BAB II
LANDASAN TEORI

2.1 Program Linier dengan Metode Simpleks

2.1.1 Program Linier

Program Linier merupakan salah satu teknik riset operasi yang digunakan paling luas dan diketahui dengan baik. Metode tersebut merupakan metode matematik yang mengalokasikan sumberdaya yang langka (terbatas) untuk mencapai suatu tujuan seperti memaksimumkan keuntungan dan meminimumkan biaya. Program linier dapat menjelaskan suatu fenomena dunia nyata yang dibentuk dengan model matematik yang terdiri dari sebuah fungsi tujuan linier dan beberapa kendalanya.

Masalah keputusan yang sering dihadapi dalam analisis adalah alokasi sumberdaya yang langka. Tujuan analisis adalah untuk mencapai hasil terbaik yang mungkin dengan keterbatasan sumberdaya ini. Hasil yang dicapai ditunjukkan sebagai maksimisasi dari beberapa ukuran seperti profit, penjualan atau minimisasi seperti pada biaya.

Setelah masalah diidentifikasikan dan tujuan ditetapkan langkah selanjutnya adalah memformulasikan model matematik yang meliputi tiga tahap sebagai berikut:

1. Tentukan variabel keputusan dan nyatakan dalam simbol matematik.
2. Membentuk fungsi tujuan yang ditunjukkan sebagai suatu hubungan linier dari variabel keputusan.

3. Menentukan semua kendala masalah tersebut dan mengekspresikan dalam persamaan atau pertidaksamaan dan merupakan hubungan linier dari variabel keputusan yang mencerminkan keterbatasan sumberdaya masalah itu.

Bentuk umum program linier adalah :

Memaksimumkan/meminimumkan $Z = \sum_{j=1}^{n} C_j X_j$

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, di mana $j=1,2,\ldots,n$
berarti di sini terdapat n variabel keputusan.

Z :nilai fungsi tujuan

C_j :sumbangan per unit kegiatan j,
Untuk masalah maksimisasi C_j, menunjukkan keuntungan atau penerimaan per unit, sementara dalam kasus minimisasi menunjukkan biaya per unit.

b_i :jumlah sumberdaya ke i $(i=1,2,\ldots,m)$,
berarti terdapat m jenis sumberdaya.

a_{ij}:banyaknya sumberdaya i yang dikonsumsi sumberdaya j.

2.1.2 Metode Simpleks

Dalam menyelesaikan masalah program linier yang melibatkan lebih dari dua variabel menjadi tak praktis atau tidak mungkin, karena sulitnya menggambarkan grafik berdimensi banyak. Dalam keadaan ini dibutuhkan metode solusi yang lebih umum. Metode umum itu dikenal dengan nama algoritma simpleks. Metode ini menyelesaikan masalah program linier melalui perhitungan-ulang (iteration) dimana langkah-langkah perhitungan yang sama diulang berkali-kali hingga solusi optimum dicapai.

Dalam menggunakan metode simpleks untuk menyelesaikan masalah-masalah program linier, model program linier harus diubah ke dalam suatu bentuk umum yang dinamakan "bentuk baku" (standard form). Ciri-ciri dari bentuk baku model program linier adalah :

1. Semua kendala berupa persamaan dengan sisi kanan non negatif.
2. Semua variabel non negatif.
3. Fungsi tujuan dapat maksimum atau minimum.

Kendala

1. Suatu kendala jenis ≤ atau ≥ dapat diubah menjadi suatu persamaan dengan menambahkan suatu variabel slack atau mengurangkan suatu variabel surplus ke sisi kiri kendala.

   Contoh :

   a. Pada kendala $X_1 + X_2 \leq 15$ ditambahkan suatu slack $X_1 \geq 0$ pada sisi kiri untuk mendapatkan persamaan $X_1 + X_2 + S_1 = 15$. Jika kendala menunjukkan keterbatasan penggunaan suatu sumberdaya, S_1 akan menunjukkan slack atau jumlah sumberdaya yang tak digunakan.
   b. Pada kendala $3X_1 + 2X_2 - 3X_3 \geq 5$ dikurangkan suatu variabel surplus $S_2 \geq 0$ 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 $5X_1 - X_2 \geq 25$.

