

OPTIMASI MASALAH PENUGASAN

Siti Maslihah

Abstrak

Pemrograman linier merupakan salah satu ilmu matematika terapan yang bertujuan untuk mencari nilai optimum dari suatu permasalahan, yang dirumuskan dalam model matematika. Salah satu aplikasinya adalah mencari nilai optimum masalah penugasan (memaksimalkan penempatan tenaga kerja yang sesuai dengan kemampuannya atau meminimalkan biaya penempatan tenaga kerja). Penyelesaian masalah penugasan bisa dilakukan dengan Metode Hungarian. Selain Metode Hungarian, sudah banyak software yang dapat digunakan untuk mengeksekusi masalah penugasan, salah satu software tersebut adalah LINGO.

Kata kunci: metode penugasan, Hungarian, LINGO.

Pendahuluan

Pemrograman Linier disingkat PL merupakan metode matematik dalam mengalokasikan sumber daya yang terbatas untuk mencapai suatu tujuan seperti memaksimalkan keuntungan dan meminimumkan biaya. PL banyak diterapkan dalam masalah ekonomi, industri, militer, sosial dan lain-lain. PL berkaitan dengan penjelasan suatu kasus dalam dunia nyata sebagai suatu model matematik yang terdiri dari sebuah fungsi tujuan linier dengan beberapa kendala linier.

Dewasa ini banyak usaha yang memproduksi barang dan jasa, di mana produk yang dihasilkan harus mampu menyaingi produk yang lain. Sehingga perusahaan harus pandai mengelola aspek produksi dan penempatan tenaga kerjanya. Diharapkan dengan penempatan tenaga kerja yang tepat akan memaksimalkan hasil kinerja yang didapatkan. Untuk itu dibutuhkan suatu metode atau cara bagaimana menempatkan tenaga kerja yang tepat agar hasil kerjanya maksimum/ optimum. PL mampu menjawab masalah tersebut untuk mengoptimalkan sumber daya/ tenaga kerja yang ada. Salah satu metode yang digunakan untuk masalah penugasan optimal

adalah Metode Hungarian. Metode ini dikembangkan oleh seorang ahli matematika yang berkebangsaan Hungaria yang bernama D Konig pada tahun 1916.

Kajian Pustaka

Masalah penugasan merupakan kasus khusus dari masalah transportasi, di mana yang menjadi sumber adalah tenaga kerja dan yang menjadi tujuan adalah jenis pekerjaan yang tersedia.

fungsi tujuan :

$$\begin{aligned} \text{Max/ Min } Z &= C_{11} X_{11} + C_{12} X_{12} + \dots + C_{1n} X_{1n} \longrightarrow \sum_j C_{1j} X_{1j} + \\ &C_{21} X_{21} + C_{22} X_{22} + \dots + C_{2n} X_{2n} \longrightarrow \sum_j C_{2j} X_{2j} + \\ &\cdot \\ &\cdot \\ &C_{m1} X_{m1} + C_{m2} X_{m2} + \dots + C_{mn} X_{mn} \longrightarrow \sum_j C_{mj} X_{mj} \end{aligned}$$

Kendala sumber :

$$\begin{aligned} X_{11} + X_{12} + \dots + X_{1n} &\leq a_1 \longrightarrow \sum_j X_{1j} \leq a_1 \\ X_{21} + X_{22} + \dots + X_{2n} &\leq a_2 \longrightarrow \sum_j X_{2j} \leq a_2 \\ &\cdot \\ &\cdot \\ X_{m1} + X_{m2} + \dots + X_{mn} &\leq a_m \longrightarrow \sum_j X_{mj} \leq a_m \end{aligned}$$

Kendala tujuan :

$$\begin{aligned} X_{11} + X_{21} + \dots + X_{m1} &\geq b_1 \longrightarrow \sum_i X_{i1} \geq b_1 \\ X_{12} + X_{22} + \dots + X_{m2} &\geq b_2 \longrightarrow \sum_i X_{i2} \geq b_2 \\ &\cdot \\ &\cdot \\ X_{1n} + X_{2n} + \dots + X_{mn} &\geq b_n \longrightarrow \sum_i X_{in} \geq b_n \end{aligned}$$

