

BAB II

LANDASAN TEORI

2.1 Pengertian Riset Operasi (*Operation Research*)

Istilah riset operasi pertama kali digunakan pada tahun 1940 oleh *Mc Closky* dan *Trefthen* di suatu kota kecil di Inggris bernama Bowdsey. Secara harfiah definisi riset operasi terbagi kedalam dua suku kata riset dan operasi, riset merupakan suatu proses yang terorganisasi dalam mencari kebenaran akan masalah sedangkan operasi dapat didefinisikan sebagai tindakan – tindakan yang diterapkan pada beberapa masalah atau hipotesa. Definisi lain dari beberapa ahli tentang riset operasi, diantaranya yaitu :

1. Menurut *Operation Research Society Of America* (1976), “Riset operasi berkaitan dengan menentukan pilihan secara ilmiah bagaimana merancang dan menjalankan system manusia – mesin secara terbaik, biasanya membutuhkan alokasi sumber daya yang langka”.
2. Menurut *Hamdi A. Taha* (1996), “Riset operasi merupakan pendekatan dalam pengambilan keputusan yang ditandai dengan penggunaan pengetahuan ilmiah melalui usaha kelompok antar disiplin yang bertujuan menentukan penggunaan terbaik sumber daya yang terbatas”.
3. Menurut *Churchman, Ackoff, dan Arnoff* (1959), mengemukakan pengertian riset operasi sebagai aplikasi metode – metode, teknik – teknik dan peralatan – peralatan ilmiah dalam menghadapi masalah – masalah yang timbul di dalam operasi perusahaan dengan tujuan ditemukannya pemecahan yang optimum masalah – masalah tersebut.
4. Menurut *Morse dan Kimball* (1951), “Mendefinisikan riset operasi sebagai metode ilmiah (*scientific method*) yang memungkinkan para manajer mengambil keputusan mengenai kegiatan yang mereka tangani dengan dasar kuantitatif”.
5. Menurut *Miller dan M.K.Starr* (1960), “Mengartikan riset operasi sebagai peralatan manajemen yang menyatukan ilmu pengetahuan, matematika,

dan logika dalam kerangka pemecahan masalah – masalah yang dihadapi sehari – hari, sehingga akhirnya permasalahan tersebut dapat dipecahkan secara optimal.

Setelah melihat definisi – definisi tentang riset operasi dari para ahli, bahwa dapat di definisikan riset operasi merupakan cara untuk memformulasikan permasalahan – permasalahan dalam kehidupan sehari – hari ke dalam model matematis untuk mendapatkan solusi yang optimal. Menurut Mulyono (2004), riset operasi memiliki beberapa ciri – ciri diantaranya sebagai berikut :

1. Riset operasi merupakan pendekatan kelompok antar disiplin untuk mencapai hasil optimum.
2. Riset operasi menggunakan teknik penelitian ilmiah untuk mendapatkan solusi optimum.
3. Riset operasi hanya memberikan jawaban yang jelek terhadap persoalan yang tersedia jawaban yang lebih jelek.
4. Riset operasi tidak memberikan jawaban yang sempurna terhadap masalah itu, sehingga riset operasi hanya memperbaiki kualitas solusi.

Dalam riset operasi terdapat beberapa tahapan, diantaranya :

1. Definisi masalah (Identifikasi model) terdiri dari :
 - Penentuan dan perumusan tujuan yang jelas dari persoalan dalam sistem model yang dihadapi.
 - Identifikasi perubah yang dipakai sebagai kriteria untuk pengambilan keputusan yang dapat dikendalian maupun yang tidak dapat dikendalikan.
2. Pembentukan model (Penyusunan model)
3. Mencari penyelesaian masalah (Analisa model)
4. Validasi model (Pengesahan model)
5. Penerapan hasil akhir (Implementasi Hasil)

2.2 Pemrograman Matematis

Pemrograman matematis (*mathematical programming*) adalah pembuatan model matematika atas suatu permasalahan yang sedang dihadapi dan menggunakan sebuah proses atau prosedur yang dapat diprogram, disebut algoritma, untuk mendapatkan solusinya. Model – model pemrograman matematika yang banyak digunakan adalah pemrograman linier (*linier programming*), pemrograman bilangan bulat (*integer programming*), pemrograman nonlinier (*non linier programming*), analisis jaringan (*network analysis*) dan pemrograman dinamis (*dynamic programming*). Model – pemrograman matematika tersebut dipakai untuk permasalahan *deterministic* (pasti).

