTG dapat dituliskan sebagai berikut:

Fungsi Tujuan :

Maksimumkan $Z_k = c_{k1}x_1 + c_{k2}x_2 + \cdots + c_{kj}x_j \quad (k = 1, 2, \cdots, n)$ (2.8)

Meminimumkan $W_s = c_{s1}x_1 + c_{s2}x_2 + \cdots + c_{sj}x_j \quad (s = 1, 2, \cdots, n)$ (2.8a)

Fungsi Kendala :

$$a_{i1}x_1 + a_{i2}x_2 \cdots + a_{ij}x_j \leq b_i \qquad (2.9)$$
$$x_1, x_2, \cdots, x_n, b_i \geq 0$$

dengan

c_{kj} : koefisien ongkos/ harga jual satu unit j dengan tujuan maksimum ke$-k$ pada PLTG

Z_k : fungsi tujuan maksimum ke$-k$ pada PLTG ($k = 1, 2, \cdots, n$)

c_{sj} : koefisien ongkos/ harga jual satu unit j dengan tujuan meminimumkan ke$-s$ pada PLTG

W_s : fungsi tujuan meminimumkan ke$-s$ pada PLTG ($s = 1, 2, \cdots, n$)

a_{ij} : koefisien kendala/ bahan mentah ke$-i$ yang digunakan untuk memproduksi satu unit produk j

b_i : suku tetap/ bahan mentah jenis ke$-i$ yang tersedia

Pada proses pengoptimalan, tidak semua masalah pemrograman linear dapat diselesaikan oleh PLTG karena kendala keterbatasan anggaran (Pert Fiala, 2011). Berdasarkan permasalahan tersebut, perlu mengkaji model PLTG lainnya salah satunya adalah model *De Novo programming*.

2. Model *De Novo Programming*

Menyelesaikan masalah tujuan ganda, tidak semuanya dapat diselesaikan menggunakan PLTG biasa, maka akan digunakan model dari perluasan PLTG yaitu model *De novo Programming*. Model *De Novo Programming* termasuk pemrograman linear dengan tujuan ganda. Menurut Zenely (2005), model *De Novo Programming* merupakan kasus *non-preemtive multiobjective programming* yang semua tujuan mempunyai kepentingan sama (Normalita Wijayanti, 2014). Perbedaan yang terjadi antara PLTG biasa dengan model *De Novo Programming* terletak pada adanya kendala yang melibatkan total anggaran (*budget*) yang tersedia dalam model *De Novo Programming*. Menggunakan formula masalah PLTG (2.8), (2.8a) dan (2.9), maka didapat model *De Novo Programming* sebagai berikut

Fungsi Tujuan :

Maksimumkan $Z_k = c_{k1}x_1 + c_{k2}x_2 + \cdots + c_{kj}x_j \quad (k = 1, 2, \cdots, n)$ (2.10)

Meminimumkan $W_s = c_{s1}x_1 + c_{s2}x_2 + \cdots + c_{sj}x_j \quad (s = 1, 2, \cdots, n)$ (2.10a)

Fungsi Kendala :

$$a_{i1}x_1 + a_{i2}x_2 \cdots + a_{ij}x_j \leq b_i \tag{2.11}$$
$$p_1b_1 + p_2b_2 + \cdots + p_ib_i \leq B \tag{2.11a}$$
$$x_1, x_2, \cdots, x_n, b_i \geq 0$$

Keterangan:

B : total anggaran (*budget*) yang tersedia

p_i : harga per unit dari sumber ke$-i$ $(i = 1, 2, \cdots, m)$

Berdasarkan informasi di atas, merujuk formula *De Novo Programming* (2.10), (2.10a), (2.11) dan (2.11a) dapat dijabarkan menjadi sebagai berikut:

Fungsi Tujuan :

Memaksimumkan :
$$\begin{aligned} Z_1 &= c_{11}x_1 + c_{12}x_2 + \cdots + c_{1j}x_j \\ \vdots \quad & \quad \vdots \qquad \vdots \qquad\qquad \vdots \\ Z_k &= c_{k1}x_1 + c_{k2}x_2 + \cdots + c_{kj}x_j \end{aligned} \tag{2.12}$$

Meminimumkan :
$$\begin{aligned} W_1 &= c_{11}x_1 + c_{12}x_2 + \cdots + c_{1j}x_j \\ \vdots \quad & \quad \vdots \qquad \vdots \qquad\qquad \vdots \\ W_s &= c_{s1}x_1 + c_{s2}x_2 + \cdots + c_{sj}x_j \end{aligned} \tag{2.12a}$$

Fungsi kendala :

$$\begin{aligned} &a_{11}x_1 + a_{12}x_2 + \cdots + a_{1j}x_j = b_1 \\ &a_{21}x_1 + a_{22}x_2 + \cdots + a_{2j}x_j = b_2 \\ &\vdots \qquad \vdots \qquad\qquad \vdots \qquad \vdots \\ &a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{ij}x_j = b_i \end{aligned} \tag{2.13}$$
$$p_1b_1 + p_2b_2 + \cdots + p_ib_i \leq B \tag{2.13a}$$