Model di atas dapat diringkas menjadi

$$\text{f.t min } Z = \sum_i \sum_j C_{ij} X_{ij} \quad \begin{array}{l} i = 1, 2, \dots, m \\ j = 1, 2, \dots, n \end{array}$$

$$\text{dk:} \quad \sum_j X_{ij} \leq a_i \quad i = 1, 2, \dots, m$$

$$\sum_i X_{ij} \geq b_j \quad j = 1, 2, \dots, n$$

$$X_{ij} \geq 0$$

Syarat penyelesaian tersebut dengan model penugasan adalah model dalam keadaan seimbang, yaitu : Banyaknya pekerja = banyaknya pekerjaan $\rightarrow \sum_i a_i = \sum_j b_j$

$$\rightarrow \sum_i a_i = \sum_j b_j$$

Masalah penugasan bermula dari penempatan para pekerja pada bidang yang tersedia agar biaya yang ditanggung perusahaan dapat diminimalkan. Jika pekerja dianggap sebagai sumber dan pekerjaan dianggap sebagai tujuan, maka model penugasan akan sama dengan masalah transportasi, dimana jumlah sumber dan tujuan sama, setiap sumber hanya menghasilkan satu demikian pula setiap tujuan hanya memerlukan satu.

Suplai pada semua sumber adalah 1, yaitu $a_i = 1$ untuk semua i . Hal yang sama juga terjadi pada tujuan, permintaan pada semua tujuan adalah 1, yaitu $b_j = 1$ untuk semua j . Karena x_{ij} menunjukkan penugasan, berarti nilainya hanya 2, yaitu 0 atau 1.

$$x_{ij} = \begin{cases} 1, & \text{jika pekerjaan } i \text{ ditugaskan ke pekerja (mesin) } j \\ 0, & \text{jika pekerjaan } i \text{ tidak ditugaskan ke pekerja (mesin) } j \end{cases}$$

Misalkan x_{ij} adalah variabel keputusan penugasan pekerja i ke pekerjaan j , c_{ij} sebagai biaya penugasan pekerja i ke pekerjaan j , maka model pemrograman liniernya adalah:

$$\text{Min/ maks } z = \sum_i \sum_j C_{ij} X_{ij}$$

Terhadap $\sum x_{ij} = 1, i = 1, 2, \dots, n$
 $\sum x_{ij} = 1, j = 1, 2, \dots, n$
 $x_{ij} = 1$ atau 0

i dan j sama karena asumsi dalam modelnya adalah jumlah pekerja sama dengan pekerjaan.

Penentuan solusi optimal dilakukan menggunakan metode Hungarian. Langkah-langkah solusi menggunakan metode Hungarian secara manual adalah sebagai berikut:

1. Menyusun tabel penugasan. Letakkan pekerjaan sebagai baris dan pekerja (mesin) sebagai kolom). Jumlah baris sama dengan jumlah kolom, untuk memenuhi asumsi. Jika tidak sama maka diperlukan dummy.
2. Untuk setiap baris, kurangkan semua nilai dengan dengan nilai terbesar (untuk kasus maksimasi) atau nilai terkecil (untuk kasus minimasi) yang ada pada baris tersebut.
3. Periksa kolom, jika ada kolom yang belum memiliki nilai nol, maka semua nilai pada kolom tersebut dikurangi dengan nil terkecil yang ada pada kolom yang bersangkutan.
4. Periksa apakah solusi layak sudah optimum. Pemeriksaan dilakukan dengan menggambarkan garis-garis vertikal dan horizontal yang melewati nilai nol. Jika jumlah garis yang terbentuk sama dengan jumlah baris/kolom maka solusi layak optimal sudah diperoleh.
5. Jika solusi layak optimal belum diperoleh, kurangkan semua nilai yang tidak dilewati garis dengan nilai terkecil, dan tambahkan nilai terkecil tersebut pada nilai yang terletak pada perpotongan garis. Nilai lainnya (yang dilewati garis tapi tidak terletak pada perpotongan) tidak berubah.
6. Kembali ke langkah 4

Selain penyelesaian masalah penugasan dengan Metode Hungarian, dalam tulisan ini juga diikutkan penyelesaian masalah penugasan dengan menggunakan software LINGO.