Sebuah model atau pemrograman matematis memuat beberapa variabel keputusan, fungsi tujuan dan beberapa fungsi kendala dalam bentuk persamaan atau pertidaksamaan, di mana :

1. Variabel – variabel keputusan adalah variabel – variabel yang menggambarkan keputusan – keputusan yang akan dibuat.
2. Fungsi tujuan adalah fungsi dari harapan atau kriteria yang ingin dicapai, yang selanjutnya akan dimaksimalkan atau diminimalkan.
3. Batasan – batasan atau kendala adalah kondisi atau syarat yang membatasi nilai – nilai dari variabel keputusan yang mungkin.
4. Daerah solusi yang layak (*feasible space*) adalah daerah dari nilai – nilai variabel keputusan yang memenuhi semua kendala, atau semua kemungkinan kombinasi variabel keputusan yang memenuhi semua kendala.
5. Solusi tidak layak (*infeasible solution*) adalah solusi yang tidak memenuhi satu kendala atau lebih.
6. Solusi tak terbatas/terhingga (*unbounded solution*) adalah nilai fungsi tujuan tak terbatas
7. Banyak solusi optimal (*multiple optimal solution*) adalah nilai fungsi tujuan paling baik dengan jumlah dua atau lebih.

Adapun bentuk model matematis adalah memaksimumkan atau meminimumkan fungsi tujuan yang memenuhi kendala – kendala, syarat atau batasan. Model matematis dibedakan menjadi 2 kelompok yaitu

1. **Deterministik**, dibentuk dalam situasi kepastian (*certainty*), model ini memerlukan penyederhanaan – penyederhanaan dari realitas karena kepastian jarang terjadi. Keuntungan model ini adalah dapat dimanipulasi dan di selesaikan lebih mudah.
2. **Probabilistik**, meliputi kasus – kasus dimana diasumsikan ketidakpastian (*uncertainty*). Meskipun penggabungan ketidakpastian dalam model dapat menghasilkan suatu penyajian sistem nyata yang lebih realistis, model ini umumnya lebih sulit di analisa.

2.3 Model Deterministik

2.3.1 *Linear Programming*

Linear programming merupakan suatu model umum yang dapat digunakan dalam pemecahan masalah pengalokasian sumber – sumber yang terbatas secara optimal. Masalah tersebut timbul apabila seseorang diharuskan untuk memilih atau menentukan tingkat setiap kegiatan yang akan dilakukannya, di mana masing – masing kegiatan membutuhkan sumber yang sama sedangkan jumlahnya terbatas. Dalam model *linear programming* dikenal dua macam fungsi yaitu :

1. Fungsi tujuan (*objective function*), merupakan fungsi yang menggambarkan tujuan/sasaran di dalam permasalahan linear programming yang berkaitan dengan pengaturan secara optimal sumber daya, untuk memperoleh keuntungan maksimal atau biaya minimal. Pada umumnya nilai yang akan dioptimalkan dinyatakan sebagai Z.
2. Fungsi batasan (*constraint function*), merupakan bentuk penyajian secara matematis batasan – batasan kapasitas yang tersedia yang akan dialokasikan secara optimal ke berbagai kegiatan.

Dalam membahas model *linear programming*, digunakan beberapa simbol agar memudahkan dalam menyelesaikan suatu permasalahan yang ada, diantaranya adalah sebagai berikut :

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

n = macam kegiatan yang menggunakan sumber fasilitas tersebut.

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

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

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

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

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

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 tersebut selanjutnya disusun ke dalam bentuk tabel standar *linear programming* seperti dibawah ini.

Tabel 2.1 Tabel Standar *Linear Programming*

Sumber/Kegiatan	Pemakaian sumber per unit kegiatan (Output)				Kapasitas sumber
	1	2	...	n	
1	a_{11}	a_{12}	...	a_{1n}	b_1
2	a_{21}	a_{22}	...	a_{2n}	b_2
-	-	-	...	-	-
-	-	-	...	-	-
m	a_{m1}	a_{m2}	...	a_{mn}	b_m
ΔZ pertambahan tiap unit	C_1	C_2	...	C_n	
Tingkat kegiatan	X_1	X_2	...	X_n	

Sumber : Subagyo, P. dkk, 2005

Berdasarkan tabel 2.1 dapat disusun model matematis yang digunakan untuk mengemukakan suatu permasalahan *linear programming*, yaitu sebagai berikut :