Fungsi kendala non-negatif

$x_1, x_2, \cdots, x_n, b_i \geq 0$

Berdasarkan fungsi kendala (2.13) dan (2.13a) dapat dibentuk variabel v_j yang artinya variabel *cost* untuk membuat 1 unit produk :

$$v_j = p_1a_{i1} + p_2a_{i2} + \cdots + p_ia_{ij} \tag{2.14}$$

dengan:

v_j : variabel *cost* untuk membuat 1 unit produk j $(j = 1, 2, \cdots, n)$

p_i : harga per unit dari sumber ke$-i$ $(i = 1, 2, \cdots, m)$

a_{ij} : koefisien kendala/ bahan mentah ke$-i$ yang digunakan untuk memproduksi satu unit produk j

Persamaan (2.14) dapat diuraikan sebagai berikut:

$$v_1 = p_1 a_{11} + p_2 a_{21} + \cdots + p_i a_{i1} \tag{2.15}$$
$$v_2 = p_1 a_{21} + p_2 a_{22} + \cdots + p_i a_{i2} \tag{2.15a}$$
$$\vdots \quad \vdots \quad \vdots \quad \vdots$$
$$v_j = p_1 a_{i1} + p_2 a_{i2} + \cdots + p_i a_{ij} \tag{2.15b}$$

Apabila persamaan (2.13) disubstitusikan ke pertidaksamaan (2.13a) maka diperoleh:

$$p_1\left(a_{11}x_1 + a_{12}x_2 + \cdots + a_{1j}x_j\right) + p_2(a_{21}x_2 + a_{22}x_2 + \cdots + a_{2j}x_j) + \cdots + p_i(a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{ij}x_j) \leq B \tag{2.16}$$

Dengan mensubstitusikan persamaan (2.15), (2.15a) dan (2.15b) ke pertidaksamaan (2.16) maka didapat pertidaksamaan sebagai berikut:

$$v_1x_1 + v_2x_2 + \cdots + v_jx_j \leq B\,.$$

Sehingga formulasi model *De Novo Programming* menjadi:

Memaksimumkan : $Z_k = c_{k1}x_1 + c_{k2}x_2 + \cdots + c_{kj}x_j$ (2.17)

Meminimumkan : $W_s = c_{s1}x_1 + c_{s2}x_2 + \cdots + c_{sj}x_j$ (2.17a)

Kendala :

$$v_1x_1 + v_2x_2 + \cdots + v_jx_j \leq B \tag{2.18}$$
$$x_1, x_2, \cdots, x_n \geq 0$$

a. Mencari solusi optimal maksimum dan solusi optimal minimum Model *De Novo Programming*

Mencari solusi optimal maksimum dan solusi optimal minimum pada model *De Novo Programming* langkah-langkahnya adalah sebagai berikut (Petr Fiala, 2011 : 31-32):

1) Menyelesaikan setiap fungsi tujuan memaksimumkan dan fungsi tujuan meminimumkan dalam model *De Novo Programming* $Z_1, Z_2, \cdots, Z_k$ dan $W_1, W_2, \cdots, W_s$ satu per satu secara terpisah, yaitu sebagai berikut:

   a) Mencari solusi optimal maksimum

      (1) Memaksimumkan: $Z_1 = c_{11}x_1 + c_{12}x_2 + \cdots + c_{1j}x_j$

      Meminimumkan: $W_1 = c_{11}x_1 + c_{12}x_2 + \cdots + c_{1j}x_j$

      Kendala : $v_1x_1 + v_2x_2 + \cdots + v_jx_j \leq B$
      $x_1, x_2, \cdots, x_n \geq 0$

      Dengan demikian, diperoleh nilai $Z_1^* =$ maks Z_1 dan $W_1^* =$ minW_1

      (2) Memaksimumkan: $Z_2 = c_{21}x_1 + c_{22}x_2 + \cdots + c_{2j}x_j$

      Meminimumkan: $W_2 = c_{11}x_1 + c_{12}x_2 + \cdots + c_{1j}x_j$

      Kendala : $v_1x_1 + v_2x_2 + \cdots + v_jx_j \leq B$
      $x_1, x_2, \cdots, x_n \geq 0$

      Dengan demikian, diperoleh nilai $Z_2^* =$ maks Z_2 dan $W_2^* =$ minW_2

      $\vdots$

      (3) Memaksimumkan: $Z_k = c_{k1}x_1 + c_{k2}x_2 + \cdots + c_{kj}x_j$

      Meminimumkan: $W_s = c_{s1}x_1 + c_{s2}x_2 + \cdots + c_{sj}x_j$

      Kendala : $v_1x_1 + v_2x_2 + \cdots + v_jx_j \leq B$
      $x_1, x_2, \cdots, x_n \geq 0$