Variabel

Sebagian atau semua variabel dikatakan unresstricted jika mereka dapat memiliki nilai negatif maupun positif. Variabel unrestricted dapat diekspresikan dalam dua variabel non negatif dengan menggunakan substitusi $X_j = X_j' - X''$

dimana X_j = variabel unresstricted dan

$X_j', X_j'' \geq 0$

Fungsi Tujuan

Meskipun model program linier 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, dan adalah sama pada kedua kasus. Perbedaannya hanya pada nilai fungsi tujuan, meski besar angka sama, tetapi tandanya berlawanan.

Contoh : Ubahlah model program linier berikut ke dalam bentuk baku.

Memaksimumkan $Z = 9X_1 + 18X_2$

$$6X_1 + 3X_2 \leq 18$$

$$2X_1 + 2X_2 \leq 16$$

X_1 unrestricted

$$X_2 \geq 0$$

Bentuk bakunya adalah :

Memaksimumkan $Z = 9X_1' - 9X'' + 18X_2 + 0S_1 + 0S_2$

Dengan syarat

$$\begin{aligned} 6X_1' - 6X'' + 3X_2 + S_1 \quad &= 18 \\ 2X_1' - X'' + 2X_2 \quad + S_2 &= 16 \\ X_1',\ X'',\ X_2,\ S_1,\ S_2 &\geq 0 \end{aligned}$$

Metode dan tabel simpleks

Dalam penyelesaian masalah program linier 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 (sesuai), 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 terhenti dan solusi optimum diperoleh.

Model Program Linier :

Maks./min. $\quad Z = C_1X_1 + C_2X_2 + \dots + C_nX_n$

berdasarkan:

$$\begin{aligned} a_{11}X_1 + a_{12}X_2 + \dots + a_{1n}X_n &= b_1 \\ a_{21}X_1 + a_{22}X_2 + \dots + a_{2n}X_n &= b_2 \\ &\vdots \\ a_{m1}X_1 + a_{m2}X_2 + \dots + a_{mn}X_n &= b_m \\ X_i \geq 0 \quad (i = 1,2,\dots,n) \end{aligned}$$

Jika didefinisikan :

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \quad X = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{pmatrix} \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

Maka pembatas dari model tersebut dapat dituliskan ke dalam bentuk sistem persamaan AX = b.

Definisi :

1. Solusi basis

   Solusi basis untuk AX = b adalah solusi dimana terdapat sebanyak-banyaknya m variabel berharga bukan nol, yang disebut variabel basis (Basis Variable) disingkat BV.

2. Solusi fisibel

Jika seluruh variabel pada suatu solusi basis berharga non negatif, maka solusi itu disebut solusi basis fisibel (BFS).

Algoritma simpleks untuk penyelesaian kasus minimal/maksimal :

1. Mengkonveksikan semua pertidaksamaan dan persamaan(fungsi pembatas) ke dalam bentuk standar.
2. Lakukan langkah-langkah iterasi sampai kondisi berikut terpenuhi, jika :
   - Baris fungsi tujuan masih ada variabel dengan koefisien negatif, atau nol, maka penyelesaian sudah optimal untuk kasus minimal.
   - Baris fungsi tujuan semua variabel dengan koefisien non negatif (positif atau nol) maka penyelesaian sudah optimal untuk kasus maksimal.
3. Langkah-langkah iterasi :
   a. Tentukan kolom kunci, dengan cara memilih harga baris fungsi tujuan yang paling negatif untuk kasus maksimal, memilih harga baris fungsi tujuan yang paling positif untuk kasus minimal.
   b. Tentukan baris kunci dengan cara memilih rasio (perbandingan b_i dengan koefisien pada kolom kunci) yang paling minimal.
   c. Tentukan elemen kunci yaitu elemen pada baris kunci kolom kunci.
   d. Lakukan operasi elementer pada matriks simpleks sedemikian rupa sehingga elemen kunci sama dengan satu sedangkan elemen baris lain pada kolom kunci sama dengan nol.
   e. Lanjutkan sampai didapatkan hasil yang optimal.