Fungsi tujuan :

$$\text{Maksimumkan } Z = C_1X_1 + C_2X_2 + \dots + C_nX_n$$

Batasan – batasan :

- 1) $a_{11}X_1 + a_{12}X_2 + \dots + a_{1n}X_n \leq b_1$
- 2) $a_{21}X_1 + a_{22}X_2 + \dots + a_{2n}X_n \leq b_2$
- 3) $a_{m1}X_1 + a_{m2}X_2 + \dots + a_{mn}X_n \leq b_m$

dan $X_1 \geq 0, X_2 \geq 0, \dots, X_n \geq 0$

Bentuk atau model *linear programming* diatas merupakan bentuk standar bagi masalah – masalah *linear programming* yang akan dipakai selanjutnya. Dengan kata lain apabila setiap masalah dapat diformulasikan secara matematis mengikuti model di atas, maka masalah tersebut dapat dipecahkan dengan teknik *linear programming*. Bentuk umum dari model *linear programming* diatas dapat diringkas sebagai berikut :

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

Seharusnya semua asumsi – asumsi dasar linear programming telah tersirat pada model yang telah dibahas diatas, tetapi ada baiknya untuk menguraikan asumsi – asumsi dasar tersebut agar penggunaan teknik linear programming ini

dapat memuaskan tanpa terbentur pada berbagai hal. Asumsi – asumsi dasar *linear programming* 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 (proportional) dengan perubahan tingkat kegiatan. Misalnya :

a) $Z = C_1X_1 + C_2X_2 + \dots + C_nX_n$

Setiap penambahan 1 unit X_1 akan menaikkan Z dengan C_1 .

Setiap penambahan 1 unit X_2 akan menaikkan nilai Z dengan C_2 , dan seterusnya.

b) $a_{11}X_1 + a_{12}X_2 + \dots + a_{1n}X_n \leq b_1$

Setiap penambahan 1 unit X_1 akan menaikkan penggunaan sumber/fasilitas 1 dengan a_{11} .

Setiap penambahan 1 unit X_2 akan menaikkan penggunaan sumber/fasilitas 1 dengan a_{12} , dan seterusnya. Dengan kata lain setiap ada kenaikan kapasitas riil tidak perlu ada biaya persiapan (*set up cost*).

2. *Additivity*

Asumsi ini berarti bahwa nilai tujuan tiap kegiatan tidak saling mempengaruhi, atau dalam linear programming 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. Misalnya :

$$3X_1 + 5X_2$$

Dimana $X_1 = 10$; $X_2 = 2$ sehingga $Z = 30 + 10 = 40$

Andaikan X_1 bertambah 1 unit, maka sesuai dengan asumsi pertama, nilai Z menjadi $40 + 3 = 43$. Jadi, nilai 3 karena kenaikan X_1 dapat langsung ditambahkan pada nilai Z mula – mula mengurangi bagian Z yang diperoleh dari kegiatan 2 (X_2). Dengan kata lain, tidak ada korelasi antara X_1 dan X_2 .

3. Divisibility

Asumsi ini menyatakan bahwa keluaran (output) yang dihasilkan oleh setiap kegiatan dapat berupa bilangan pecahan. Demikian pula dengan nilai Z yang dihasilkan. Misalnya : $X_1 = 6,5$; $Z = 1.000,75$.

4. Deterministic (Certainty)

Asumsi ini menyatakan bahwa semua parameter yang terdapat dalam model linear programming (a_{ij} , b_i , C_j) dapat diperkirakan dengan pasti, meskipun jarang dengan tepat.

2.3.2 Model Transportasi

Model transportasi merupakan suatu metode yang digunakan untuk mengatur distribusi dari sumber – sumber yang menyediakan produk yang sama, ke tempat – tempat yang membutuhkan secara optimal. *F.L. Hitchcock* (1941), *T.C. Koopmans* (1949), dan *G.B. Dantzig* (1951) adalah orang – orang pertama sebagai kontributor yang mengembangkan teknik – teknik transportasi. Alokasi produk harus diatur sedemikian rupa, karena terdapat perbedaan biaya – biaya alokasi dari satu sumber ke tempat – tempat tujuan berbeda – beda, dan dari beberapa sumber ke suatu tempat tujuan juga berbeda – beda. Ada beberapa macam metode transportasi, yang semuanya terarah pada penyelesaian optimal dari masalah – masalah transportasi yang terjadi. Tabel awal dalam permasalahan transportasi dapat dibuat dengan menggunakan 2 metode, yaitu :

2.3.2.1 Metode North West Corner (NWC)

Merupakan metode transportasi yang penyelesaiannya dimulai dari pojok kiri atas tanpa memperhitungkan besarnya biaya.

