PERBANDINGAN METODE *HUNGARIAN* DAN PENDEKATAN PROGRAM DINAMIS DALAM PEMECAHAN *ASSIGNMENT PROBLEM*

Budi Marpaung

Fakultas Teknik Jurusan Teknik Industri
Universitas Kristen Krida Wacana
budimarpg_ti@yahoo.com

Abstract

During this assignment problem can be solved only by the Hungarian method. Though highly effective approach to the dynamic program to solve the assignment problem. This paper describes a problem solving assignment for five machines and five jobs. The results showed that both methods give the same solution. Even the difficulty in precisely Hungarian Method can be helped by using a dynamic program approach.

Keywords: *assignment problem, Hungarian Method, dynamic programming, foreward recursive equation, backward recursive equation.*

1. PENDAHULUAN

1.1 Latar Belakang Masalah

Masalah Penugasan *(Assignment Problem)* merupakan kasus khusus dari model transportasi. Model ini menetapkan penggunaan sumber daya, seperti tenaga kerja dan mesin, untuk melaksanakan suatu pekerjaan/*job*, agar diperoleh biaya paling minimum, atau keuntungan maksimum. Selama ini masalah penugasan dipecahkan dengan menggunakan Metode *Hungarian*, bahkan sering dianggap sebagai bagian dari Metode tersebut.

Selain itu Masalah Penugasan sesungguhnya tidak hanya dapat dipecahkan dengan Metode *Hungarian*, karena ada metode lain yang dapat digunakan dan terbukti efektif, yaitu pendekatan program dinamis. Kesulitan untuk mendapatkan solusi optimal pada Metode *Hungarian* dapat diatasi dengan menggunakan pendekatan program dinamis. Namun sejauh ini pendekatan program dinamis tidak lazim digunakan untuk memecahkan masalah penugasan.

Paper ini mencoba memecahkan masalah penugasan dengan Metode *Hungarian* dan pendekatan program dinamis secara bersama-sama. Proses dan hasil keduanya kemudian dibandingkan, untuk selanjutnya dianalisis.

1.2 Perumusan Masalah

Pokok permasalahan yang dibahas dalam *paper* ini adalah menyelesaikan masalah penugasan *(assignment problem)* dengan Metode *Hungarian* dan Pendekatan

Program Dinamis, kemudian membandingkan kedua pendekatan dengan memperhatikan tahapan dan hasil akhirnya.

1.3 Tujuan dan Manfaat Penelitian

Tujuan *paper* ini adalah menemukan karakteristik penyelesaian masalah penugasan dengan Metode *Hungarian* dan Pendekatan Program Dinamis. Pembahasan dalam *paper* ini dapat menjadi masukan dalam penyelesaian masalah penugasan.

1.4 Pembatasan Masalah

Masalah yang dibahas dalam penelitian ini hanya dibatasi pada kondisi mesin tertentu dan setiap mesin hanya mengerjakan *job* tertentu, tidak dapat mengerjakan *job* lainnya.

2. TINJAUAN LITERATUR

2.1 Masalah Penugasan (*Assignment Problem*)

Masalah Penugasan *(Assignment Problem)* pada dasarnya merupakan masalah yang berbentuk program linier. Hal yang dibahas dalam masalah penugasan adalah bagaimana menetapkan penggunaan *supply* yang terbatas untuk memenuhi permintaan. Penetapan penggunaan sumber daya tertentu untuk memenuhi permintaan tertentu akan menimbulkan konsekuensi berupa munculnya biaya yang perlu diminimumkan, atau laba yang perlu dimaksimumkan.

Misalkan sebuah pabrik memiliki n mesin yang dinyatakan oleh M_1, M_2, ..., M_n. Sekumpulan n jenis pekerjaan (*job*) J_1, J_2, ...J_n ditugaskan untuk mesin-mesin tersebut. Untuk setiap *job*, biaya permesinan tergantung pada mesin yang digunakan. Misalnya c_{ij} menyatakan biaya mengerjakan *job* j pada mesin i. Bila sebuah *job* tidak dapat dikerjakan pada satu mesin, maka c_{ij} yang ditetapkan sangat besar. Setiap mesin hanya dapat mengerjakan satu jenis *job*. Persoalan dalam hal ini adalah menugaskan pekerjaan (*job*) kepada mesin sehingga total biaya permesinan menjadi minimum [1].

