PENYELESAIAN MASALAH PENUGASAN YANG DIPERUMUM DENGAN MENGGUNAKAN ALGORITMA *BRANCH-AND-BOUND* YANG DIREVISI

Siti Nur Aisyah[1)], Khusnul Novianingsih[2)], Entit Puspita[3)]

[1), 2), 3)] Departemen Pendidikan Matematika FPMIPA UPI

*Surel: sitinuraisyah5995@gmail.com

ABSTRAK. Masalah penugasan yang diperumum merupakan perumuman dari masalah penugasan klasik. Pada masalah penugasan yang diperumum, satu agen dapat dipasangkan dengan lebih dari satu tugas dan terdapat kendala berupa pembatasan sumber daya yang diberikan kepada setiap agen dalam melakukan tugasnya. Berbasis algoritma *branch-and-bound*, penelitian ini mengembangkan algoritma tersebut untuk menyelesaikan masalah penugasan yang diperumum. Yaitu algoritma *branch-and-bound* yang direvisi yang dibahas dalam penelitian ini. Algoritma ini bekerja dengan cara menyelesaikan serangkaian masalah *knapsack* biner untuk memperbaiki nilai batas bawah dari setiap node, sehingga diperoleh percabangan yang lebih sedikit dibandingkan dengan algoritma *branch-and-bound* biasa. Akibatnya solusi optimal diperoleh dengan waktu yang lebih cepat.

Kata Kunci: Masalah penugasan yang diperumum, algoritma *branch-and-bound* yang direvisi, masalah *knapsack* biner.

ABSTRACT. Generalized assignment problem is a generalization of the classical assignment problem. In the generalized assignment problem, the agent can be assigned with more than one task and there are constraint that is restrictions on the resources provided to each agent for handle the task. Based on branch-and-bound algorithm, this research develops the algorithm for solving the generalized assignment problem. That is revised branch-and-bound algorithm that used in this research. The algorithm works by solving a series of binary knapsack problems to improve the lower bounds of each node. So that we obtain branches less than the branches obtained by the classical branch-and-bound algorithm. Hence the optimal solution obtained in more efficiently.

Keywords: Generalized assignment problem, revised branch-and-bound algorithm, binary knapsack problem.

1. PENDAHULUAN

Masalah penugasan klasik adalah masalah penugasan yang memasangkan setiap agen yang ada dengan satu tugas atau pekerjaan, sehingga tepat satu agen dipasangkan dengan satu tugas. Namun pada kenyataannya bisa juga terjadi penugasan dengan setiap agen mendapatkan lebih dari satu tugas, atau sebaliknya. Masalah di mana setiap agen dapat dipasangkan dengan lebih dari satu tugas disebut dengan "*multiple assignment problem*". Dalam beebrapa kasus penugasan juga terjadi adanya kendala berupa pembatasan sumber daya yang diberikan untuk setiap agennya. Jadi penugasan beberapa tugas bisa diberikan kepada agen *i* jika tugas-tugas tersebut tidak membutuhkan sumber daya lebih dari sumber daya yang tersedia untuk agen *i*. Masalah penugasan ini dapat diwakili oleh model masalah penugasan yang diperumum.

Pada dasarnya masalah penugasan dapat diselesaikan dengan menggunakan algoritma *branch-and-bound*. Namun, seringkali dalam menyelesaikan suatu masalah algoritma *branch-and-bound* menghasilkan dibutuhkan cabang yang sangat banyak sampai diperoleh solusi optimal. Akibatnya dibutuhkan waktu yang lama untuk menemukan solusi optimal. dalam penelitian ini dilakukan sejumlah revisi pada langkah penyelesaiannya agar solusi optimal dapat ditemukan dengan waktu yang lebih cepat.

Tujuan dari penelitian ini yaitu untuk mengetahui cara menyelesaikan masalah penugasan yang diperumum dengan menggunakan algoritma *branch-and-bound* yang direvisi dan implementasi dari contoh kasusnya dalam kehidupan nyata.

Penelitian ini berdasar pada malasah penugasan klasik, yaitu masalah penugasan yang memasangkan setiap agen dengan tepat satu tugas. Menurut Hillier Liebermann (2003, hlm.382), pada masalah penugasan klasik terdapat beberapa asumsi yang harus dipenuhi diantaranya yaitu:

1) Banyaknya agen harus sama dengan banyaknya tugas yang tersedia ($m = n$)
2) Setiap agen dipasangkan dengan tepat satu tugas
3) Setiap tugas dikerjakan oleh tepat satu agen
4) Terdapat biaya c_i yang berkaitan dengan agen *i* saat melakukan tugas *j*
5) Fungsi objektif masalah yaitu untuk menentukan bagaimana semua tugas dapat dilakukan dengan meminimumkan total biaya yang dikeluarkan.

Dalam masalah penugasan terdapat variabel keputusan yang didefinisikan sebagai berikut.

$$x_{ij} = \begin{cases} 1, \text{jika agen } i \text{ ditetapkan untuk tugas } j, \\ 0, \text{jika agen } i \text{ tidak ditetapkan pada tugas } j. \end{cases}$$

Adapun Model optimisasi masalah penugasan klasik adalah sebagai berikut.

Meminimumkan :

$$z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}\, x_{ij} \quad , \qquad (1.1)$$

terhadap :

$$\sum_{j=1}^{n} x_{ij} = 1 \text{ , i= } 1,2,..,m \qquad (1.2)$$

$$\sum_{i=1}^{m} x_{ij} = 1 \text{ , } j = 1,2,..,n \qquad (1.3)$$

$$x_{ij} \in \{0,1\}. \qquad (1.4)$$

2. MODEL OPTIMISASI MASALAH PENUGASAN YANG DIPERUMUM

2.1 Masalah Penugasan yang Diperumum

Masalah penugasan yang diperumum dideskripsikan sebagai masalah pemasangan sejumlah tugas kepada sejumlah agen dengan ketentuan setiap agen dapat dipasangkan pada lebih dari satu tugas dan terdapat batasan sumber daya untuk setiap agen. Model optimisasi masalah penugasan adalah sebagai berikut.