2.3.2.2 Metode Least Cost (LC)

Merupakan metode transportasi yang penyelesaiannya dimulai dengan biaya terkecil terlebih dahulu. Berikut ini merupakan contoh dari metode *Least Cost* (LC) :

Tabel 2.2 Contoh Tabel Transportasi Metode *North West Corner*

Dari/Ke	Gudang A		Gudang B		Gudang C		Kapasitas Pabrik
Pabrik W	50	20	40	5		8	90
Pabrik H		15	60	20		10	60
Pabrik P		25	10	10	40	19	50
Kebutuhan Gudang	50		110		40		200

Sumber : Subagyo, P. dkk, 2005

Biaya yang dikeluarkan dari metode *North West Corner* yaitu :

$$(50 \times 20) + (40 \times 5) + (60 \times 20) + (10 \times 10) + (40 \times 19) = 3260$$

Tabel 2.3 Contoh Tabel Transportasi Metode *Least Cost*

Dari/Ke	Gudang A		Gudang B		Gudang C		Kapasitas Pabrik
Pabrik W		20	90	5		8	90
Pabrik H	20	15		20	40	10	60
Pabrik P	30	25	20	10		19	50
Kebutuhan Gudang	50		110		40		200

Sumber : Subagyo, P. dkk, 2005

Biaya yang dikeluarkan dari metode *Least Cost* yaitu :

$$(90 \times 5) + (40 \times 10) + (20 \times 10) + (20 \times 15) + (30 \times 25) = 2400$$

2.3.2.3 Metode Vogel's Approximation

Metode *Vogel* atau *Vogel's Approximation Method (VAM)* merupakan metode yang lebih mudah dan lebih cepat untuk dapat mengatur alokasi dari beberapa sumber ke beberapa daerah pemasaran. Adapun langkah – langkah untuk mengerjakannya adalah sebagai berikut :

1. Susunlah kebutuhan, kapasitas masing – masing sumber, dan biaya pengangkutan ke dalam matriks.
2. Carilah perbedaan dari dua biaya terkecil (dalam bilai absolut), yaitu biaya terkecil dan terkecil kedua untuk tiap baris dan kolom pada matriks (Cij). Misalnya pada baris W biaya angkut terkecil = Rp5,00 dan nomor dua dari yang terkecil = Rp8,00. Jadi nilai baris W = $8 - 5 = 3$. Demikian seterusnya nilai – nilai yang lain sebagai berikut :

$$\text{Baris H} = 15 - 10 = 5$$

$$\text{Baris P} = 19 - 10 = 9$$

$$\text{Kolom A} = 20 - 15 = 5$$

$$\text{Kolom B} = 10 - 5 = 5$$

$$\text{Kolom C} = 10 - 8 = 2$$

Tabel 2.4 Feasible solution awal dari metode *Vogel's Approximation*

Sumber		Gudang			Kapasitas	Perbedaan Baris
		A	B	C		
Pabrik	W	20	5	8	90	3
	H	15	20	10	60	5
	P	25	10	19	50	9
Kebutuhan		50	110	40	Pilihan $X_{PB} = 50$	
Perbedaan kolom		5	5	2	Hilangkan baris P	

Sumber : Subagyo, P. dkk, 2005

3. Pilihlah 1 nilai perbedaan – perbedaan yang terbesar di antara semua nilai perbedaan pada kolom dan baris. Dalam hal ini baris P mempunyai nilai perbedaan sebesar 9.
4. Isilah pada salah satu segi empat yang termasuk dalam kolom atau baris terpilih, yaitu pada segi empat yang biayanya terendah di antara segi empat lain pada kolom/baris itu. Isiaannya sebanyak mungkin yang bisa dilakukan.

Misalnya pada baris P, biaya angkut untuk segi empat PA = 25, segi empat PB = 10, dan segi empat PC = 19, yang terkecil adalah biaya pada segi empat PB. Maka kita isi segi empat PB dengan 50 satuan (lebih dari 50 satuan tidak mungkin karena kapasitas pabrik P = 50).