      Dengan demikian, diperoleh nilai $Z_k^* =$ maks Z_k dan $W_s^* =$ minW_s

   b) Mencari solusi optimal minimum

      (1) Meminimumkan: $Z_1 = c_{11}x_1 + c_{12}x_2 + \cdots + c_{1j}x_j$

      Memaksimumkan: $W_1 = c_{11}x_1 + c_{12}x_2 + \cdots + c_{1j}x_j$

      Kendala : $v_1x_1 + v_2x_2 + \cdots + v_jx_j \leq B$

$$x_1, x_2, \cdots, x_n \geq 0$$

Dengan demikian, diperoleh nilai $Z_1^* = \min Z_1$ dan $W_1^* = \text{maks} W_1$

(2) Meminimumkan: $Z_2 = c_{21}x_1 + c_{22}x_2 + \cdots + c_{2j}x_j$

Memaksimumkan: $W_2 = c_{21}x_1 + c_{22}x_2 + \cdots + c_{2j}x_j$

Kendala : $v_1x_1 + v_2x_2 + \cdots + v_jx_j \leq B$

$x_1, x_2, \cdots, x_n \geq 0$

Dengan demikian, diperoleh nilai $Z_2^* = \min Z_2$ dan $W_2^* = \text{maks} W_2$

$\vdots$

(3) Meminimumkan: $Z_k = c_{k1}x_1 + c_{k2}x_2 + \cdots + c_{kj}x_j$

Memaksimumkan: $W_s = c_{s1}x_1 + c_{s2}x_2 + \cdots + c_{sj}x_j$

Kendala : $v_1x_1 + v_2x_2 + \cdots + v_jx_j \leq B$

$x_1, x_2, \cdots, x_n \geq 0$

Dengan demikian, diperoleh nilai $Z_k^* = \min Z_k$ dan $W_s^* = \text{maks} W_s$

2) Menurut Li dan Lee (Nurullah Umarusman, 2013) bahwa solusi optimal maksimum dan solusi optimal minimum dapat ditulis sebagai berikut:

Nilai maksimum untuk Z_k adalah $Z_k^* = \text{maks}\, Z_k$ dan untuk W_s adalah $W_s^* = \min W_s$ dituliskan pada persamaan (2.19a) seperti berikut:

$$I^* = \{Z_1^*, \cdots, Z_k^* \,;\, W_1^*, \cdots, W_s^*\} \quad (2.19a)$$

Nilai minimum untuk Z_k adalah $Z_k^* = \min Z_k$ dan W_s adalah $W_s^* = \text{maks} W_s$ dituliskan pada persamaan (2.19b) seperti berikut:

$$I^- = \{Z_1^-, \cdots, Z_k^- \,;\, W_1^-, \cdots, W_s^-\} \quad (2.19b)$$

Tujuannya adalah untuk mengidentifikasi solusi optimal maksimum dan solusi optimal minimum setiap fungsi tujuan. Untuk memperjelas cara mencari solusi optimal maksimum dan solusi optimal minimum dalam model *De Novo Programming*, dapat diilustrasikan dalam sebuah Contoh 2.2 sebagai berikut:

Contoh 2.2 :

Menyelesaikan masalah tujuan ganda model *De Novo Programming* untuk mencari solusi optimal maksimum dan solusi optimal minimum

Maksimumkan $Z_1 = 50x_1 + 100x_2 + 17{,}5x_3$ (keuntungan)

$Z_2 = 92x_1 + 75x_2 + 50x_3$ (kualitas)

$Z_3 = 25x_1 + 100x_2 + 75x_3$ (upah pekerja)

Dengan kendala :

$$12x_1 + 17x_2 \leq b_1$$
$$3x_1 + 9x_2 + 8x_3 \leq b_2$$
$$10x_1 + 13x_2 + 15x_3 \leq b_3$$
$$6x_1 + 16x_3 \leq b_4$$
$$12x_2 + 7x_3 \leq b_5$$
$$9{,}5x_1 + 9{,}5x_2 + 4x_3 \leq b_6$$
$$x_1, x_2, x_3 \geq 0$$

Diasumsikan fungsi tujuan Z_1, Z_2 dan Z_3 memiliki prioritas yang sama. Akan diidentifikasi tingkat sumber daya dari b_1 sampai $b_6$dengan harga unit $p_1 = 0{,}75$, $p_2 = 0{,}60$, $p_3 = 0{,}35$, $p_4 = 0{,}45$, $p_5 = 1{,}15$ dan $p_6 = 0{,}65$. Anggaran awal (*Budget*) dimisalkan dengan $B = \$\ 4.658{,}75$