BV	Kegiatan (perubah) $x_1, x_2, \ldots, x_n$	Solusi	R_i
Z_j	Koefisien fungsi tujuan		
	Koefisien subtitusi (input – output)		

Contoh 1: Untuk meminimalkan fungsi W di mana :

$W_{min} = 2X_1 - 3X_2$

Dengan pembatas : $X_1 + X_2 \leq 4$

$X_1 - X_2 \leq 6$

$X_1, X_2 \geq 0$

Bentuk standar: $X_1 + X_2 + X_3 = 4$

$X_1 - X_2 + X_4 = 6$

$W - 2X_1 + 3X_2 - 0X_3 - 0X_4 = 0$

BV	W	X_1	X_2	X_3	X_4	Solusi	R_i
W	1	-2	3	0	0	0	
X_3	0	1	1	1	0	4	4/1
X_4	0	1	-1	0	1	6	6/-1

BV	W	X_1	X_2	X_3	X_4	Solusi
W	1	-5	0	-3	0	-12
X_2	0	1	1	1	0	4
X_4	0	2	0	1	1	10

Hasil sudah optimal karena koefisien fungsi tujuan berharga non positif.

Jadi $W = -12$

$X_1 = 0$ $X_3 = 0$

$X_2 = 4$ $X_4 = 10$

Contoh 2: Untuk memaksimalkan fungsi Z di mana :

$$Z_{max} = 3X_1+2X_2$$

Dengan pembatas : $X_1 + X_2 \leq 15$

$2X_1 + X_2 \leq 28$

$X_1 + 2X_2 \leq 20$

$X_1 , X_2 \geq 0$

Bentuk standar: $X_1 + X_2 + S_1 = 15$

$2X_1 + X_2 + S_2 = 28$

$X_1 + 2X_2 + S_3 = 20$

$Z - 3X_1 + 2X_2 - 0S_1 - 0S_2 - 0S_3 = 0$

BV	z	X_1	X_2	s_1	s_2	s_3	Solusi	R_i
Z	1	-3	2	0	0	0	0	
s_1	0	1	1	1	0	0	15	15/1
s_2	0	2	1	0	1	0	28	28/2
s_3	0	1	2	0	0	1	20	20/1

BV	z	X_1	X_2	s_1	s_2	s_3	Solusi	R_i
Z	1	0	-1/2	0	3/2	0	42	
s_1	0	0	1/2	1	-1/2	0	1	2
X_1	0	1	1/2	0	1/2	0	14	28
S_3	0	0	3/2	0	-1/2	1	6	4

BV	z	X_1	X_2	s_1	s_2	s_3	Solusi
Z	1	0	0	1	1	0	42
X_2	0	0	1	2	-1	0	2
X_1	0	1	0	-1	1	0	13
S_3	0	0	0	-3	1	1	3

Hasil sudah optimal karena koefisien fungsi tujuan berharga non negatif

Jadi $Z = 43$

$X_1 = 13$ $S_1 = 0$ $S_3 = 3$

$X_2 = 2$ $S_2 = 0$

2.2 Linier Goal Programming

Linier Goal Programming (LGP) merupakan perluasan dari Program Linier yang diperkenalkan oleh Channes dan Cooper pada tahun 60-an. Goal Programming merupakan salah satu model matematika untuk masalah optimasi multi kriteria. Goal Programming akan menyusun skala prioritas sasaran untuk setiap tujuan yang ada. Dengan menggunakan Goal Programming maka akan dapat memenuhi target yang ditetapkan sesuai prioritasnya.

2.2.1 Terminologi Linier Goal Programming

- Variabel keputusan : seperangkat variabel yang tidak diketahui (dalam model LGP dilambangkan dengan X_j, dimana $j = 1,2,\ldots,n$) yang akan dicari nilainya.
- Variabel simpangan : variabel-variabel yang menunjukkan kemungkinan penyimpangan negatif atau positif dari suatu nilai sisi kanan (dilambangkan dengan d_i^- atau d_i^+, dimana $i = 1,2,\ldots,m$ dan m adalah banyaknya kendala tujuan dalam model).
- Nilai sisi kanan : nilai-nilai yang biasanya menunjukkan ketersediaan sumberdaya (dilambangkan dengan b_i) yang akan ditentukan kekurangan atau kelebihan penggunaannya.

- Kendala tujuan : suatu tujuan yang diekspresikan dalam persamaan matemetik dengan memasukkan variabel simpangan.
- Tujuan : keinginan untuk meminimumkan angka penyimpangan dari suatu nilai sisi kanan pada suatu kendala tujuan tertentu.
- Bobot : timbangan matematik (pembobotan) yang diekspresikan dengan angka kardinal (dilambangkan dengan W_{ki} dimana $k = 1,2,\ldots,K$; $i = 1,2,\ldots,m$) dan digunakan untuk membedakan variabel simpangan i dalam suatu tingkat prioritas K.
- Skala Prioritas : suatu sistem urutan (dilambangkan dengan P_k, dimana $k = 1,2,\ldots,K$ dan K menunjukkan banyaknya tujuan dalam model) yang memungkinkan tujuan-tujuan disusun secara ordinal dalam model LGP. Sistem urutan itu menempatkan tujuan-tujuan dalam susunan dengan hubungan sebagai berikut:

  $P_1 > P_2 >>> P_k$

  P_1 = tujuan paling penting.