5. Hilangkan baris P karena baris tersebut sudah diisi sepenuhnya (kapasitas penuh) sehingga tidak mungkin diisi lagi. Kemudian perhatikan kolom dan baris yang belum terisi/teralokasi (baris W, H, dan kolom A, B, C).
6. Tentukan kembali perbedaan (selisih) biaya pada langkah ke – 2 untuk kolom dan baris yang belum terisi. Ulangi langkah 3 sampai dengan langkah 5, sampai semua baris dan kolom sepenuhnya teralokasi lihat tabel 2.5

Tabel 2.5 Feasible solution dari metode *Vogel's Approximation* lanjutan 1

Sumber		Gudang			Kapasitas	Perbedaan Baris
		A	B	C		
Pabrik	W	20	5	8	90	3
	H	15	20	10	60	5
Kebutuhan		50	110	40	Pilihan $X_{WB} = 50$	
Perbedaan kolom		5	15	2	Hilangkan baris B	

Sumber : Subagyo, P. dkk, 2005

Tabel 2.6 Feasible solution dari metode *Vogel's Approximation* lanjutan 2

Sumber		Gudang		Kapasitas	Perbedaan Baris
		A	C		
Pabrik	W	20	8	30	12
	H	15	10	60	5
Kebutuhan		50	40	Pilihan $X_{WC} = 30$	
Perbedaan kolom		5	2	Hilangkan baris W	

Sumber : Subagyo, P. dkk, 2005

Tabel 2.7 Feasible solution dari metode *Vogel's Approximation* lanjutan 3

Sumber		Gudang		Kapasitas	Perbedaan Baris
		A	C		
Pabrik	H	15	10	60	5
Kebutuhan		50	10	Pilihan $X_{HA} = 50$	
Perbedaan kolom		5	2	$X_{HC} = 10$	

Sumber : Subagyo, P. dkk, 2005

Tabel 2.8 Matriks hasil akhir alokasi dengan metode VAM

Dari/Ke	Gudang A	Gudang B	Gudang C	Kapasitas Pabrik
Pabrik W	20	60	30	90
Pabrik H	50	20	10	60
Pabrik P	25	50	19	50
Kebutuhan Gudang	50	110	40	200

Sumber : Subagyo, P. dkk, 2005

7. Setelah terisi semua, maka biaya transportasinya yang harus dibayar adalah 60 (Rp 5,00) + 30 (Rp 8,00) + 50 (Rp 15,00) + 10 (Rp 10,00) + 50 (Rp 10,00) = Rp 1.890,00.

8. Bila nilai perbedaan biaya ada 2 yang besarnya sama, misal yang satu terletak pada kolom maka :

Lihatlah segi empat yang masuk ke dalam kolom maupun baris yang mempunyai nilai terbesar. Bila segi empat ini mempunyai biaya terendah di antara segi empat pada baris atau kolomnya, maka isikan alokasi maksimum pada segi empat ini. Bila biayanya tidak terendah, maka pilihlah segi empat yang akan diisi berdasar salah satu, baris terpilih atau kolom terpilih, seperti pada langkah 4 dan 5.

Kebaikan dari metode *Vogel's* ini adalah mudah menghitungnya, tetapi hasil pemecahan dari metode ini kadang – kadang masih dapat di optimalkan dengan memakai metode lain, misalnya metode *Simpleks*.

2.3.3 Model Penugasan

Seperti masalah transportasi, masalah penugasan (*assignment problem*) merupakan suatu kasus khusus dari masalah *linear programming* pada umumnya. Dalam dunia usaha (bisnis) dan industri, manajemen sering menghadapi masalah – masalah yang berhubungan dengan penugasan optimal dari bermacam – macam sumber yang produktif atau personalia yang mempunyai tingkat efisiensi yang berbeda – beda untuk tugas yang berbeda pula. Metode Hungarian adalah salah satu dari beberapa teknik – teknik pemecahan yang tersedia untuk permasalahan penugasan. Metode ini awal mulanya dikembangkan oleh seorang ahli matematika berkebangsaan Hungaria bernama *D. Konig* pada tahun 1916.

Untuk dapat menerapkan metode Hungarian, jumlah sumber – sumber yang ditugaskan harus sama persis dengan jumlah tugas yang akan diselesaikan. Selain itu, setiap sumber harus ditugaskan hanya untuk satu tugas. Jadi, masalah penugasan akan mencakup sejumlah n sumber yang mempunyai n tugas. Ada $n!$ (n factorial) penugasan yang mungkin dalam suatu masalah karena perpasangan satu – satu. Masalah ini dapat dijelaskan dengan mudah oleh bentuk matriks segi empat, di mana baris – barisnya menunjukkan sumber – sumber dan kolom – kolomnya menunjukkan tugas – tugas.

