BAB II

LANDASAN TEORI

2.1 PENYELIDIKAN OPERASIONAL

Pola dasar penerapan penyelidikan operasional terhadap suatu masalah dapat dipisahkan menjadi beberapa tahap, yaitu : [SRI,91]

1. Merumuskan masalah.
2. Pembentukan model.
3. Mencari penyelesaian masalah.
4. Validasi model.
5. Penerapan hasil akhir.

Pembentukan model yang cocok hanyalah salah satu tahap dari aplikasi penyelidikan operasional. Model adalah abstraksi atau penyederhanaan realitas sistem yang kompleks dimana hanya komponen-komponen yang relevan atau faktor-faktor yang dominan dari masalah yang dianalisa diikut sertakan. Model menunjukan hubungan-hubungan (langsung dan tidak langsung) dari aksi dan reaksi dalam pengertian sebab akibat. Karena sebuah model adalah suatu abstraksi realitas, model akan tampak kurang kompleks jika dibandingkan realitas itu sendiri. Model itu untuk menjadi lengkap harus mencerminkan semua aspek-aspek realitas yang sedang diteliti. [SRI,91]

Jenis model dalam penyelidikan operasional terdiri atas tiga jenis, yaitu : [TAH,96]

1. Pemrograman Matematis, terdiri atas Pemrograman Linear Metode Simpleks, Pemrograman Linear Model Transportasi, Pemrograman Linear Integer, Model Jaringan, dan Pemrograman Dinamis (Multitahap).
2. Model-model Probabilistik, terdiri atas Teori Keputusan dan Permainan, Penjadwalan Proyek dengan PERT-CPM, Model-model Sediaan, Model-model Antrian, Pemodelan Similasi dengan SIMNET II, dan Proses Keputusan Markov.
3. Pemrograman Nonlinier, terdiri atas Teori Optimasi Klasik dan Algoritma Pemrograman Nonlinier.

2.2 PEMROGRAMAN LINEAR

2.2.1 Arti Pemrograman Linear

Salah satu keputusan manajerial yang sangat penting ialah berkenaan dengan penggunaan sumber secara efisien atau alokasi sumber-sumber yang terbatas (tenaga kerja terampil, bahan mentah , modal) untuk mencapai tujuan yang diinginkan *(desired objective)*. Dalam keadaan sumber yang terbatas harus dicapai suatu hasil yang optimum. Dengan perkataan lain bagaimana caranya agar dengan masukan *(input)* yang serba terbatas dapat dicapai hasil kerja yaitu keluaran *(output)* berupa produksi barang atau jasa yang optimum. Metode analisis yang paling bagus untuk menyelesaikan persoalan alokasi sumber ialah metode pemrograman linear.

Pemrograman Linear ialah salah satu teknik dari Riset Operasi untuk memecahkan masalah optimasi (maksimisasi atau minimisasi) dengan menggunakan persamaan dan pertidaksamaan linear dalam rangka untuk mencari pemecahan yang optimum dengan memperhatikan pembatasan-pembatasan yang ada [SUP,88]. Pokok pikiran yang utama dalam menggunakan program linear ialah merumuskan masalah dengan jelas dengan menggunakan sejumlah informasi yang tersedia. Sesudah masalah terumuskan dengan baik, maka langkah berikutnya ialah menerjemahkan masalah ini ke dalam bentuk model matematika selanjutnya menentukan fungsi objektif yang linear yang harus dioptimunkan dengan pembatasan-pembatasan yang harus dinyatakan dalam ketidaksamaan yang linear. Jawaban yang ditemukan dari hasil perhitungan lebih mudah dinilai atau dievaluasi kebenarannya. Jawaban yang lebih ampuh (terbaik) akan ditetapkan sebagai keputusan akhir dan siap untuk dilaksakan. [SIA,87]