  P_2 = tujuan kurang penting dan seterusnya.

2.2.2 Unsur-unsur Linier Goal Programming

Setiap model LGP paling sedikit terdiri dari tiga komponen, yaitu sebuah fungsi tujuan, kendala-kendala tujuan, dan kendala non negatif.

♦ Fungsi Tujuan :

$$\text{Meminimumkan } Z = \sum_{i=1}^{m} W_{ki} P_k (d_i^- + d_i^+), \; \forall k = 1,2,\ldots,K$$

Dalam fungsi tujuan di atas, tujuan-tujuan diurutkan dan variabel simpangan pada setiap tingkat prioritas dibedakan dengan menggunakan bobot yang berlainan W_{ki}.

♦ Kendala Tujuan :

Bentuk persamaan kendala tujuan secara umum :

$$a_{ij}x_j \ (\leq ,=,\geq) b_i$$

dan kemudian dikonversikan secara umum menjadi :

$$a_{ij}x_j + d_i^- - d_i^+ = b_i$$

Untuk menentukan variabel simpangan yang akan diminimumkan ,harus dilihat terlebih dahulu hubungannya dengan fungsi tujuan yang diinginkan. Ada enam jenis kendala tujuan seperti terlihat pada tabel 1.

Tabel 1. Jenis-jenis Kendala Tujuan

Persamaan ke	Kendala Tujuan	Variabel Simpangan Dalam Fungsi Tujuan	Kemungkinan Simpangan	Penggunaan Nilai RHS yang diinginkan
1	$A_{ij}X_j + d_i^- = b_i$	d_i^-	Negatif	$= b_i$
2	$A_{ij}X_j - d_i^+ = b_i$	d_i^+	Positif	$= b_i$
3	$A_{ij}X_j + d_i^- - d_i^+ = b_i$	d_i^-	Negatif / positif	b_i atau lebih
4	$A_{ij}X_j + d_i^- - d_i^+ = bi$	d_i^+	Negatif / positif	B_i atau kurang
5	$A_{ij}X_j + d_i^- - d_i^+ = b_i$	d_i^- dan d_i^+	negatif / positif	= bi
6	$A_{ij}X_j - d_i^+ = b_i$	d_i^+ (artificial)	tidak ada	Pas $= b_i$

Dari tabel di atas dapat dijelaskan bahwa persamaan ke-1 maknanya serupa dengan kendala pertidaksamaan ≤ dalam masalah program linier maksimisasi. Persamaan ke-2 maknanya serupa dengan kendala pertidaksamaan ≥ dalam masalah program linier minimisasi. Persamaan ke-3, ke-4, dan ke-5 semuanya memperbolehkan penyimpangan dua arah, tetapi persamaan ke-5 mencari penggunaan sumberdaya yang diinginkan sama dengan b_i. Ini serupa dengan kendala persamaaan dalam LP, tetapi tidak menempel pada solusi karena dimungkinkan adanya penyimpangan negatif dan positif. Jika kendala persamaan diperlukan dalam perumusan model LGP, maka dapat dimasukkan dengan menempatkan sebuah variabel artificial d_i^+ seperti pada persamaan ke-6.

- Kendala struktual : kendala-kendala lingkungan yang tidak berhubungan langsung dengan tujuan-tujuan masalah yang dihadapi. Variabel simpangan tidak dimasukkan dalam kendala, karena kendala ini tidak merupakan fungsi tujuan.

2.2.3 Perumusan Masalah Linier Goal Programming

Perumusan suatu masalah LGP sangat mirip dengan perumusan sebuah masalah LP. Penjelasan variabel keputusan X_j, koefisien teknologi a_{ij}, dan nilai sisi kanan b_i diperlukan baik pada LP maupun LGP. Langkah-langkah perumusan LGP meliputi beberapa tahap:

1. Tentukan variabel keputusan.
   Di sini kuncinya adalah menyatukan dengan jelas variabel keputusan yang tak diketahui. Makin tepat definisi akan makin mudah pekerjaan permodelan yang lain.
2. Nyatakan sistem kendala.
   Kuncinya pertama adalah menentukan nilai sisi kanan dan kemudian menentukan koefisien teknologi yang cocok dan variabel keputusan yang diikutsertakan dalam kendala. Juga perhatikan jenis penyimpangan yang diperbolehkan dari nilai RHS. Jika penyimpangan

diperbolehkan dalam dua arah, tempatkan kedua variabel simpangan pada kendala itu. Jika penyimpangan hanya diperbolehkan pada satu arah, tempatkan hanya satu variabel simpangan yang tepat pada kendala yang bersangkutan.