Masalah penugasan dapat dinyatakan secara matematis dalam suatu bentuk *linear programming* sebagai berikut :

Minimumkan (maksimumkan) :

$$Z = \sum_{i=1}^m \sum_{j=1}^n C_{ij} X_{ij} \dots\dots\dots (1)$$

Dengan batasan :

$$\sum_{i=1}^m X_{ij} = \sum_{j=1}^n X_{ij} = 1 \dots\dots\dots (2)$$

Dan $X_{ij} \geq 0$ ($X_{ij} = X_{ij}^2$)

Di mana C_{ij} adalah tetapan yang telah diketahui.

2.3.3.1. Langkah – Langkah Penyelesaian Metode Penugasan (*Hungarian*)

Berikut ini merupakan tahapan dalam menyelesaikan permasalahan penugasan menurut Subagyo, dkk (1995) adalah :

1. Mengubah matriks biaya menjadi matriks *opportunity cost*. Ini dicapai dengan memilih elemen terkecil dari setiap baris dari matriks biaya mula – mula untuk mengurangi seluruh elemen (bilangan) dalam setiap baris.
2. *Reduced cost matrix* diatas terus dikurangi untuk mendapatkan *total – opportunity – cost matrix*. Hal ini dapat dicapai dengan memilih elemen terkecil dari setiap kolom pada *reduced – cost matrix* untuk mengurangi seluruh elemen dalam kolom – kolom tersebut.
3. Mencari *schedule* penugasan dengan suatu *total – opportunity – cost* nol. Prosedur praktis untuk melakukan test optimalisasi adalah dengan menarik sejumlah minimum garis horisontal dan atau vertikal untuk meliputi seluruh elemen bernilai nol dalam *total – opportunity – cost matrix*.
4. Untuk merevisi *total – opportunity – cost matrix*, pilih elemen terkecil yang belum terliput garis – garis (yaitu *opportunity – cost* terendah) untuk mengurangi seluruh elemen yang belum terliput.

Untuk mempermudah pemahaman tentang metode penugasan (*Hungarian*), maka akan dijelaskan berdasarkan contoh seperti berikut ini untuk kasus minimasi dan minimasi.

a. Masalah Minimasi

Suatu perusahaan kecil mempunyai 4 (empat) pekerjaan yang berbeda untuk diselesaikan oleh 4 (empat) karyawan. Biaya penugasan seorang karyawan untuk pekerjaan yang berbeda adalah berbeda karena sifat pekerjaan berbeda – beda. Setiap karyawan mempunyai tingkat keterampilan, pengalaman kerja, dan latar belakang pendidikan serta latihan yang berbeda pula. Sehingga biaya penyelesaian pekerjaan yang sama oleh karyawan – karyawan yang berlainan juga berbeda. Berikut ini merupakan matriks biaya yang dikeluarkan yaitu :

Tabel 2.9 Matriks biaya

Karyawan/Pekerjaan	I	II	III	IV
A	Rp 15.00	Rp 20.00	Rp 18.00	Rp 22.00
B	14	16	21	17
C	25	20	23	20
D	17	18	18	16

Sumber : Subagyo, P. dkk, 2005

Karena metode penugasan Hungarian mensyaratkan perpasangan satu – satu, maka ada $4!$ ($4 \cdot 3 \cdot 2 \cdot 1 = 24$) kemungkinan penugasan. Langkah – langkah penyelesaiannya adalah sebagai berikut :

1. Langkah pertama adalah mengubah matriks biaya menjadi matriks *opportunity cost*, ini dicapai dengan memilih elemen terkecil dari setiap baris dari matriks biaya mula – mula untuk mengurangi seluruh elemen (bilangan) dalam setiap baris. Sebagai contoh, elemen terkecil baris A (=15) digunakan untuk mengurangi seluruh elemen pada baris A. Sehingga paling sedikit akan diperoleh satu elemen yang bernilai nol sebagai hasilnya. Prosedur yang sama diulang untuk setiap baris. Berikut merupakan hasil pengurangan dari setiap baris.