2.2.2 Model Pemrograman Linear

Model matematis perumusan masalah umum pengalokasian sumber daya untuk berbagai kegiatan, disebut sebagai model pemrogramman linear. Model pemrograman linear ini merupakan bentuk susunan dari dalam menyajikan masalah-masalah yang akan dipecahkan dengan teknik pemrograman linear. Dalam model pemograman linear dikenal dua macam fungsi, yaitu fungsi tujuan *(objective fungtion)* dan fungsi-fungsi batasan *(constraint functions)*. Fungsi tujuan adalah fungsi yang menggambarkan tujuan/sasaran di dalam permasalahan pemograman linear yang berkaitan dengan pengaturan secara optimal sumber daya-sumber daya, untuk memperoleh keuntungan maksimal atau biaya minimal. Pada umumnya nilai yang akan dioptimalkan dinyatakan sebagai Z. Sedang fungsi batasan merupakan bentuk penyajian secara matematis batasan-batasan kapasitas yang tersedia yang akan dialokasikan secara optimal ke berbagai kegiatan.

Agar memudahkan pembahasan model pemrograman linear digunakan simbol-simbol sebagai berikut :

m = macam batasan-batasan sumber atau fasilitas yang tersedia.

n = macam kegiatan-kegiatan yang mengunakan sumber atau fasilitas tersebut.

i = nomor setiap macam sumber atau fasilitas yang tersedia ($i = 1,2,...,m$).

j = nomor setiap macam kegiatan yang mengunakan sumber atau fasilitas yang tersedia ($j = 1,2,...,n$).

x_j = tingkat kegiatan ke j ($j = 1,2,...,n$).

a_{ij} = banyaknya sumber i yang diperlukan untuk menghasilkan setiap unit keluaran (output) kegiatan j ($i = 1,2,...,n$ dan $j = 1,2,...,m$).

b_i = banyaknya sumber atau fasilitas i yang tersedia untuk dialokasikan ke setiap unit kegiatan ($i = 1,2,...,n$).

Z = nilai yang dioptimalkan (maksimum atau minimum).

C_j = kenaikan nilai Z apabila ada pertambahan tingkat kegiatan (x_j) dengan satu satuan (unit); atau merupakan *sumbangan* setiap satuan keluaran kegiatan j terhadap nilai Z.

Keseluruhan simbol-simbol diatas selanjutnya disusun kedalam bentuk tabel standar pemograman linear.

Tabel 2.1. Data untuk model pemrograman linear

Kegiatan / Sumber	Pemakaian sumber per unit kegiatan (keluaran) 1	2	3	....	n	Kapasitas sumber
1	a_{11}	a_{12}	a_{13}	....	a_{1n}	b_1
2	a_{21}	a_{22}	a_{23}	....	a_{2n}	b_2
3	a_{31}	a_{32}	a_{33}	....	a_{3n}	b_3
:	:	:	:		:	:
:	:	:	:		:	:
m	a_{m1}	a_{m2}	a_{m3}	....	a_{nm}	b_m
ΔZ pertambahan tiap unit	C_1	C_2	C_3	....	C_n	
Tingkat kegiatan	X_1	X_2	X_3	....	X_n	

Atas dasar tabel diatas kemudian dapat disusun suatu model matematis yang digunakan untuk mengemukakan suatu permasalahan pemograman linear sebagai berikut:

Fungsi tujuan :

$$\text{Maksimumkan } Z = C_1X_1 + C_2X_2 + C_3X_3 + \ldots\ldots\ldots + C_nX_n \qquad (2.1)$$

Batasan-batasan :

1). $a_{11}X_1 + a_{12}X_2 + a_{13}X_3 + \ldots\ldots\ldots + a_{1n}X_n \leq b_1$

2). $a_{21}X_1 + a_{22}X_2 + a_{23}X_3 + \ldots\ldots\ldots + a_{2n}X_n \leq b_2$

:

:

m). $a_{m1}X_1 + a_{m2}X_2 + a_{m3}X_3 + \ldots\ldots\ldots + a_{mn}X_n \leq b_m$

dan

$X_1 \geq 0, X_2 \geq 0, \ldots\ldots\ldots X_n \geq$

Fungsi tujuan dalam pemograman linear mencerminkan atau menggambarkan tujuan yang ingin dicapai dalam pembicaraan suatu masalah pemrograman linear. Terminologi umum untuk model pemrograman linear yang diuraikan diatas dapat diringkas sebagai berikut :