Contoh Kasus

Sebuah perusahaan membutuhkan 7 orang tenaga kerja yang akan ditempatkan di 7 jenis posisi yang masih kosong, setelah dilakukan tes terpilihlah 7 orang. Biaya penugasan seorang karyawan untuk pekerjaan yang berbeda adalah berbeda karena sifat pekerjaan berbeda-beda. Setiap karyawan mempunyai tingkat ketrampilan, pengalaman kerja dan latar belakang pendidikan serta latihan yang berbeda pula. Sehingga biaya solusi pekerjaan yang sama oleh para karyawan yang berlainan juga berbeda. Tabel biaya sebagai berikut:

	1	2	3	4	5	6	7
A	1.5	2	1	1.5	2	3	4
B	3	3	1.5	2	2.5	3	2
C	2	1	4	2	1	1	1
D	1.5	1.5	2	3	3	1	1.5
E	4	3	3	1.5	1.5	3	1
F	2	2	3	2	4	1.5	2
G	2.5	2.5	1.5	1	1.5	2	1.

Bagaimanakah seorang manager harus menempatkan tenaga kerjanya agar biaya yang dikeluarkan perusahaan seminimum mungkin.

Solusi:

Banyaknya pekerjaan = banyaknya pekerja, sehingga asumsi sudah dipenuhi

1. Tabel penugasannya

P E K E R J A	P E K E R J A						
	1	2	3	4	5	6	7
A	1.5	2	1	1.5	2	3	4
B	3	3	1.5	2	2.5	3	2
C	2	1	4	2	1	1	1
D	1.5	1.5	2	3	3	1	1.5
E	4	3	3	1.5	1.5	3	1
F	2	2	3	2	4	1.5	2
G	2.5	2.5	1.5	1	1.5	2	1.

2. Pengurangan nilai terkecil pada setiap baris

P E K E R J A A N	P E K E R J A							
		1	2	3	4	5	6	7
A	0.5	1	0	0.5	1	2	3	
B	1.5	1.5	0	0.5	1	1.5	0.5	
C	1	0	3	1	0	0	0	
D	0.5	0.5	1	2	2	0	0.5	
E	3	2	2	0.5	0.5	2	0	
F	0.5	0.5	1.5	0.5	2.5	0	0.5	
G	1.5	1.5	0.5	0	0.5	1	0.5	

3. Berdasarkan tabel di atas, kolom 1 belum memiliki angka 0, dan nilai terkecil pada kolom tersebut adalah 0.5, sehingga semua angka pada kolom 1 dikurangi dengan nilai terkecil pada kolom tersebut yaitu angka 0.5, sehingga tabelnya menjadi seperti berikut:

P E K E R J A A N	P E K E R J A							
		1	2	3	4	5	6	7
A	0	1	0	0.5	1	2	3	
B	1	1.5	0	0.5	1	1.5	0.5	
C	0.5	0	3	1	0	0	0	
D	0	0.5	1	2	2	0	0.5	
E	2.5	2	2	0.5	0.5	2	0	
F	0	0.5	1.5	0.5	2.5	0	0.5	
G	1	1.5	0.5	0	0.5	1	0.5	

4. Berdasarkan tabel di atas, karena banyaknya garis tidak sama dengan banyaknya baris/kolom maka solusi belum optimal, sehingga angka yang tidak terkena garis harus dikurangi dengan angka terkecil yang tidak terkena garis tersebut, dan angka yang terkena coretan sebanyak 2 kali ditambahkan dengan angka terkecil yang tidak terkena garis tersebut. Angka terkecil yang tidak dilalui garis pada tabel di atas adalah angka 0.5

Optimasi Masalah Penugasan

P E K E R J A	P E K E R J A						
		1	2	3	4	5	6
A	0	0.5	0	0	0.5	2	3
B	1	1	0	0	0.5	1.5	0.5
C	1	0	3.5	1	0	0.5	0.5
D	0	0	1	1.5	1.5	0	0.5
E	2.5	1.5	2	0	0	2	0
F	0	0	1.5	0	2	0	0.5
G	1.5	1.5	1	0	0.5	1.5	1

5. Pada tabel di atas banyaknya garis adalah 7 sama dengan banyaknya baris/kolom sehingga dikatakan solusi sudah optimal.