Tabel 2.10 *Reduced-cost matrix*

Karyawan/Pekerjaan	I	II	III	IV
A	0	5	3	7
B	0	2	7	3
C	5	0	3	0
D	1	2	2	0

Sumber : Subagyo, P. dkk, 2005

2. *Reduced cost-matrix* di atas terus dikurangi untuk mendapatkan *total-opportunity-cost matrix*. Hal ini dapat dicapai dengan memilih elemen terkecil dari setiap kolom pada *reduced-cost matrix* untuk mengurangi seluruh elemen dalam kolom – kolom tersebut. Pada tabel diatas hanya dilakukan pada kolom III karena semua kolom lainnya telah mempunyai elemen yang bernilai nol. Maka akan didapatkan hasil seperti pada tabel dibawah ini.

Tabel 2.11 Total-opportunity-cost matrix

Karyawan/Pekerjaan	I	II	III	IV
A	0	5	1	7
B	0	2	5	3
C	5	0	1	0
D	1	2	0	0

Sumber : Subagyo, P. dkk, 2005

3. Langkah berikutnya adalah mencari skedul penugasan dengan suatu total-*opportunity-cost* nol. Untuk mencapai penugasan ini dibutuhkan 4 (empat) “*independent zeros*” dalam matriks. Ini berarti setiap karyawan harus ditugaskan hanya untuk satu pekerjaan dengan *opportunity-cost* nol atau setiap pekerjaan harus diselesaikan hanya oleh satu karyawan. Prosedur praktis untuk melakukan test optimalisasi adalah dengan menarik sejumlah minimum garis *horizontal/vertical* untuk meliputi seluruh elemen bernilai nol dalam *total-opportunity-cost* matrix. Bila jumlah garis sama dengan jumlah baris/kolom penugasan telah optimal (*feasible*). Bila tidak sama maka matriks harus direvisi.

Tabel 2.12 Test for optimality

Karyawan/Pekerjaan	I	II	III	IV
A	0	5	1	7
B	0	2	5	3
C	5	0	1	0
D	1	2	0	0

Sumber : Subagyo, P. dkk, 2005

4. Untuk merevisi *total-opportunity-cost* matrix, pilih elemen terkecil yang belum terliput garis – garis (*opportunity-cost* terendah) untuk mengruangi seluruh elemen yang belum terliput. Kemudian tambahkan dengan jumlah yang sama (nilai elemen terkecil) pada seluruh elemen – elemen yang mempunyai dua garis yang saling bersilangan.

Tabel 2.13 Revised matrix dan test for optimality

Karyawan/Pekerjaan	I	II	III	IV
A	0	4	0	6
B	0	1	4	2
C	5	0	1	0
D	2	2	0	0

Sumber : Subagyo, P. dkk, 2005

5. Dalam tabel diatas, dibutuhkan empat garis untuk meliputi seluruh nilai nol atau sama dengan jumlah baris atau kolom, sehingga matriks penugasan optimal telah tercapai.

Tabel 2.14 Schedule penugasan akhir

Schedule Penugasan	Biaya
A - III	Rp 18.00
B - I	14
C - II	20
D - IV	10
Total	Rp 68.00

Sumber : Subagyo, P. dkk, 2005

b. Jumlah Pekerjaan Tidak Sama dengan Jumlah Karyawan

Untuk memenuhi persyaratan suatu matriks segi empat bujur sangkar, agar metode Hungarian dapat diterapkan, bila terdapat jumlah pekerjaan lebih besar dari jumlah karyawan, maka harus ditambahkan suatu karyawan semu (*dummy worker*). Biaya semua adalah sama dengan nol, karena tidak akan terjadi biaya bila suatu pekerjaan ditugaskan ke karyawan semu atau dengan kata lain karena sebenarnya pekerjaan tersebut tidak dilaksanakan. Sebaliknya bila jumlah karyawan lebih besar dari jumlah pekerjaan, maka harus ditambahkan suatu pekerjaan semu (*dummy job*). Sedangkan prosedur penyelesaiannya sama dengan langkah – langkah di atas. Sebagai contoh dapat dilihat pada tabel dibawah ini.

Tabel 2.15 Jumlah pekerjaan lebih besar dari jumlah karyawan

Karyawan/Pekerjaan	I	II	III	IV
A	Rp 15.00	Rp 20.00	Rp 18.00	Rp 22.00
B	14	16	21	17
C	25	20	23	20
D	17	18	18	16
<i>Dummy E</i>	0	0	0	0

Sumber : Subagyo, P. dkk, 2005

c. Masalah Maksimasi

Dalam masalah maksimasi, matriks elemen – elemen menunjukkan tingkat keuntungan atau indeks produktivitas. Efektivitas pelaksanaan tugas oleh karyawan – karyawan individual diukur dengan jumlah kontribusi keuntungan. Berikut merupakan contoh dari masalah penugasan maksimasi.