Berdasarkan data yang diketahui, langkah awal adalah mengubah fungsi kendala tersebut dalam model *De Novo Programming* dengan merujuk (2.14) dan (2.18) sehingga menjadi

$$23{,}475x_1 + 42{,}674x_2 + 28{,}7x_3 \leq 4.658{,}75$$
$$x_1, x_2, x_3 \geq 0$$

Selanjutnya fungsi kendala tersebut akan diubah ke bentuk kanonik dengan mengubah ($\leq$) menjadi ($=$) dengan menambahkan variabel *slack*, sehingga bentuk permasalahan pada model *De Novo Programming* Contoh 2.2 adalah

Fungsi tujuan :

$$\left.\begin{aligned} \text{Maksimumkan } Z_1 &= 50x_1 + 100x_2 + 17{,}5x_3 \\ Z_2 &= 92x_1 + 75x_2 + 50x_3 \\ Z_3 &= 25x_1 + 100x_2 + 75x_3 \end{aligned}\right\} \quad (2.20)$$

Dengan kendala :

$$23{,}475x_1 + 42{,}674x_2 + 28{,}7x_3 + s_1 = 4.658{,}75 \quad (2.21)$$
$$x_1, x_2, x_3 \geq 0$$

Selanjutnya masalah ini diselesaikan satu per satu menggunakan metode simpleks seperti berikut:

Fungsi tujuan:
Memaksimumkan $Z_1 = 50x_1 + 100x_2 + 17.5x_3$

Fungsi kendala:
$23{,}475x_1 + 42{,}674x_2 + 28{,}7x_3 + s_1 = 4.658{,}75$

Tabel 2.6 Tabel simpleks iterasi awal

	c_j	50	100	17.5	0		
$\bar{c}_i$	$\bar{x}_i/x_j$	x_1	x_2	x_3	s_1	b_i	r_i
0	s_1	23,475	42,674	28,7	1	4.658,75	109,1707
	z_j	0	0	0	0	0	
	$z_j - c_j$	-50	-100	-17,5			

Berdasarkan Tabel 2.6 dapat dilihat bahwa nilai dari $z_j - c_j < 0$ belum optimal. Nilai $z_j - c_j$ negatif terkecil ada pada kolom variabel x_2 sehingga x_2 merupakan variabel baru yang masuk dalam kolom basis. Nilai r_i terkecil adalah 109,1707 yang terletak pada variabel s_1 sehingga s_1 keluar digantikan oleh variabel x_2. Elemen yang terletak pada perpotongan kolom x_2 dan baris s_1 merupakan elemen pivot, untuk mengubah 42,674 menjadi 1 dilakukan Operasi Baris Elementer (OBE). Diperoleh tabel simpleks baru sebagai berikut:

Tabel 2.7 Tabel simpleks iterasi akhir

	c_j	50	100	17,5	0		
$\bar{c}_i$	$\bar{x}_i/x_j$	x_1	x_2	x_3	s_1	b_i	r_i
100	x_2	0,550101	1	0,672541	0,023433	109,1707	
	z_j	55,01008	100	67,25407	2,343347	10,917,07	
	$z_j - c_j$	5,010076	0	49,75407	2,343347		

Dari Tabel 2.7 terlihat bahwa tidak ada lagi $z_j - c_j < 0$ dengan demikian telah didapatkan solusi optimal maksimum untuk Z_1, $Z_1^* = \{10.917{,}07\}$.

Fungsi Tujuan:
Memaksimumkan $Z_2 = 92x_1 + 75x_2 + 50x_3$

Fungsi kendala:
$23{,}475x_1 + 42{,}674x_2 + 28{,}7x_3 + s_1 = 4.658{,}75$

Tabel 2.8 Tabel simpleks iterasi awal

	c_j	92	75	50	0		
$\bar{c}_i$	$\bar{x}_i/x_j$	x_1	x_2	x_3	s_1	b_i	r_i
0	s_1	23,475	42,674	28,7	1	4658,75	198,4558
	z_j	0	0	0	0	0	
	$z_j - c_j$	-92	-75	-50			

Berdasarkan Tabel 2.8 dapat dilihat bahwa nilai dari $z_j - c_j < 0$ belum optimal. Nilai $z_j - c_j$ negatif terkecil ada pada kolom variabel x_1 sehingga x_1 merupakan variabel baru yang masuk dalam kolom basis. Nilai r_i terkecil adalah 198,4558 yang terletak pada variabel s_1 sehingga s_1 keluar digantikan oleh variabel x_1. Elemen yang terletak pada perpotongan kolom x_1 dan baris s_1

merupakan elemen pivot, untuk mengubah 23,475 menjadi 1 dilakukan Operasi Baris Elementer (OBE). Diperoleh tabel simpleks baru sebagai berikut:

Tabel 2.9 Tabel simpleks iterasi akhir

	c_j	92	75	50	0		
$\bar{c}_i$	$\bar{x}_i/x_j$	x_1	x_2	x_3	s_1	b_i	r_i
92	x_1	1	1,817849	1,222577	0,042599	198,4558	
	z_j	92	167,2421	112,4771	3,919063	18257,93	
	$z_j - c_j$	0	92,24209	62,4771	3,919063		

Dari Tabel 2.9 terlihat bahwa tidak ada lagi $z_j - c_j < 0$ dengan demikian telah didapatkan solusi optimal maksimum untuk Z_2. $Z_2^* = \{18.257{,}93\}$.