1. Fungsi yang akan dimaksimumkan : $C_1X_1 + C_2X_2 + C_3X_3 + \dots\dots\dots + C_nX_n$ disebut fungsi tujuan *(objective function)*.
2. Fungsi-fungsi batasan dikelompokan menjadi dua macam, yaitu :
   a. *Fungsi batasan fungsional*, yaitu fungsi-fungsi batasan sebanyak m (yaitu $a_{i1}X_1 + a_{i2}X_2 + a_{i3}X_3 + \dots\dots\dots + a_{im}X_n$).
   b. *Fungsi batasan non-negatif (non-negatif-constraints)* yaitu fungsi-fungsi batasan yang dinyatakan dengan $X_i \geq 0$.
3. Variabel-variabel X_j disebut sebagai *decision variables*.
4. a_{ij} , b_i , dan C_j , yaitu masukan-masukan (input) konstan ; disebut sebagai parameter model.

Pada prakteknya tidak semua masalah pemrograman linear dapat persis mengikuti model. Masalah-masalah tersebut antara lain :

1. Masalah minimisasi, dimana seseorang dituntut untuk menentukan kombinasi *(output)* yang dapat minimumkan pengorbanan (misal: biaya).
2. Masalah dengan fungsi batasan funsional yang memiliki tanda matematis $\geq$ (lebih besar sama dengan).
3. Masalah dengan fungsi batasan fungsional yang memiliki tanda matematis = (sama dengan).
4. Masalah tertentu, dimana fungsi batasan non-negatif tidak diperlukan atau dengan kata lain x_j tidak terbatas.

2.2.3 Asumsi-asumsi Dasar Pemograman Linear

Model pemrograman linear mengandung asumsi-asumsi implisit tertentu yang harus dipenuhi agar definisinya sebagai suatu masalah pemrograman linear menjadi absah. Asumsi itu menuntut bahwa hubungan fungsional dalam masalah itu adalah linear dan aditif, dapat dibagi dan deterministik. Asumsi-asumsi dasar pemrograman linear dapat diperinci sebagai berikut :

1. *Proportionality*

   Asumsi ini berarti bahwa naik turunnya nilai Z dan penggunaan sumber atau fasilitas yang tersedia akan berubah secara sebanding *(propotional)* dengan perubahan tingkat kegiatan.

2. *Additivity*

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

3. *Divisibility*

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

4. *Deterministic (Certainty)*

   Asumsi ini menyatakan bahwa semua parameter yang terdapat dalam model pemrograman linear dapat diperkirakan dengan pasti, meskipun jarang dengan tepat.

2.3 METODE SIMPLEKS

2.3.1 Konsep dan Karakteristik Dasar Metode Simpleks

Apabila suatu masalah program linear hanya mengandung dua kegiatan (atau variable-variabel keputusan) saja, maka akan dapat diselesaikan dengan metode grafik. Tetapi bila melibatkan lebih dari dua kegiatan maka metode grafik tidak dapat digunakan lagi, sehingga diperlukan metode simpleks. Pada umumnya program linear mempunyai lebih dari tiga variabel. Metode simpleks merupakan suatu cara yang lazim dipakai untuk menentukan kombinasi yang optimal dari tiga variabel atau lebih.

Metode simpleks ialah suatu metode yang secara sistematis dimulai dari suatu pemecahan dasar yang fisibel ke pemecahan dasar yang fisibel *(feasible)* lainnya dan dilakukan berulang-ulang (dengan jumlah ulangan yang terbatas) sehingga akhirnya tercapai suatu pemecahan dasar yang optimum dan pada setiap step menghasilkan suatu nialai dari fungsi tujuan yang lebih besar (lebih kecil) atau sama dari step-step sebelumnya [SUP,88]. Metode simpleks lebih efisien serta dilengkapi dengan suatu *test criteria* yang bisa memberitahukan kapan hitungan harus dihentikan dan kapan harus dilanjutkan sampai diperoleh suatu solusi yang optimal. Pada umumnya dipergunakan tabel-tabel, dari tabel pertama yang memberikan pemecahan dasar permulaan yang fisibel *(initial basic feasible solution)* sampai pada pemecahan terakhir yang memberikan solusi yang optimal.