Pendekatan sederhana untuk menyelesaikan *problem* ini adalah mencoba semua alternatif penugasan *job* ke mesin. Untuk setiap jenis penugasan dihitung total biayanya, demikian seterusnya hingga ditemukan penugasan yang memiliki biaya paling minimum. Pendekatan seperti ini tentu sangat tidak efektif dan efisien, karena jumlah penugasan yang mungkin sebanyak n!. Bayangkan bila n=10, maka akan ada sebanyak 10! = 3.628.800 kemungkinan solusi.

Formulasikan *problem* di atas sebagai program linier [1], defenisikan:

$$x_{ij} = \begin{cases} 1 \text{ bila } job \text{ j ditugaskan ke mesin i} & \text{..... (1)} \\ 0 \text{ untuk yang lain} \end{cases}$$

Karena satu mesin ditugaskan hanya pada satu *job*, maka diperoleh:

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \text{untuk i = 1, 2, ..., n} \qquad \text{..... (2)}$$

Demikian juga setiap *job* ditugaskan untuk satu *job*, maka di dapat

$$\sum_{i=1}^{n} x_{ij} = 1 \text{ untuk j = 1, 2, ..., n} \quad \text{..... (3)}$$

Sedangkan fungsi tujuan adalah meminimumkan:

$$Z = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} x_{ij} \quad \text{.....(4)}$$

Bentuk di atas merupakan formulasi *problem* transportasi standar dengan n gudang dan n pasar, dimana *supply* $a_i = 1$ untuk i = 1,2, ..., n dan *demand* $b_j = 1$ untuk j =1, 2, ...n. Tabel atau matriks transportasi *problem* tersebut diuraikan sebagai berikut:

Tabel 1. Matriks transportasi *problem* penugasan

		Job					
		J_1	J_2	.	.	.	J_n	
Mesin	M_1	c_{11}	c_{12}	.	.	.	c_{1n}	1
	M_2	c_{21}	c_{22}	.	.	.	c_{2n}	1
	.							.
	.							.
	.							.
	M_n	c_{n1}	c_{n2}	.	.	.	c_{nn}	1
		1	1	.	.	.	1	

Kenyataannya masalah penugasan tidak selalu berbentuk standar, dan dalam hal ini terdapat M mesin dan N *job*, dimana $M \neq N$. Untuk mengubah *problem* ini ke bentuk *problem* penugasan standar dengan jumlah mesin dan *job* yang sama, dapat diciptakan mesin *dummy* atau *job dummy*. Andaikan jumlah mesin lebih banyak dari *job* (M > N), maka agar seimbang perlu diciptakan sebanyak (M-N) *job dummy*. Adapun biaya permesinan *job dummy* ditetapkan sebesar nol, sehingga tidak akan mempengaruhi fungsi objektif. Ketika *job dummy* ditugaskan untuk sebuah mesin tertentu, maka mesin itu akan menganggur. Dengan cara yang sama ketika *job* lebih banyak dari mesin (N > M), maka beberapa *job* tidak dapat ditugaskan. Dalam hal ini ditetapkan sebanyak (N-M) mesin *dummy*, dengan biaya permesinan sebesar nol [1].

2.2 Metode *Hungarian*

Metode yang lebih efisien untuk menyelesaikan *problem* penugasan dikembangkan oleh ahli matematika berkebangsaan *Hungarian*, bernama Konig (selanjutnya namanya kadang disebut menjadi nama metode ini). Pada Metode *Hungarian* diasumsikan bahwa semua elemen biaya (c_{ij}) adalah nonnegatif. Prinsip dasar metode ini adalah penugasan optimal tidak terpengaruh bila sebuah bilangan konstanta ditambahkan atau dikurangi pada kolom dan baris matrik penugasan. Misalnya, jika biaya mengerjakan sebuah *job* pada mesin 1 dikurangi sebesar k, maka fungsi objektif *problem* menjadi [1]:

Minimumkan $Z = \sum_{j=1}^{n}(c_{1j} - k) + \sum_{i=2}^{n}\sum_{j=1}^{n} c_{ij} x_{ij}$ .....(5)

$$= \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} - k\sum_{j=1}^{n} x_{ij} \quad \text{.....(6)}$$

Bagaimanapun, $\sum_{j=1}^{n} x_{ij} = 1$, karena mesin 1 harus ditugaskan secara pasti untuk satu *job*. Dengan demikian fungsi objektif baru menjadi:

Z = (objektif awal) – k ....(7)

Prosedur solusi paling penting dalam Metode *Hungarian* adalah mengurangkan sebuah nilai biaya yang sangat besar dari sejumlah kolom atau baris, sehingga penugasan optimal diperoleh melalui pemeriksaan yang cermat. Algoritma pemecahan dimulai dengan menentukan nilai terkecil untuk setiap baris dan kolom. Selanjutnya nilai ini dikurangkan pada setiap elemen baris dan kolom, sehingga matriks biaya memuat angka 0 pada setiap baris dan kolom.Untuk mencoba mendapat solusi layak gunakan kolom dengan biaya nol, demikian seterusnya hingga diperoleh solusi optimal. Hal ini dikarenakan oleh elemen biaya c_{ij} adalah nonnegatif dan nilai minimum fungsi objektif $\sum_i \sum_j c_{ij}x_{ij}$ tidak lebih kecil dari 0. Dengan demikian sudah diperoleh solusi optimal [2].

2.3 Pendekatan Program Dinamis

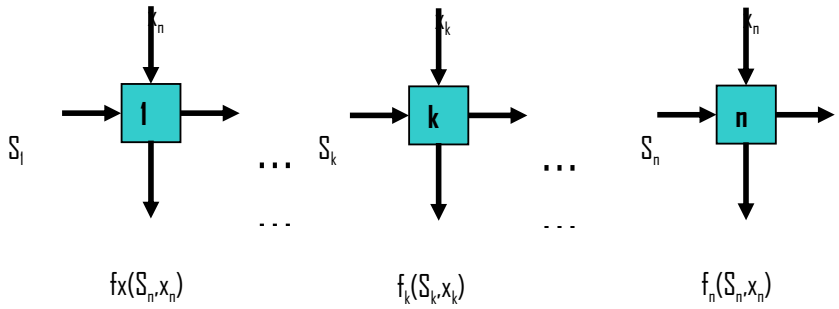
Program Dinamis adalah suatu teknik matematik yang biasa digunakan untuk membuat suatu keputusan dari serangkaian keputusan yang saling berkaitan. Tujuan utama model ini untuk mempermudah penyelesaian persoalan optimasi yang mempunyai karakteristik tertentu. Ide dasar pendekatan program dinamis adalah membagi persoalan menjadi beberapa bagian yang lebih kecil, sehingga memudahkan penyelesaiannya. Berbeda dengan Program Linier yang memiliki formulasi matematis standar, program dinamis tidak memiliki formulasi matematis yang standar. Setiap masalah memiliki struktur penyelesaian yang unik [3].

Program Dinamis dikembangkan pertama sekali oleh Richard Bellman, seorang peneliti pada *The Rand Corporation*, akhir tahun 1940-an. Metode ini kemudian dikembangkan oleh sejumlah ilmuwan lainnya, seperti Dreyfus (1962), Aris (1961), Nemhauser (1964), Wilde (1967), L.G. Mitten (1964), Denardo (1967), Beightler (1976), Don T. Philips (1982), Bertele & Brioschi (1972), Kaufmann (1967), Saaty (1959), dan lain-lain [1]. Ciri-ciri dasar dari masalah program dinamis [4], yaitu:

a. Dalam masalah program dinamis, keputusan tentang suatu masalah ditandai denggan optimasi pada tahap berikutnya, bukan secara serentak. Dalam hal ini masalah diuraikan menjadi beberapa *subproblem*.
b. Program dinamis berkaitan dengan masalah-masalah dimana pilihan atau keputusan dibuat pada masing-masing tahap. Seluruh kemungkinan pilihan ditentukan oleh sistem status pada setiap tahap.
c. Setiap keputusan pada setiap tahap menimbulkan konsekuensi yang dinyatakan dalam *return function* berupa maksimisasi atau minimisasi.
d. Setiap tahap keputusan terhubung dengan tahap yang berdekatan melalui fungsi transisi. Fungsi ini dapat berupa fungsi diskrit atau kontinu tergantung pada masalahnya.
e. Suatu hubungan rekursif digunakan untuk menghubungkan kebijakan yang optimum pada tahap n dengan n-1. Terdapat dua hubungan yang dapat digunakan, yaitu

perhitungan dari depan ke belakang *(foreward recursive equation)* dan perhitungan dari belakang ke depan *(backward recursive equation).*

Keunikan pendekatan Program Dinamis yaitu adanya pemecahan masalah atas beberapa tahapan *(multistage).* Adapun *multistage problem* Program Dinamis, sebagai berikut.



Gambar 1. *Multistage dynamic programming*

3. STUDI KASUS PENUGASAN MESIN

3.1 Gambaran Umum

Terdapat 5 mesin dan 5 *job* dengan waktu proses (dalam menit) sebagai berikut.

Tabel 2. Waktu proses *job* pada mesin

	J_1	J_2	J_3	J_4	J_5
M_1	4	6	2	4	8
M_2	3	7	4	6	7
M_3	5	9	3	3	6
M_4	7	5	3	5	7
M_5	6	2	5	3	8

Setiap mesin memiliki *job* tertentu yang tidak sama dengan mesin lainnya, demikian juga setiap *job* dikerjakan mesin tertentu dan tidak sama dengan *job* lainnya. Dalam hal ini setiap mesin memiliki pasangan *job* yang unik. Masalahnya adalah bagaimana penetapan *job* untuk setiap mesin agar waktu proses secara keseluruhan menjadi minimum.

3.2 Metode *Hungarian*

Karena jumlah mesin dan *job* sama maka *problem* sudah standar. Untuk memperoleh matriks biaya yang sudah terreduksi, maka dilakukan pengurangan berikut pada setiap baris, yaitu 2 dari baris (1); 3 dari baris 2; (3) dari baris (3); 3 dari baris (4); dan 2 dari baris (5). Prosedur ini menghasilkan matriks biaya yang sudah terreduksi, sebagai berikut.

Tabel 3. Matriks biaya telah direduksi pengurangan baris

	J_1	J_2	J_3	J_4	J_5
M_1	2	4	0	2	6
M_2	0	4	1	3	4
M_3	2	6	0	0	3
M_4	4	2	0	2	4
M_5	4	0	3	1	6

Terlihat bahwa mesin M_1 dan M_4 memiliki nilai 0 hanya pada *job* 3, demikian juga dengan *job* 5 tidak memiliki nilai 0 sama sekali. Dengan kondisi seperti ini maka solusi layak dengan biaya nol tidak diperoleh. Untuk memperoleh biaya bernilai nol, kurangkan bilangan terkecil pada setiap kolom, sehingga diperoleh sebagai berikut:

Tabel 4. Matriks biaya telah direduksi pengurangan kolom

	J_1	J_2	J_3	J_4	J_5
M_1	2	4	0	2	3
M_2	0	4	1	3	1
M_3	2	6	0	0	0
M_4	4	2	0	2	1
M_5	4	0	3	1	3

Dari tabel di atas terlihat bahwa M_1 dan M_4 hanya memiliki nilai 0 pada J_3 dan tidak memiliki nilai 0 pada *job* lainnya, sehingga solusi layak tetap belum ditemukan. Dalam hal seperti ini, digunakan prosedur baru, yaitu menggambar garis yang melalui semua angka nol, dengan jumlah garis paling minimum. Jumlah minimum garis sama dengan jumlah maksimum *job* yang sudah memiliki angka 0. Dari contoh di atas diperoleh gambar sebagai berikut:

	J_1	J_2	J_3	J_4	J_5
M_1	2	4	0	2	3
M_2	0	4	1	3	1
M_3	2	6	0	0	0
M_4	4	2	0	2	1
M_5	4	0	3	1	3

Gambar 2. Gambar garis pada matriks biaya

Selanjutnya tentukan bilangan terkecil dari sel yang tersisa (belum digaris), dalam hal ini bernilai 1. Kemudian kurangkan bilangan tersebut terhadap sel yang belum tertutup, dan tambahkan pada sel yang merupakan perpotongan garis. Dengan demikian diperoleh hasil sebagai berikut:

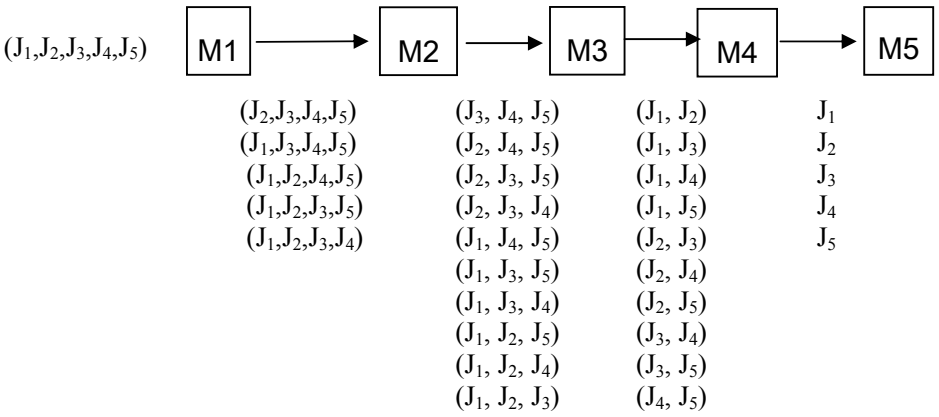
Tabel 5. Solusi optimal

	J_1	J_2	J_3	J_4	J_5
M_1	2	4	0	1	2
M_2	0	4	1	2	0
M_3	3	7	1	0	0
M_4	4	2	0	1	0
M_5	5	0	4	1	3

Solusi yang diperoleh sudah layak dan menjadi solusi optimal, yaitu M_1- J_3; M_2 - J_1; M_3 - J_4; M_4 – J_5 dan M_5 – J_2. Adapun total biaya minimum sebesar (2+3+3+7+2) = 17.

3.3 Pendekatan Program Dinamis

Dengan jumlah mesin dan *job* sebanyak lima buah, maka *stage problem* ini sebanyak 5. Adapun *problem* ini digambarkan sebagai berikut.



Gambar 3. *Multistage dynamic programming problem* lima mesin dan lima *job*

Pemecahan masalah dengan menggunakan pendekatan program dinamis diuraikan pada beberapa tabel berikut ini.

Tabel 6. *Problem 5-stage*

S	$f_5(S, x_5) = f_5^*(s)$					$f_5(*)$	x_5
	J_1	J_2	J_3	J_4	J_5		
J_1	6	-	-	-	-	6	J_1
J_2	-	2	-	-	-	2	J_2
J_3	-	-	5	-	-	5	J_3
J_4	-	-	-	3	-	3	J_4
J_5	-	-	-	-	8	8	J_5

Tabel 7. *Problem 4-stage*

S	$f_4(S, x_4) = f_4^*(s) + f_5^*$					$f_5(*)$	x_4
	J_1	J_2	J_3	J_4	J_5		
J_1, J_2	7+2= 9	5+6=11	-	-	-	9	J_1
J_1, J_3	7+5=12	-	3+6=9	-	-	9	J_3
J_1, J_4	7+3=10	-	-	5+5=11	-	10	J_1
J_1, J_5	7+8=15	-	-	-	7+6=13	13	J_5
J_2, J_3	-	5+5 =10	3+2=5	-	-	5	J_3
J_2, J_4	-	5+3 = 8	-	5+2=9	-	8	J_2
J_2, J_5	-	5+8=13	-	-	7+2=9	9	J_5
J_3, J_4	-	-	3+3=6	5+5=10	-	6	J_3
J_3, J_5	-	-	3+8=11	-	7+5=12	11	J_3
J_4, J_5	-	-	-	5+8=13	7+3=10	10	