3. Tentukan prioritas utama.
   Kuncinya disini adalah membuat urutan tujuan-tujuan. Biasanya urutan tujuan menyatakan preferensi individu. Jika persoalannya tidak memiliki urutan tujuan, lewati langkah ini dan kemudian ke langkah berikutnya.
4. Menentukan bobot.
   Disini kuncinya adalah membuat urutan di dalam suatu tujuan tertentu. Jika tidak diperlukan lewati langkah ini.
5. Nyatakan fungsi tujuan.
   Disini kuncinya adalah memilih variabel simpangan yang benar untuk dimasukkan dalam fungsi tujuan. Gunakan tabel 1 untuk meyakinkan penggunaan nilai RHS yang diinginkan adalah konsisten dengan keperluan persoalan. Kedua, tambahkan prioritas dan bobot yang tepat jika diperlukan.

Model Linier Goal Programming

Meminimalkan : $$f = \sum_{i=1}^{P} (P_k W_{i,k}^{+} d_i^{+} + P_s W_{i,s}^{-} d_i^{-})$$

Kendala : $$\sum_{j=1}^{n} m_{ij} X_j + d_i^{-} - d_i^{+} = g_i \qquad i=1,\dots,s$$

$$\sum_{j=1}^{n} a_{ij} X_j \le b_i \qquad i = s+1,\dots,s+m$$

$$X_j, d_i^-, d_i^+ \geq 0 \qquad j=1,\ldots,n \ ; \ i=1,\ldots,s$$

dimana : d_i^-, d_i^+ = simpangan pada tujuan ke-i

P_k, P_s = skala prioritas

$W_{i,k}$; $W_{i,s}$ = bobot relatif pada variabel simpangan dan terdapat sejumlah s tujuan, m pembatas bukan tujuan, serta sejumlah n variabel keputusan (X_j).

2.2.4 Metode Modifikasi Simpleks untuk Linier Goal Programming

Untuk menyesuaikan model Linier Goal Programming dengan metode simpleks, terlebih dahulu kita buat tabel simpleks.

Tabel 2.

Struktur tabel simpleks model Linier Goal Programming :

C_j			0 0 ... 0 P1 ... 0 Pn 0 0		
I	C_b	X_b Basis	$X_1, \ldots, X_n$ d_i^+ d_i^- d_n^+ d_n^- S_1 S_2	Solusi b_i	R_i
1 2 ⋮ m		Perubah-perubah yang menjadi basis	Koefisien Substitusi (input – output)		Nilai perbandingan antara solusi dengan kolom kunci
P_n		Z_j $C_j - Z_j$	Evaluasi fungsi tujuan		
⋮					
P_1		Z_j $C_j - Z_j$	Evaluasi fungsi tujuan		

di mana :

C_j = koefisien fungsi tujuan.

C_b = koefisien peubah pada fungsi tujuan yang terpilih sebagai solusi yang layak.

X_b = peubah yang menjadi basis.

b_i = koefisien sebelah kanan dari pembatas.

Z_j = matriks kolom untuk evaluasi fungsi tujuan.

Selanjutnya cara penyelesaian LGP hampir sama dengan Program Linier, namun harus memenuhi hal-hal berikut :