2.3.2 Langkah-langkah Metode Simpleks

Untuk membantu membicarakan langkah-langakah yang dipakai dalam metode simpleks, akan diberikan contoh masalah sederhana berikut ini :

Perusahaan sepatu "IDEAL" membuat dua macam sepatu. Macam pertama merek I_1 dengan sol dari karet. Dan macam kedua merek I_2 dengan sol dari kulit. Untuk membuat sepatu-sepatu itu perusahaan mempunyai 3 macam mesin. Mesin 1 khusus membuat sol dari karet, mesin 2 khusus membuat sol dari kulit, dan mesin 3 membuat bagian atas sepatu dan melakukan assembling bagian atas dengan sol. Setiap lusin sepatu merek I_1 mula-mula dikerjakan di mesin 1 selama 2 jam, kemudian tanpa melalui mesin 2 terus dikerjakan dimesin 3 selama 6 jam. Sedang untuk sepatu merek I_2 tidak diproses dimesin 1, tetapi pertama kali dikerjakan dimesin 2 selama 3 jam kemudian di mesin 3 selama 5 jam. Jam kerja maksimum setiap hari untu mesin 1 = 8 jam, mesin 2 = 15 jam, mesin 3 = 30 jam. Sumbangan terhadap laba untuk setiap lusin merek I_1 = Rp. 30.000,00 sedang merek I_2 = Rp.50.000,00. Masalahnya adalah menetukan berapa lusin sebaiknya sepatu merek I_1 dan merek I_2 yang dibuat agar bisa memaksimumkan laba.

Langkah penyelesaian dari contoh diatas adalah sebagai berikut :

Langkah 1 : Mengubah fungsi tujuan dan batasan- batasan

Fungsi tujuan diubah menjadi fungsi implisit. Misalnya fungsi tujuan pada contoh diatas $Z = 3X_1 + 5X_2$ diubah menjadi $Z - 3X_1 - 5X_2 = 0$.

Pada bentuk standar, semua batasan mempunyai tanda $\leq$. Ketidaksamaan ini harus diubah menjadi kesamaan. Caranya dengan menambah *slack variable*. Variabel slack ini adalah X_{n+1} , X_{n+2} ....X_{n+m}. Karena tingkat atau hasil kegiatan-kegiatan yang ada diwakili oleh X_1 dan X_2, maka variabel slack dimulai dari X_3 , X_4 seperti tiga contoh dibawah ini dan seterusnya disebut sebagai batasan-batasan :

(1) $2X_1 \leq 8$ menjadi $2X_1 + X_3 = 8$

(2) $3X_2 \leq 15$ menjadi $3X_2 + X_4 = 15$

(3) $6X_1 + 5X_2 \leq 30$ menjadi $6X_1 + 5X_2 + X_5 = 30$

Langkah 2 : Menyusun persamaan persamaan di dalam tabel

Setelah formulasi diubah kemudian disusun kedalam tabel, dalam bentuk symbol seperti tampak pada Tabel 2.2.

Tabel 2.2. Tabel simpleks dalam bentuk simbol

Variabel dasar	Z	X_1	X_2	...	X_n	X_{n+1}	X_{n+2}	...	X_{n+m}	NK
Z	1	C_1	$-C_2$	...	$-C_n$	0	0	...	0	0
X_{n+1}	0	a_{11}	a_{12}	...	a_{1n}	1	0	...	0	b_1
X_{n+2}	0	a_{21}	a_{22}	...	a_{2n}	0	1	...	0	b_2
:	:	:	:		:	:	:		:	:
:	:	:	:		:	:	:		:	:
X_{n+m}	0	a_{m1}	a_{m2}	...	a_{mn}	0	0	...	1	b_m

NK adalah nilai kanan persamaan, yaitu nilai dibelakang tanda sama dengan (=). Untuk batasan 1 sebesar 8, batasan 2 sebesar 15, dan batasan 3 sebesar 30.