Pembacaan tabel optimlnya dengan mencoba menugaskan pekerja ke pekerjaan dengan nilai pada tabel optimal adalah 0. Berdasarkan nilai 0 pada tabel di atas dapat digambarkan sebagai berikut:

Pekerja	Kemungkinan Pekerjaan	pekerjaan	Biaya
1	A, D, F	A	1.5
2	C, D, F	D	1.5
3	A, B	B	1.5
4	A, B, E, F, G	G	1
5	C, E	C	1
6	D, F	F	1.5
7	E	E	1
Total nilai objektif			9

Masalah penugasan tersebut di atas bisa diselesaikan dengan menggunakan software LINGO. Berikut ini adalah syntaxnya:

sets:

pekerja/1..7/;

pekerjaan/A B C D E F G/;

links(pekerja,pekerjaan):cost,penugasan;

endsets

!fungsi objektif;

min = @sum(links:penugasan*cost);

!kendala;

@for (pekerjaan(j):@sum (pekerja(i):penugasan(i,j))=1;);

@for(pekerja(i):@sum (pekerjaan(j):penugasan(i,j))=1;);

```

data:
cost=1.5 2 1 1.5 2 3 4
3 3 1.5 2 2.5 3 2
2 1 4 2 1 1 1
1.5 1.5 2 3 3 1 1.5
4 3 3 1.5 1.5 3 1
2 2 3 2 4 1.5 2
2.5 2.5 1.5 1 1.5 2 1.5;
enddata
end

```

Hasil running dari syntax di atas adalah sebagai berikut:

```

Global optimal solution found at iteration:      13
Objective value:                               9.000000

```

Variable	Value	Reduced Cost
COST(1, A)	1.500000	0.000000
COST(1, B)	2.000000	0.000000
COST(1, C)	1.000000	0.000000
COST(1, D)	1.500000	0.000000
COST(1, E)	2.000000	0.000000
COST(1, F)	3.000000	0.000000
COST(1, G)	4.000000	0.000000
COST(2, A)	3.000000	0.000000
COST(2, B)	3.000000	0.000000
COST(2, C)	1.500000	0.000000
COST(2, D)	2.000000	0.000000
COST(2, E)	2.500000	0.000000
COST(2, F)	3.000000	0.000000
COST(2, G)	2.000000	0.000000
COST(3, A)	2.000000	0.000000
COST(3, B)	1.000000	0.000000
COST(3, C)	4.000000	0.000000
COST(3, D)	2.000000	0.000000
COST(3, E)	1.000000	0.000000
COST(3, F)	1.000000	0.000000
COST(3, G)	1.000000	0.000000
COST(4, A)	1.500000	0.000000
COST(4, B)	1.500000	0.000000
COST(4, C)	2.000000	0.000000
COST(4, D)	3.000000	0.000000
COST(4, E)	3.000000	0.000000
COST(4, F)	1.000000	0.000000
COST(4, G)	1.500000	0.000000
COST(5, A)	4.000000	0.000000

Optimasi Masalah Penugasan

COST(5, B)	3.000000	0.000000
COST(5, C)	3.000000	0.000000
COST(5, D)	1.500000	0.000000
COST(5, E)	1.500000	0.000000
COST(5, F)	3.000000	0.000000
COST(5, G)	1.000000	0.000000
COST(6, A)	2.000000	0.000000
COST(6, B)	2.000000	0.000000
COST(6, C)	3.000000	0.000000
COST(6, D)	2.000000	0.000000
COST(6, E)	4.000000	0.000000
COST(6, F)	1.500000	0.000000
COST(6, G)	2.000000	0.000000
COST(7, A)	2.500000	0.000000
COST(7, B)	2.500000	0.000000
COST(7, C)	1.500000	0.000000
COST(7, D)	1.000000	0.000000
COST(7, E)	1.500000	0.000000
COST(7, F)	2.000000	0.000000
COST(7, G)	1.500000	0.000000
PENUGASAN(1, A)	1.000000	0.000000
PENUGASAN(1, B)	0.000000	0.500000
PENUGASAN(1, C)	0.000000	0.000000
PENUGASAN(1, D)	0.000000	0.500000
PENUGASAN(1, E)	0.000000	0.500000
PENUGASAN(1, F)	0.000000	2.000000
PENUGASAN(1, G)	0.000000	2.500000
PENUGASAN(2, A)	0.000000	1.000000
PENUGASAN(2, B)	0.000000	1.000000
PENUGASAN(2, C)	1.000000	0.000000
PENUGASAN(2, D)	0.000000	0.500000
PENUGASAN(2, E)	0.000000	0.500000
PENUGASAN(2, F)	0.000000	1.500000
PENUGASAN(2, G)	0.000000	0.000000
PENUGASAN(3, A)	0.000000	1.000000
PENUGASAN(3, B)	0.000000	0.000000
PENUGASAN(3, C)	0.000000	3.500000
PENUGASAN(3, D)	0.000000	1.500000
PENUGASAN(3, E)	1.000000	0.000000
PENUGASAN(3, F)	0.000000	0.500000
PENUGASAN(3, G)	0.000000	0.000000
PENUGASAN(4, A)	0.000000	0.000000
PENUGASAN(4, B)	0.000000	0.000000
PENUGASAN(4, C)	0.000000	1.000000
PENUGASAN(4, D)	0.000000	2.000000
PENUGASAN(4, E)	0.000000	1.500000