1. Kriteria variabel yang masuk (memilih variabel yang mempunyai nilai C_j-Z_j yang negatif terbesar sebagai variabel yang masuk sebagai basis untuk kasus minimal) adalah tidak lebih dari yang diperlihatkan sebagai sebuah baris (single) terbawah pada tabel simpleks. Malahan secara terpisah terdapat baris Z_j dan C_j - Z_j untuk setiap prioritas $P_1,\ldots,P_n$. Ini diperlukan karena tidak dapat menambahkan variabel simpangan tujuan P_1 ke variabel simpangan tujuan ke P_2 dan seterusnya, karena ketiga tujuan tersebut berbeda. Nilai Z_j sesuai dengan baris masing-masing menunjukkan sumbangan pada fungsi sasaran dari simpangan pada tingkat prioritas.
2. Nilai C_j - Z_j untuk suatu kolom ditunjukkan dalam baris-baris prioritas pada tabel di sebelah bawah.
3. Dalam pemilihan variabel yang masuk sebagai basis, mulai dengan prioritas yang tertinggi P_1, dan variabel yang menjadi basis adalah variabel yang mempunyai nilai negatif terbesar pada baris C_j - Z_j. Sedangkan kolom yang mempunyai nilai C_j - Z_j yang negatif, disebut sebagai kolom kunci. Jika pada

baris P_1 sudah tidak terdapat nilai $C_j - Z_j$ yang negatif, maka bergerak ke prioritas berikutnya P_2, dan sekaligus menguji $C_j - Z_j$ pada baris tersebut. Demikian seterusnya, jika sudah tidak terdapat nilai negatif pada baris tersebut, maka penyelesaian optimal sudah tercapai.

4. Dalam pemilihan variabel yang akan dikeluarkan dari baris, kriteria yang biasa digunakan dalam PL dapat dipakai. Sesuai dengan hal tersebut dapat menghitung perbandingan antara nilai pada kolom penyelesaian dengan nilai pada kolom kunci yang sebaris. Nilai perbandingan yang paling kecil dan positif dijadikan baris kunci dan digunakan untuk menentukan penggantian variabel basis lama dengan yang baru. Elemen yang merupakan perpotongan antara kolom kunci dan baris kunci disebut elemen kunci.
5. Jika menemukan sebuah nilai $C_j - Z_j$ negatif pada salah satu baris prioritas sedangkan pada baris prioritas di bawahnya (lebih tinggi) bernilai positif, maka tidak akan dipermasalahkan lebih jauh. Hal ini dilakukan karena nilai positif berarti bahwa simpangan dari tujuan/prioritas yang lebih rendah akan meningkat jika dimasukkan variabel tersebut ke dalam basis. Hal ini harus dihindari, karena tidak akan memperbaiki keadaan, dan pada kenyataannya akan lebih memperburuk.

Contoh :

Perusahaan Pegasus membuat tiga jenis model sepatu, yaitu : model Mercury, model Flying Feet, model Swift Runner.

Model Sepatu	Tenaga Kerja (jam/pasang)	Bahan (pon/pasang)	Keuntungan ($/pasang)
Mercury	3	2	10
Flying feet	2	1	7
Swift Runner	1	1	5

Perusahaan Pegasus sekarang ini mempunyai kapasitas produksi 200 jam tenaga kerja sehari dan setiap hari tersedia 100 pon bahan. Manajemen perusahaan pegasus mengembangkan tujuan yang akan diwujudkan, dengan mempertimbangkan kepentingan bagi perusahaan.

Tujuannya yaitu :

1. Menghasilkan 50 pasang Swift Runner tiap hari (P_1).
2. Mendapatkan keuntungan paling sedikit $ 750 sehari (P_2).
3. Menghindari tenaga kerja yang berhenti (P_3).

Membentuk kendala tujuan :

I. **Prioritas P_1** : memproduksi paling sedikit 50 pasang sepatu model Swift Runner dalam sehari.

$X_3 + d_3^- - d_3^+ = 50$

d_3^+ = kelebihan produksi Swift Runner dari tujuan.

d_3^- = kekurangan produksi Swift Runner dari tujuan.

II. **Prioritas P_2** : mendapatkan keuntungan paling sedikit $ 750 sehari.

$10X_1 + 7X_2 + 5X_3 - d_1^+ + d_1^- = 750$

d_1^+ = kelebihan keuntungan tiap hari dari tujuan keuntungan sebesar $ 750

d_1^- = kekurangan keuntungan tiap hari dari tujuan keuntungan sebesar $ 750

III. **Prioritas P_3** : menghindari tenaga kerja yang berhenti, ini berarti sama dengan menggunakan waktu kerja sepenuhnya, tidak ada jam tenaga kerja yang tidak terpakai.

$3X_1 + 2X_2 + X_1 + d_2^- - d_2^+ = 200$

d_2^- = jumlah tenaga kerja yang kurang dari tujuan yang akan dicapai.

d_2^+ = kekurangan jumlah waktu tenaga kerja dari tujuan yang akan dicapai.