Variabel dasar adalah variabel yang nilainya sama dengan sisi kanan dari persamaan. Pada persamaan $2X_1 + X_3 = 8$, kalau belum ada kegiatan apa-apa, berarti nilai $X_1 = 0$, dan semua kapasitas masih menganggur, maka pengangguran ada 8 satuan, atau nilai $X_3 = 8$. Pada tabel tersebut nilai variabel dasar (X_3, X_4, X_5) pada fungsi tujuan pada tabel permulaan ini harus 0, dan nilainya pada batasan-batasan bertanda positif.

Tabel 2.3. Tabel simpleks yang pertama

Variabel dasar	Z	X_1	X_2	X_3	X_4	X_5	NK
Z	1	-3	-5	0	0	0	0
X_3	0	2	0	1	0	0	8
X_4	0	0	3	0	1	0	15
X_5	0	6	5	0	0	1	30

Setelah data disusun dalam tabel diatas (Tabel 2.3) kemudian diadakan perubahan selanjutnya (Langkah 3) agar dapat mencapai titik optimal.

Langkah 3 : Memilih kolom kunci

Kolom kunci adalah kolom yang merupakan dasar untuk mengubah tabel diatas (Tabel 2.3). Pilihlah kolom yang mempunyai nilai pada garis fungsi tujuan yang bernilai negatif dengan angka terbesar. Dalam hal ini kolom X_2 dengan nilai pada baris persamaan tujuan -5. Apabila suatu tabel sudah tidak memiliki nilai negatif pada baris fungsi tujuan berarti tabel itu sudah optimal.

Tabel 2.4. Tabel pemilihan kolom kunci pada tabel pertama

Variabel dasar	Z	X_1	X_2	X_3	X_4	X_5	NK	Keterangan
Z	1	-3	-5	0	0	0	0	
X_3	0	2	0	1	0	0	8	
X_4	0	0	3	0	1	0	15	15/3 = 5 (minimum)
X_5	0	6	5	0	0	1	30	30/5 = 6

↑
kolom kunci (X_2)

Langkah 4 : Memilih baris kunci

Baris kunci adalah baris yang merupakan dasar untuk mengubah tabel tersebut diatas (Tabel 2.4). Untuk itu terlebih dahulu carilah indeks tiap-tiap baris dengan cara membagi nilai-nilai pada kolom NK dengan nilai yang sebaris pada kolom kunci (kolom X_2). Untuk lebih jelasnya lihat kolom "keterangan" pada

Tabel 2.4. Untuk baris batasan 1 besar indeks = 8/0 = ~, baris batasan 2 = 15/3 = 5, dan baris batasan 3 = 30/5 = 6. Pilih baris yang mempunyai indeks positif dengan angka terkecil. Dalam hal ini batasan ke-2 (baris X_4) yang terpilih sebagai baris kunci. Nilai yang masuk dalam kolom kunci dan juga termasuk dalam baris kunci (masuk dalam kolom X_2 dan baris X_4 = 3) disebut angka kunci.

Tabel 2.5. Cara mengubah nilai baris kunci

Variabel dasar	Z	X_1	X_2	X_3	X_4	X_5	NK
Z	1	-3	-5	0	0	0	0
X_3	0	2	0	1	0	0	8
X_4	0	0	$\boxed{3}$	0	1	0	15
X_5	0	6	5	0	0	1	30
Z	1						
X_3	0						
X_2	0	0	1	0	1/3	0	5
X_5	0						

Langkah 5 : Mengubah nilai–nilai baris kunci

Nilai baris kunci diubah dengan cara membaginya dengan angka kunci, seperti terlihat pada Tabel 2.5 bagian bawah (0/ 3 = 0; 3/3 = 1; 0/3 = 0; 1/3 = 1/3; 0; 15/3 = 5). Gantilah variabel dasar pada baris itu dengan variabel yang terdapat dibagian atas kolom kunci (X_2).