Fungsi tujuan:
Memaksimumkan $Z_3 = 25x_1 + 100x_2 + 75x_3$

Fungsi kendala:
$23{,}475x_1 + 42{,}674x_2 + 28{,}7x_3 + s_1 = 4.658{,}75$

Tabel 2.10 Tabel simpleks iterasi awal

	c_j	25	100	75	0		
$\bar{c}_i$	$\bar{x}_i/x_j$	x_1	x_2	x_3	s_1	b_i	r_i
0	s_1	23,475	42,674	28,7	1	4658,75	109,1707
	z_j	0	0	0	0	0	
	$z_j - c_j$	-25	-100	-75	0		

Berdasarkan Tabel 2.10 dapat dilihat bahwa nilai dari $z_j - c_j < 0$ belum optimal. Nilai $z_j - c_j$ negatif terkecil ada pada kolom variabel x_2 sehingga x_2 merupakan variabel baru yang masuk dalam kolom basis. Nilai r_i terkecil adalah 109,1707 yang terletak pada variabel s_1 sehingga s_1 keluar digantikan oleh variabel x_2. Elemen yang terletak pada perpotongan kolom x_2 dan baris s_1

merupakan elemen pivot, untuk mengubah 42,674 menjadi 1 dilakukan Operasi Baris Elementer (OBE). Diperoleh tabel simpleks baru sebagai berikut:

Tabel 2.11 Tabel simpleks iterasi pertama

	c_j	25	100	75	0		
$\bar{c}_i$	$\bar{x}_i/x_j$	x_1	x_2	x_3	s_1	b_i	r_i
100	x_2	0,550101	1	0,672541	0,023433	109,1707	162,3258
	z_j	55,01008	100	67,25407	2,343347	10917,07	
	$z_j - c_j$	30,01008	0	-7,74593	2,343347		

Berdasarkan Tabel 2.11 dapat dilihat bahwa nilai dari $z_j - c_j < 0$ artinya tabel masih belum optimal. Nilai $z_j - c_j$ negatif terkecil ada pada kolom variabel x_3 sehingga x_3 merupakan variabel baru yang masuk dalam kolom basis. Nilai r_i terkecil adalah 162,3258 yang terletak pada variabel x_2 sehingga x_2 keluar digantikan oleh variabel x_3. Elemen yang terletak pada perpotongan kolom x_3 dan baris x_2 merupakan elemen pivot, untuk mengubah 0,672541 menjadi 1 dilakukan Operasi Baris Elementer (OBE). Diperoleh tabel simpleks baru sebagai berikut:

Tabel 2.12 Tabel simpleks iterasi akhir

	c_j	25	100	75	0		
$\bar{c}_i$	$\bar{x}_i/x_j$	x_1	x_2	x_3	s_1	b_i	r_i
75	x_3	0,817944	1,486899	1	0,034843	162,3258	162,3258
	z_j	61,34582	111,5174	75	2,61324	12174,43	
	$z_j - c_j$	36,34582	11,51742	0	2,61324		

Dari Tabel 2.12 terlihat bahwa tidak ada lagi $z_j - c_j < 0$ dengan demikian telah didapatkan solusi optimal maksimum untuk Z_3. $Z_3^* = \{12.174,43\}$.

Berdasarkan hasil perhitungan menggunakan metode simpleks diperoleh solusi optimal maksimum $Z_k^* = \text{maks}\, Z_k$,

$Z_k^* = \{10.916,81 ; 18.257,93 ; 12.174,43\}$

Langkah selanjutnya adalah mencari nilai $Z_k^* = \min Z_k$ menggunakan fungsi tujuan (2.20) dan kendala (2.21) dengan tujuan meminimumkan.