Kendala Struktural = $2X_1 + X_2 + X_3 \leq 100$

Tujuan	Prioritas	Variabel Deviasi yang digunakan
1. Memproduksi paling sedikit 50 pasang "Swift Runner"	P_1	d_3^-
2. Keuntungan \$ 750 perhari	P_2	d_1^-
3. Menghindari tenaga kerja yang tidak bekerja	P_3	d_2^-

Model Linier Goal Programming :

Meminimalkan $Z = P_1\, d_3^- + P_2\, d_1^- + P_3\, d_2^-$

Kendala : $10X_1 + 7X_2 + 5X_3 - d_1^+ + d_1^- = 750$

$$3X_1 + 2X_2 + X_3 - d_2^+ + d_2^- = 200$$

$$X_3 - d_3^+ + d_3^- = 50$$

$$2X_1 + X_2 + X_3 + S_1 = 100$$

$$X_1, X_2, X_3, d_1^+, d_1^-, d_2^+, d_2^-, d_3^+, d_3^-, S_1 \geq 0$$

Mencari $C_j - Z_j$:

Untuk kolom

$X_1 = 0 - (10.P_2 + 3.P_3 + 0.P_1 + 2.0) = -10.P_2 - 3.P_3$

$X_2 = 0 - (7.P_2 + 2.P_3 + 0.P_1 + 1.0) = -7.P_2 - 2.P_3$

$X_3 = 0 - (5.P_2 + 1.P_3 + 1.P_1 + 1.0) = -5.P_2 - P_3 - P_1$

$d_1^+ = 0 - (-1.P_2) = P_2$

$d_1^- = P_2 - (1.P_2) = 0$

$d_2^+ = 0 - (-1.P_3) = P_3$

$d_2^- = P_3 - (1.P_3) = 0$

$d_3^+ = 0 - (-1.P_1) = P_1$

$d_3^- = P_1 - (1.P_1) = 0$

$S_1 = 0 - (1.0) = 0$

Perhitungan Simpleks.

Iterasi 1

	C_j	0	0	0	0	P_2	0	P_3	0	P_1	0		
C_b	Basis	X_1	X_2	X_3	d_1^+	d_1^-	d_2^+	d_2^-	d_3^+	d_3^-	S_1	Solusi	R_i
P_2	d_1^-	10	7	5	-1	1	0	0	0	0	0	750	750/5=150
P_3	d_2^-	3	2	1	0	0	-1	1	0	0	0	200	200/1=200
P_1	d_3^-	0	0	1	0	0	0	0	-1	1	0	50	50/1 = 50
0	S_1	2	1	1	0	0	0	0	0	0	1	100	100/1=100
P_3	Z_j	3	2	1	0	0	-1	1	0	0	0	200	
	$C_j - Z_j$	-3	-2	-1	0	0	1	0	0	0	0		
P_2	Z_j	10	7	5	-1	1	0	0	0	0	0	750	
	$C_j - Z_j$	-10	-7	-5	1	0	0	0	0	0	0		
P_1	Z_j	0	0	1	0	0	0	0	-1	1	0	50	
	$C_j - Z_j$	0	0	-1	0	0	0	0	1	0	0		

Iterasi 2

	C_j	0	0	0	0	P_2	0	P_3	0	P_1	0		
C_b	Basis	X_1	X_2	X_3	d_1^+	d_1^-	d_2^+	d_2^-	d_3^+	d_3^-	S_1	Solusi	R_i
P_2	d_1^-	10	7	0	-1	1	0	0	5	-5	0	500	500/10=50
P_3	d_2^-	3	2	0	0	0	-1	1	1	-1	0	150	150/3=50
0	X_3	0	0	1	0	0	0	0	-1	1	0	50	——
0	S_1	2	1	0	0	0	0	0	1	-1	1	50	50/2=25
P_3	Z_j	3	2	0	0	0	-1	1	1	-1	0	150	
	$C_j - Z_j$	-3	-2	0	0	0	1	0	-1	1	0		
P_2	Z_j	10	7	0	-1	1	0	0	5	-5	0	500	
	$C_j - Z_j$	-10	-7	0	1	0	0	0	-5	5	0		
P_1	Z_j	0	0	0	0	0	0	0	0	0	0	0	
	$C_j - Z_j$	0	0	0	0	0	0	0	0	1	0		