Langkah 6 : Mengubah nilai-nilai selain pada baris kunci

Nilai-nilai baris yang lain, selain pada baris kunci dapat diubah dengan rumus sebagai berikut : Baris baru sama dengan baris lama dikurangi koefisien pada kolom kunci dikali nilai baru baris kunci.

Untuk data diatas, nilai baru baris pertama (Z) sebagai berikut :

	[-3	-5	0	0	0,	0]	
(-5)	[0	1	0	1/3	0,	5]	(-)
Nilai baru =	[-3	0	0	5/3	0,	25]	

Baris ke-2 (batasan 1) :

	[2	0	1	0	0,	8]	
0	[0	1	0	1/3	0,	5]	(-)
Nilai baru =	[2	0	1	0	0,	8]	

Baris ke-4 (batasan 3) :

	[6	5	0	0	1,	30]	
5	[0	1	0	1/3	0,	5]	(-)
nilai baru =	[6	0	0	-5/3	1,	5]	

Nilai-nilai baru diatas dipakai untuk melengkapi isi Tabel 2.5 bagian bawah, hasilnya terlihat pada Tabel 2.6.

Tabel 2.6. Tabel pertama nilai lama dan tabel kedua nilai baru

Variabel dasar	Z	X_1	X_2	X_3	X_4	X_5	NK
Z	1	-3	-5	0	0	0	0
X_3	0	2	0	1	0	0	8
X_4	0	0	3	0	1	0	15
X_5	0	6	5	0	0	1	30
Z	1	-3	0	0	5/3	0	25
X_3	0	2	0	1	0	0	8
X_2	0	0	1	0	1/3	0	5
X_5	0	6	0	0	-5/3	1	5

Langkah 7 : Perbaikan dan perubahan

Ulangilah langkah-langkah perbaikan mulai langkah 3 sampai langkah ke-6 untuk memperbaiki tabel-tabel yang telah diubah/diperbaiki nilainya. Perubahan baru berhenti setelah *pada baris pertama (fungsi tujuan) tidak ada yang negatif.*

Kalau tabel kedua (hasil perubahan) pada bagian bawah dari Tabel 2.9 itu kita ubah lagi, maka kolom dan baris kuncinya seperti terlihat pada Tabel 2.7.

Tabel 2.7. Kolom dan baris dari hasil perbaikan pertama, dan nilai baru baris kunci hasil perbaikan kedua

Variabel dasar	Z	X_1	X_2	X_3	X_4	X_5	NK
Z	1	-3	0	0	5/3	0	25
X_3	0	2	0	1	0	0	8 → 8/2=4
X_2	0	0	1	0	1/3	0	5
X_5	0	6	0	0	-5/3	1	5 → 5/6=5/6 ↑ minimum
Z	1						
X_3	0						
X_2	0						
X_1	0	1	0	0	-5/18	1/6	5/6

Nilai baru baris-baris yang lain kecuali baris kunci sebagai berikut :

Baris ke-1 :

[-3 0 0 5/3 0, 25]
(-3) [1 0 0 -5/18 1/6, 5/6] (-)

nilai baru = [0 0 0 5/6 1/2, 27½]

Baris ke-2 :

[2 0 1 0 0, 8]
2 [1 0 0 -5/18 1/6, 5/6] (-)

nilai baru = [0 0 1 5/9 -1/3, 6⅓]

Baris ke-3 : tidak berubah, karena nilai pada kolom kunci = 0.

Kalau hasil perubahan diatas kita masukan kedalam Tabel 2.7 bagian bawah, hasilnya seperti terlihat pada Tabel 2.8.

Tabel 2.8. Hasil perubahan/perbaikan kedua

Variabel dasar	Z	X_1	X_2	X_3	X_4	X_5	NK
Z	1	0	0	0	5/6	1/2	$27\frac{1}{2}$
X_3	0	0	0	1	5/9	-1/3	$6^1/_3$
X_2	0	0	1	0	1/3	0	5
X_1	1	1	0	0	-5/18	1/6	5/6

Kalau dilihat baris pertama (Z) pada Tabel 2.8 tidak ada lagi yang bernilai negatif, semuanya positif. Berarti tabel itu tidak dapat dioptimalkan lagi, sehingga hasil dari tabel tersebut sudah merupakan optimal.