Fungsi tujuan :

Meminimumkan $Z_1 = 50x_1 + 100x_2 + 17{,}5x_3$

$Z_2 = 92x_1 + 75x_2 + 50x_3$

$Z_3 = 25x_1 + 100x_2 + 75x_3$

Dengan kendala :

$23{,}475x_1 + 42{,}674x_2 + 28{,}7x_3 + s_1 = 4{,}658{,}75$

$x_1, x_2, x_3, \geq 0$

Selanjutnya masalah ini diselesaikan satu per satu menggunakan metode simpleks seperti berikut:

Fungsi Tujuan:
Meminimumkan $Z_1 = 50x_1 + 100x_2 + 17.5x_3$

Fungsi Kendala:
$23{,}475x_1 + 42{,}674x_2 + 28{,}7x_3 + s_1 = 4.658{,}75$

Tabel 2.13 Tabel simpleks iterasi awal

	c_j	50	100	17,5	0		
$\bar{c}_i$	$\bar{x}_i/x_j$	x_1	x_2	x_3	s_1	b_i	r_i
0	s_1	23,475	42,674	28,7	1	4.658,75	162,3258
	z_j	0	0	0	0	0	
	$z_j - c_j$	-50	-100	-17,5	0		

Berdasarkan Tabel 2.13 dapat dilihat bahwa nilai dari $z_j - c_j > 0$ artinya tabel belum optimal. Nilai $z_j - c_j$ positif terbesar ada pada kolom variabel x_3

sehingga x_3 merupakan variabel baru yang masuk dalam kolom basis. Nilai r_i terkecil adalah 162,3258 yang terletak pada variabel s_1 sehingga s_1 keluar digantikan oleh variabel x_3. Elemen yang terletak pada perpotongan kolom x_3 dan baris s_1 merupakan elemen pivot, untuk mengubah 28,7 menjadi 1 dilakukan Operasi Baris Elementer (OBE). Diperoleh tabel simpleks baru sebagai berikut:

Tabel 2.14 Tabel simpleks iterasi akhir

	c_j	50	100	17.5	0		
$\bar{c}_i$	$\bar{x}_i/x_j$	x_1	x_2	x_3	s_1	b_i	r_i
17.5	x_3	0.81794425	1.486898955	1	0.034843	162.3258	
	z_j	14.3140244	26.02073171	17.5	0.609756	2840.701	
	$z_j - c_j$	-35.6859756	-73.97926829	0	0.609756		

Dari Tabel 2.14 terlihat bahwa tidak ada lagi $z_j - c_j > 0$ dengan demikian telah didapatkan solusi optimal minimum untuk Z_1, $Z_1^* = \{2.840,701\}$.

Fungsi Tujuan:
Meminimumkan $Z_2 = 92x_1 + 75x_2 + 50x_3$

Fungsi Kendala:
$23{,}475x_1 + 42{,}674x_2 + 28{,}7x_3 + s_1 = 4.658{,}75$

Tabel 2.15 Tabel simpleks iterasi awal

	c_j	92	75	50	0		
$\bar{c}_i$	$\bar{x}_i/x_j$	x_1	x_2	x_3	s_1	b_i	r_i
0	s_1	23.475	42.674	28.7	1	4658.75	198.4558
	z_j	0	0	0	0	0	
	$z_j - c_j$	-92	-75	-50			

Berdasarkan Tabel 2.15 dapat dilihat bahwa nilai dari $z_j - c_j > 0$ artinya tabel belum optimal. Nilai $z_j - c_j$ positif terbesar ada pada kolom variabel x_3 sehingga x_3 merupakan variabel baru yang masuk dalam kolom basis. Nilai r_i

terkecil adalah 198,4558 yang terletak pada variabel s_1 sehingga s_1 keluar digantikan oleh variabel x_3. Elemen yang terletak pada perpotongan kolom x_3 dan baris s_1 merupakan elemen pivot, untuk mengubah 28,7 menjadi 1 dilakukan Operasi Baris Elementer (OBE). Diperoleh tabel simpleks baru sebagai berikut:

Tabel 2.16 Tabel simpleks iterasi akhir

	c_j	92	75	50	0		
$\bar{c}_i$	$\bar{x}_i/x_j$	x_1	x_2	x_3	s_1	b_i	r_i
50	x_3	0,81794425	1,486898955	1	0,034843	162,3258	198,4558
	z_j	40,8972125	74,34494774	50	1,74216	8116,289	
	$z_j - c_j$	-51,1027875	-0,655052265	0	1,74216		

Dari Tabel 2.16 terlihat bahwa tidak ada lagi $z_j - c_j > 0$ dengan demikian telah didapatkan solusi optimal minimum untuk Z_2, $Z_2^* = \{8.116{,}289\}$.

Fungsi Tujuan:
Meminimumkan $Z_3 = 25x_1 + 100x_2 + 75x_3$

Fungsi Kendala:
$23{,}475x_1 + 42{,}674x_2 + 28{,}7x_3 + s_1 = 4.658{,}75$

Tabel 2.17 Tabel simpleks iterasi awal

	c_j	25	100	75	0		
$\bar{c}_i$	$\bar{x}_i/x_j$	x_1	x_2	x_3	s_1	b_i	r_i
0	s_1	23.475	42.674	28.7	1	4658.75	198.4558
	z_j	0	0	0	0	0	
	$z_j - c_j$	-25	-100	-75	0		