Iterasi 3

C_b	Basis	X_1	X_2	X_3	d_1^+	d_1^-	d_2^+	d_2^-	d_3^+	d_3^-	S_1	Solusi	R_i
	C_j	0	0	0	0	P_2	0	P_3	0	P_1	0		
P_2	d_1^-	0	2	0	-1	1	0	0	0	0	-5	250	250/2=125
P_3	d_2^-	0	½	0	0	0	-1	1	½	-½	-3/2	75	75/½=150
0	X_3	0	0	1	0	0	0	0	-1	1	0	50	——
0	X_1	1	½	0	0	0	0	0	½	-½	½	25	25/½=50
P_3	Z_j	0	½	0	0	0	-1	1	½	-½	-3/2	75	
	$C_j - Z_j$	0	-½	0	0	0	1	0	-½	½	3/2		
P_2	Z_j	0	2	0	-1	1	0	0	0	0	-5	250	
	$C_j - Z_j$	0	-2	0	1	0	0	0	0	0	5		
P_1	Z_j	0	0	0	0	0	0	0	0	0	0	0	
	$C_j - Z_j$	0	0	0	0	0	0	0	0	1	0		

Iterasi 4

C_b	Basis	X_1	X_2	X_3	d_1^+	d_1^-	d_2^+	d_2^-	d_3^+	d_3^-	S_1	Solusi
	C_j	0	0	0	0	P_2	0	P_3	0	P_1	0	
P_2	d_1^-	-4	0	0	-1	1	0	0	-2	2	-7	150
P_3	d_2^-	-1	0	0	0	0	-1	1	-1	1	-2	50
0	X_3	0	0	1	0	0	0	0	-1	1	0	50
0	X_2	2	1	0	0	0	0	0	1	-1	1	50
P_3	Z_j	-1	0	0	0	0	-1	1	-1	1	-2	50
	$C_j - Z_j$	1	0	0	0	0	1	0	1	-1	2	
P_2	Z_j	-4	0	0	-1	1	0	0	-2	2	-7	150
	$C_j - Z_j$	4	0	0	1	0	0	0	2	-2	7	
P_1	Z_j	0	0	0	0	0	0	0	0	0	0	0
	$C_j - Z_j$	0	0	0	0	0	0	0	0	1	0	

Tabel iterasi ke-4 merupakan iterasi terakhir karena nilai C_j - Z_j telah bernilai positif semua. Jadi hasil yang diperoleh :

1. Produksi Swift Runner sehari = 50 pasang terpenuhi, karena d_3^- = 0.

2. Keuntungan harian sebesar $ 750 tidak terpenuhi, karena d_1^- = 150 atau keuntungan harian hanya sebesar $ 600.
3. Jam tenaga kerja sebesar 200 jam sehari tidak terpenuhi, karena d_2^- = 50 atau jam tenaga kerja hanya 150 sehari.

2.3 Manajemen Persediaan

Salah satu unsur dalam operasi pabrik yang sangat penting adalah manajemen persediaan. Yang dimaksud dengan persediaan adalah tersedianya barang/komponen yang disediakan untuk memperlancar proses produksi, agar perusahaan tidak mengalami kekurangan bahan. Jenis-jenis persediaan meliputi:

a. Batch Stock : persediaan yang diadakan karena membeli barang dalam jumlah yang melebihi jumlah yang dibutuhkan saat itu.
b. Fluctuation Stock : persediaan yang diadakan untuk menghindari fluktuasi permintaan konsumen yang tidak dapat diramalkan.
c. Antiapation Stock : persediaan yang diadakan untuk mengantisipasi permintaan yang dapat diramalkan.

Di dalam pembuatan model matematika untuk menajemen persediaan, faktor yang sangat dominan adalah faktor biaya. Secara umum terdapat empat kategori biaya persediaan, yaitu :

a. Biaya pembelian/produksi.
b. Biaya pengadaan : biaya pemeriksaan, biaya pemesanan, biaya penerimaan, biaya dokumen, biaya telepon, dan lain-lain.

c. Biaya penyimpanan : bunga modal, sewa gedung, asuransi, pajak, dan lain-lain.
d. Biaya Stock Out : biaya yang disebabkan tidak terpenuhinya kebutuhan pelanggan.

Selanjutnya manajemen persediaan yang akan dibahas adalah manajemen persediaan untuk suatu perusahaan di mana perusahaan tersebut memiliki keterbatasan dana pembelian dan gudang penyimpanan. Sehingga perusahaan tersebut harus menentukan tingkat persediaan aman setiap produknya.