Tabel 2.16 Matriks keuntungan

Karyawan/Pekerjaan	I	II	III	IV	V
A	Rp 10.00	Rp12.00	Rp10.00	Rp 8.00	Rp15.00
B	14	10	9	15	13
C	9	8	7	8	12
D	13	15	8	16	11
E	10	13	14	11	17

Sumber : Subagyo, P. dkk, 2005

Langkah pertama dalam masalah maksimasi adalah mengubah matriks keuntungan menjadi suatu matriks *opportunity-loss*. Dalam masalah ini, A menyumbang keuntungan tertinggi Rp 15 bila dia ditugaskan pada pekerjaan V. Oleh karena itu, bila A ditugaskan pada pekerjaan I (yang kontribusi keuntungannya = Rp 10), ada sebesar Rp 5 sebagai *opportunity-loss* yang terjadi dengan penugasan ini, dan seterusnya. Seluruh elemen dalam setiap baris dikurangi dengan nilai maksimum dalam baris yang sama. Berikut ini merupakan hasilnya.

Tabel 2.17 Matriks *opportunity-loss*

Karyawan/Pekerjaan	I	II	III	IV	V
A	5	3	5	7	0
B	1	5	6	0	2
C	3	4	5	4	0
D	3	1	8	0	5
E	7	4	3	0	0

Sumber : Subagyo, P. dkk, 2005

Seperti sebelumnya, setiap baris akan berisi nilai nol. Langkah berikutnya dengan meminimumkan *opportunity-loss* akan memaksimumkan kontribusi keuntungan total. Berikut merupakan hasil pengurangan untuk setiap kolom yang belum mempunyai nilai 0.

Tabel 2.18 Matriks *total-opportunity-loss*

Karyawan/Pekerjaan	I	II	III	IV	V
A	4	2	2	7	0
B	0	4	3	0	2
C	2	3	2	4	0
D	2	0	5	0	5
E	6	3	0	6	0

Sumber : Subagyo, P. dkk, 2005

Pada tabel diatas seluruh elemen bernilai nol dapat diliput hanya dengan empat garis. Jadi, matriks harus dikurangi menurut langkah ke-4 seperti yang telah dijelaskan di muka. Matriks baru ditunjukkan pada tabel dibawah ini, di mana penugasan optimal dapat ditentukan.

Tabel 2.19 Tabel optimal

Karyawan/Pekerjaan	I	II	III	IV	V
A	2	0	0	5	0
B	0	4	0	0	4
C	0	1	0	2	0
D	2	0	0	0	7
E	6	3	0	6	2

Sumber : Subagyo, P. dkk, 2005

Schedule penugasan optimal dan keuntungan total untuk dua *alternative* penyelesaian adalah :

Tabel 2.20 *Schedule* penugasan optimal

<i>Schedule</i> Penugasan 1	Keuntungan	<i>Schedule</i> Penugasan 2	Keuntungan
A - II	Rp 12.00	A - V	Rp 15.00
B - I	14	B - IV	15
C - V	12	C - I	9
D - IV	16	D - II	15
E - III	14	E - III	14
Total	Rp 68.00	Total	Rp 68.00

Sumber : Subagyo, P. dkk, 2005

2.3.3.2. Masalah – Masalah Penugasan Tambahan

Dalam masalah – masalah penugasan, terkadang beberapa elemen matriks tidak diketahui sama sekali, ada beberapa alasan mengapa terdapat elemen – elemen yang tidak diketahui. Dalam masalah penugasan personalia, sebagai contoh seorang karyawan tertentu tidak dapat ditugaskan untuk melaksanakan pekerjaan tertentu karena tidak memenuhi persyaratan keterampilan yang diperlukan, defisiensi dalam pengetahuan teknis, latihan yang tidak tepat, ketidakmampuan fisik, dan sebagainya. Penugasan untuk keadaan tersebut bisa tidak mungkin dilakukan, atau tidak menguntungkan bila dilakukan.

Untuk pemecahan suatu masalah penugasan yang tidak mungkin dilakukan, kita hanya menandai setiap elemen penugasan yang tidak mungkin dengan suatu nilai sangat besar yang tidak diketahui, M (yaitu M untuk masalah minimasi dan $-M$ untuk masalah maksimasi).