PENUGASAN(4, F)	1.000000	0.000000
PENUGASAN(4, G)	0.000000	0.000000
PENUGASAN(5, A)	0.000000	3.000000
PENUGASAN(5, B)	0.000000	2.000000
PENUGASAN(5, C)	0.000000	2.500000
PENUGASAN(5, D)	0.000000	1.000000
PENUGASAN(5, E)	0.000000	0.500000
PENUGASAN(5, F)	0.000000	2.500000
PENUGASAN(5, G)	1.000000	0.000000
PENUGASAN(6, A)	0.000000	0.000000
PENUGASAN(6, B)	1.000000	0.000000
PENUGASAN(6, C)	0.000000	1.500000
PENUGASAN(6, D)	0.000000	0.500000
PENUGASAN(6, E)	0.000000	2.000000
PENUGASAN(6, F)	0.000000	0.000000
PENUGASAN(6, G)	0.000000	0.000000
PENUGASAN(7, A)	0.000000	1.000000
PENUGASAN(7, B)	0.000000	1.000000
PENUGASAN(7, C)	0.000000	0.500000
PENUGASAN(7, D)	1.000000	0.000000
PENUGASAN(7, E)	0.000000	0.000000
PENUGASAN(7, F)	0.000000	1.000000
PENUGASAN(7, G)	0.000000	0.000000

Row	Slack or Surplus	Dual Price
1	9.000000	-1.000000
2	0.000000	-1.500000
3	0.000000	-1.500000
4	0.000000	-1.000000
5	0.000000	-1.000000
6	0.000000	-1.500000
7	0.000000	-1.000000
8	0.000000	-1.500000
9	0.000000	0.000000
10	0.000000	-0.500000
11	0.000000	0.500000
12	0.000000	0.000000
13	0.000000	0.500000
14	0.000000	-0.500000
15	0.000000	0.000000

Hasil dari eksekusi dengan LINGO adalah nilai objektif 9 dengan iterasi sebanyak 13 dan penempatan pekerja sebagai berikut:

Pekerja	pekerjaan	Biaya
1	A	1.5
2	C	1.5
3	E	1
4	F	1
5	G	1
6	B	2
7	D	1
Total nilai objektif		9

Terdapat perbedaan dalam penempatan pekerja antara penyelesaian dengan Metode Hungarian dan eksekusi dengan LINGO akan tetapi nilai objektifnya besarnya sama. Hal ini menunjukkan adanya alternative lain dalam penempatan pekerja di perusahaan.

Kesimpulan

Linier programming merupakan cabang ilmu matematika yang bisa diaplikasikan ke berbagai bidang keilmuan seperti ekonomi, teknologi, farmasi, transportasi, industri dll. Dengan *Linier programming* akan memudahkan pelaku industri untuk menentukan penugasan karyawan yang tepat agar memaksimalkan keuntungan atau meminimalkan biaya. Penggunaan Metode Hungarian bisa membantu dalam pencarian solusi optimal dari masalah penugasan secara manual. LINGO sangat memberi kemudahan kepada pelaku dalam mencari penyelesaian masalah dari pada penghitungan manual.

Daftar Pustaka

- Hotniar Siringoringo. 2005. Seri Teknik Riset Operasional. Yogyakarta. Graha Ilmu.
- Ruminta. 2009. Matriks, persamaan dan pemrograman linier. Bandung. Rekayasa Sains.
- _____. 2006. *Optimization Modeling With LINGO*. Chicago. LINDO Systems, Inc.
- Idris, Endang Lily, Sukamto. Modifikasi Metode Hungaian untuk Menyelesaikan Masalah Penugasan. Jurnal. Universitas Riau.
- N. Soemartojo, Marthen Tapilouw. 1997. Program Linier. Jakarta. Depdiknas.

Siti Maslihah. S.Pd., M.Si
IAIN Walisongo, Semarang
Email :