Langkah-langkah secara keseluruhan :

Kalau tabel awal (sebelum diubah), tabel hasil perubahan pertama dan tabel hasil perubahan kedua dijadikan satu maka akan tampak jelas perubahannya seperti terlihat pada Tabel 2.9. Dari tabel ini akan tampak maksud dari tiap variabel dan nilai-nilai yang ada pada tabel optimal, yakni :

X_1 = 5/6, sehingga I_1 = 5/6 lusin setiap hari.

X_2 = 5, sehingga I_2 = 5 lusin setiap hari.

Z maksimum = $27\frac{1}{2}$, artinya laba yang akan diperoleh = Rp.275.000,00 setiap hari.

Tabel 2.9. Tabel-tabel yang diperoleh, dari tabel pertama sampai perubahan terakhir

Variabel dasar	Z	X_1	X_2	X_3	X_4	X_5	NK
Z	1	-3	-5	0	0	0	0
X_3	0	2	0	1	0	0	8
X_4	0	0	3	0	1	0	15
X_5	0	6	5	0	0	1	30
Z	1	-3	0	0	5/3	0	25
X_3	0	2	0	1	0	0	8
X_2	0	0	1	0	1/3	0	5
X_5	0	6	0	0	-5/3	1	5
Z	1	0	0	0	5/6	1/2	27½
X_3	0	0	0	1	5/9	-1/3	$6^1/_3$
X_2	0	0	1	0	1/3	0	5
X_1	0	1	0	0	-5/18	1/6	5/6

2.3.3 Ketentuan-ketentuan Tambahan Metode Simpleks

Masalah yang dihadapi kadang-kadang dapat menghasilkan dua kolom kunci, dua baris kunci, dan *multiple solution*. Maka terdapat ketentuan tambahan seperti dibawah ini:

1. *Terdapat lebih dari satu kolom bernilai negatif dengan angka terbesar*

Kalau pada baris fungsi tujuan terdapat lebih dari satu kolom yang mempunyai nilai negatif yang angkanya terbesar, maka ada dua kolom yang bisa terpilih menjadi kolom kunci. Untuk mengatasi hal ini bisa kita pilih salah satu di antara dua itu secara sembarang, akan menghasilkan keputusan yang sama.

2. *Dua baris atau lebih mempunyai indeks positif terkecil*

Kalau ada dua baris atau lebih yang mempunyai nilai positif terkecil, maka ada beberapa baris yang dapat terpilih sebagai baris kunci. Untuk mengatasi masalah ini dapat dipilih baris kunci secara bebas diantara keduanya, hasilnya juga nanti akan sama.

3. *Kenaikan nilai Z (tujuan) tidak terbatas*

Nilai Z (tujuan) suatu permasalahan dapat bertambah terus bila paling tidak ada satu kegiatan yang tidak ada batasannya. Sehingga dalam pemrograman linear hal-hal semacam ini tidak perlu dilanjutkan, cukup disebutkan bahwa kenaikan nilai Z (tujuan) dapat tidak terbatas. Disamping itu ada baiknya pula apabila diteliti lagi formulasi masalahnya, sebab hal ini dapat pula terjadi karena kesalahan dalam formulasi.

4. *Multiple Optimal Solution*

Dalam motode simpleks tabel, untuk mengetahui apakah suatu masalah pemrograman linear bersifat *multiple solution* atau tidak, dilihat dari baris fungsi tujuan pada tabel terakhir (optimal). Apabila di dalam baris fungsi tujuan terdapat paling tidak satu kolom variabel dasar yang mempunyai nilai 0 (nol) maka masalah itu bersifat *multiple solutions*. Dengan kata lain masalah itu akan menghasilkan paling tidak dua alternatif yang mempunyai nilai Z (tujuan) sama. Untuk mengatasi masalah ini bisa kita pilih salah satu diantara keduanya hasilnya akan sama.