Berdasarkan Tabel 2.17 dapat dilihat bahwa nilai dari $z_j - c_j > 0$ artinya tabel belum optimal. Nilai $z_j - c_j$ terbesar ada pada kolom variabel x_1 sehingga x_1 merupakan variabel baru yang masuk dalam kolom basis. Nilai r_i terkecil adalah 198,4558 yang terletak pada variabel s_1 sehingga s_1 keluar digantikan oleh variabel x_1. Elemen yang terletak pada perpotongan kolom x_1 dan baris s_1 merupakan elemen pivot, untuk mengubah 23,475 menjadi 1 dilakukan Operasi Baris Elementer (OBE). Diperoleh tabel simpleks baru sebagai berikut:

Tabel 2.18 Tabel simpleks iterasi akhir

	c_j	25	100	75	0		
$\bar{c}_i$	$\bar{x}_i/x_j$	x_1	x_2	x_3	s_1	b_i	r_i
25	x_1	1	1,817848775	1,2225772	0,042599	198,4558	198,4558
	z_j	25	45,44621938	30,56443	1,064963	4961,395	
	$z_j - c_j$	0	-54,55378062	-44,43557	1,064963		

Dari Tabel 2.18 terlihat bahwa tidak ada lagi $z_j - c_j > 0$ dengan demikian telah didapatkan solusi optimal minimum untuk Z_3, $Z_3^* = \{4.961{,}395\}$.

Berdasarkan hasil perhitungan menggunakan metode simpleks diperoleh solusi optimal minimum $Z_k^* = \min Z_k$, $Z_k^* = \{2.840{,}70\,;8.116{,}29\,;4.961{,}40\}$. Merujuk persamaan (2.19a) dan (2.19b) didapatkan solusi optimal maksimum dan solusi optimal minimum secara berturut-turut adalah:

$I^* = \{10{,}916{,}81\,;18{,}257{,}93\,;12{,}174{,}43\}$ dan

$I^- = \{2{,}840{,}701\,;8{,}116{,}289\,;4{,}961{,}395\}$

3. Metode Goal Programming

A.Charnes dan W.M. Cooper melihat bahwa pemrograman linear sederhana tidak mampu menyelesaikan kasus-kasus manajemen yang menghendaki sasaran-sasaran tertentu dicapai secara optimal. Pada tahun 1961 mereka mulai mempopulerkan model *Goal Programming* yang mampu menyelesaikan kasus-kasus pemrograman linear yang memiliki lebih dari satu sasaran yang hendak dicapai (Siswanto, 2007 : 341).

Model *Goal Programming* merupakan pemrograman linear tujuan ganda perluasan dari model pemrograman linear, sehingga seluruh asumsi, notasi, formulasi model matematis, prosedur perumusan model dan penyelesaiannya tidak jauh berbeda. Perbedaan hanya terletak pada kehadiran sepasang variabel deviasional yang muncul di fungsi tujuan dan fungsi-fungsi kendala (Siswanto, 2007 : 341). Di dalam model *Goal Programming*, Charnes dan Cooper menghadirkan sepasang variabel yang dinamakan variabel deviasional. Variabel deviasional berfungsi untuk menampung penyimpangan atau deviasi yang akan terjadi pada nilai ruas kiri suatu persamaan kendala terhadap nilai ruas kanannya.

Menyelesaikan model *Goal Programming* dapat dilakukan dengan beberapa metode dan pendekatan di antaranya adalah *lexicographic Goal Programming* yang merupakan kategori *pre-emptive*, *Chebyshev Goal Programming*, *fuzzy Goal Programming* dan *Weighted Goal Programming* (WGP). Metode WGP merupakan metode yang termasuk dalam kasus *non-preemptive multiobjective programming* yang artinya masing-masing tujuan mempunyai kepentingan yang sama. Tujuan dari metode *Weighted Goal Programming* adalah untuk meminimalkan semua deviasi (penyimpangan) yang terbentuk antara fungsi tujuan dan fungsi kendala (Carlos Romero, 1991 : 3). Secara algoritma metode *Weighted Goal Programming* ditulis sebagai berikut:

Fungsi tujuan:

$$\min \sum_{i=1}^{k} (\alpha_i d_i^- + \beta_i d_i^+) \qquad (2.22)$$

fungsi kendala

$$f_i(x) + d_i^- - d_i^+ = b_i \qquad (2.23)$$

$$x, d_i^-, d_i^+ \geq 0$$

Keterangan:

α_i dan β_i : bobot positif masing-masing fungsi tujuan ke$-i$

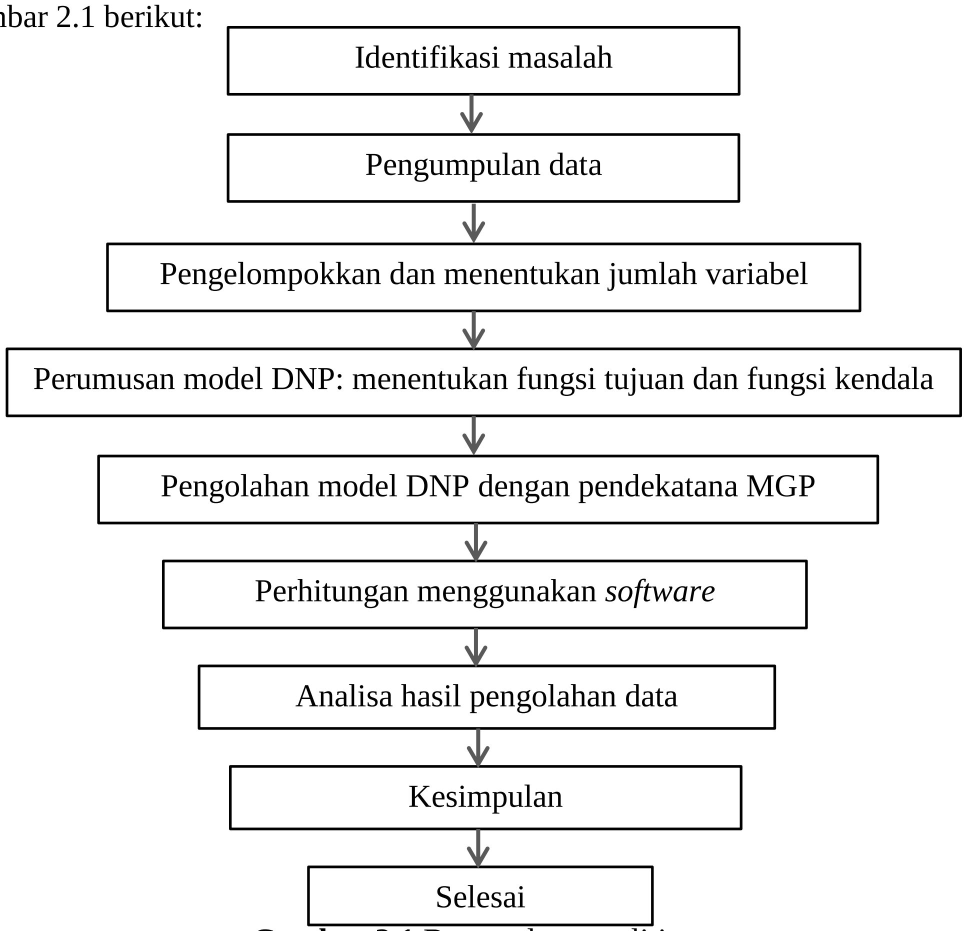
d_i^- : jumlah unit deviasi yang kekurangan terhadap tujuan ke$-i$

d_i^+ : jumlah unit deviasi yang kelebihan terhadap tujuan$-i$

b_i : suku tetap / bahan mentah jenis ke$-i$ yang tersedia

Metode *Weighted Goal Programming* tersebut akan digunakan untuk membentuk metode min-max *Goal Programming* yang akan dibahas pada bab III.

Secara umum, alur penelitian menggunakan model *De Novo Programming* dengan pendekatan min-max *Goal Programming* dapat diilustrasikan pada Gambar 2.1 berikut:



Gambar 2.1 Bagan alur penelitian

C. Perencanaan Produksi

Perencanaan produksi adalah suatu kegiatan yang berkenaan dengan penentuan apa yang harus diproduksi, berapa banyak yang diproduksi, kapan diproduksi dan apa sumber daya yang dibutuhkan untuk mendapatkan produk yang telah ditetapkan (Sukaria Sinulingga, 2009 : 26). Menurut Hendra Kusuma (2009 : 1) bahwa tujuan dari perencanaan produksi adalah merencanakan aliran material ke dalam, di dalam, dan keluar perusahaan sehingga posisi keuntungan menjadi optimal yang merupakan tujuan yang akan dicapai.

Pada dasarnya fungsi dasar yang harus dipenuhi oleh aktivitas perencanaan dan pengendalian produksi adalah (Hendra Kusuma, 2009 : 2):

a. Menetapkan jumlah dan saat pemesanan bahan baku serta komponen secara ekonomis dan terpadu,
b. Menetapkan keseimbangan antara tingkat kebutuhan produksi, teknik pemenuhan pesanan, serta memonitor tingkat persediaan produk jadi setiap saat, membandingkan dengan rencana persediaan; serta
c. Membuat jadwal produksi, penugasan, pembebanan mesin dan tenaga kerja yang terperinci sesuai dengan ketersediaan kapasitas dan fluktuasi permintaan pada suatu periode.

Dalam penelitian ini hanya akan ditentukan banyaknya produksi bakpia di *home industry* Bakpia 716 Annur Yogyakarta. Data yang digunakan adalah data produksi bakpia selama satu tahun, data pemesanan bakpia selama satu tahun, data komposisi bahan baku per unit dan harga bahan baku.