Tabel 8. *Problem 3-stage*

S	$f_3(S, x_3) = f_3^*(s) + f_4^*$					$f_3(*)$	x_3
	J_1	J_2	J_3	J_4	J_5		
J_1, J_2, J_3	5+5=10	9+9=18	3+9=12	-	-	10	J_1
J_1, J_2, J_4	5+8=13	9+10=19	-	3+9=12	-	12	J_4
J_1, J_2, J_5	5+9=14	9+13=22	-	-	6+9=15	14	J_1
J_1, J_3, J_4	5+6=11	-	3+10=13	3+9=12	-	11	J_1
J_1, J_3, J_5	5+11=16	-	3+13=16	-	6+9=15	15	J_5
J_1, J_4, J_5	5+10=15	-	-	3+13=16	-	15	J_1
J_2, J_3, J_4	-	9+6=15	3+8=11	3+5=8	-	8	J_4
J_2, J_3, J_5	-	9+11=20	3+9=12	-	6+5=11	11	J_5
J_2, J_4, J_5	-	9+10=19	-	3+9=12	6+8=14	12	J_4
J_3, J_4, J_5	-	-	3+10=13	3+11=14	6+6=12	12	J_5

Tabel 9. *Problem 2-stage*

S	$f_2(S, x_2) = f_2^*(s) + f_3^*$					$f_2(*)$	x_2
	J_1	J_2	J_3	J_4	J_5		
J_1, J_2, J_3, J_4	3+8=11	7+11=18	4+12=16	6+10=16	-	11 `	J_1
J_1, J_2, J_3, J_5	3+11=14	7+15=22	4+14=18	-	7+10=17	14	J_1
J_1, J_2, J_4, J_5	3+12=15	7+15=22	-	6+14=20	7+12=15	15	J_1, J_5
J_1, J_3, J_4, J_5	3+12=15	-	4+15=19	6+15=21	7+11=18	15	J_1
J_2, J_3, J_4, J_5	-	7+12=19	4+12=16	6+11=17	7+8=15	15	J_5

Tabel 10. *Problem 1-stage*

S	$f_1(S, x_1) = f_1^*(s) + f_2^*$					$f_1(*)$	x_1
	J_1	J_2	J_3	J_4	J_5		
$(J_1, J_2, J_3, J_4, J_5)$	4+15=19	6+15=21	2+15=17	4+14=18	8+11=19	` 17	J_3

Solusi optimal yang diperoleh yaitu M_1- J_3; M_2 - J_1; M_3 - J_4; M_4 – J_5 dan M_5 – J_2 sedangkan total biaya minimum sebesar 17.

4. KESIMPULAN

Dari hasil pembahasan di atas diperoleh kesimpulan sebagai berikut:

1. Metode *Hungarian* dan pendekatan Program Dinamis memberikan solusi optimal yang sama.
2. Kedua metode menetapkan penugasan yang optimal, yaitu Mesin 1 mengerjakan *Job* 3, Mesin 2 mengerjakan *Job* 1, Mesin 3 mengerjakan *Job* 4, Mesin 4 mengerjakan *Job* 5, dan Mesin 5 mengerjakan *Job* 2.
3. Total biaya minimum penugasan kelima mesin pada kelima *job* sebesar 17.

REFERENSI

[1]. Don T. Philips, et.al., "*Operation Research: Principle and Practice"*, 2nd edition, John Wiley and Sons, 1987.
[2]. Hillier and Lieberman, "*Introduction to Operations Research"*, Eighth Edition, McGraw-Hill, 2005.
[3]. Dimyati, Tjutju Tarliah, dan Ahmad Dimyati, "*Operations Research, Model-model Pengambilan Keputusan"*, Cetakan 4, Sinar Baru Algensindo, 1999.
[4]. Sri Mulyono, "*Riset Operasi"*, Edisi Revisi, Lembaga Penerbit FE UI, 2007.