(OP)

Meminimumkan:

$$z = \sum_{i \in I} \sum_{j \in J} c_{i} \, x_{i} \quad , \qquad (2.1)$$

terhadap :

$$\sum_{j \in J} r_{i} \, x_{i} \leq b_{i} \, , i \in I \, , \qquad (2.2)$$

$$\sum_{i \in I} x_{i} = 1 \, , j \in J \, , \qquad (2.3)$$

$$x_{i} \in \{0,1\}. \qquad (2.4)$$

2.2 Metode Penyelesaian

Setiap masalah penugasan pada dasarnya dapat diselesaikan dengan menggunakan algoritma *branch-and-bound*, karena algoritma *branch-and-bound* biasa digunakan untuk menyelesaikan masalah *Integer Linear Programming* (ILP) melalui LP relaksasi dengan membuat sub - sub masalah yang harus diselesaikan. Namun seringkali, penyelesaian masalah ILP dengan metode *branch-and-bound* membutuhkan percabangan yang sangat banyak sampai diperoleh solusi optimal akibatnya dibutuhkan waktu yang lama untuk menemukan solusi optimal. Dalam penelitian ini dilakukan sejumlah revisi pada langkah penyelesaian model. Tujuan dari penggunaan algoritma *branch-and-bound* yang direvisi ini adalah untuk efisiensi waktu penyelesaian masalah penugasan sehingga solusi optimal dapat diperoleh dengan waktu yang lebih cepat.

Algoritma *branch-and-bound* yang direvisi ini pada dasarnya adalah salah satu tipe algoritma *branch-and-bound* biasa di mana nilai batasnya dihitung dengan menyelesaikan masalah *knapsack* biner dalam sub – sub masalah yang ada (G.T. Ross dan R.M. Soland, 1973, hlm.93).

3. HASIL DAN PEMBAHASAN

Pada setiap sub masalah, langkah-langkah yang dilakukan adalah sebagai berikut.

1) Menyelesaikan model relaksasi dari masalah penugasan yang diperumum

   Penyelesaian dari model ini adalah suatu penugasan awal di mana tugas-tugas dipasangkan kepada agen dengan melihat biaya yang paling kecil tanpa memperhatikan kendala (3.5). Model relaksasi dari masalah penugasan yang diperumum adalah sebagai berikut.

(OPR)

Meminimumkan :

$$z = \sum_{i \in I} \sum_{j \in J} c_{i} \; x_{i} \quad ,(2.5)$$

terhadap :

$$\sum_{i \in I} x_{i} \; = \; 1 \; , j \in J \; , (2.6)$$

$$x_{i} \; \epsilon \, \{0,1\}. \qquad (2.7)$$

Solusi dari model OPR diperoleh melalui langkah-langkah berikut.

(1) Tentukan sub himpunan $i_j \in I$ sedemikian sehingga $c_{i_j j} = m \quad {}_{i \in I}\{c_{i} \}$, untuk semua $j \in J$.

(2) Beri nilai variabel keputusan yang didefinisikan seperti berikut.

$$x_{i} \; = \begin{cases} 1, & \text{untuk } i \in i_j, \text{dan } i_j \neq i \;, \\ 0, & \text{untuk } i \text{ lainnya.} \end{cases}$$

Solusi dari masalah OPR menghasilkan nilai batas bawah seperti berikut.

$$z = \sum_{j \in J} c_{i_j j}. \qquad (2.8)$$

2) Dari solusi masalah OPR, selanjutnya dilakukan cek kelayakan terhadap kendala (2.2).

- Jika semua agen *i* memenuhi kendala (2.2) maka solusi OPR menjadi kandidat solusi optimal.
- Jika terdapat agen *i* yang tidak memenuhi kendala (2.2) maka selanjutnya menyelesaikan model *knapsack* biner (PK_i).

Sebelumnya dilakukan pengumpulan sub himpunan sebagai berikut.

(1) Sub himpunan J_i dengan

$$J_i \equiv \left\{ j : \; x_{i} \; = 1, j \in J \right\}, \; \forall \, i \; \in I.$$

Sub himpunan ini berisi tugas-tugas yang dipasangkan pada agen *i*.

(2) Sub himpunan I' dengan

$$I' \equiv \left\{ i : \; \sum_{j \in J_i} r_{i} \, x_{i} \; > b_i \; , i \in I \right\}.$$

Sub himpunan ini berisi agen-agen yang melanggar kendala (2.2).

3) Menyelesaikan model *knapsack* biner (PK_i) untuk setiap $i \in I'$.

Parameter-parameter yang diperlukan dalam model *knapsack* biner (PK_i) adalah sebagai berikut.

(1) p_j, menyatakan penalti minimum yang akan dikeluarkan jika tugas *j* dipindahkan ke agen lain. Parameter ini dihitung sebagai :

$$p_j = \; m \quad {}_{k \in I - \{i_j\}} \left\{ c_k \; - c_{i_j j} \right\}. \qquad (2.9)$$

(2) d_i, menyatakan total sumber daya yang dibutuhkan untuk agen *i* saat melakukan tugas *j* dikurangkan dengan sumber daya yang tersedia untuk agen *i*. Parameter ini dihitung sebagai :

$$d_i = \sum_{j \in J} r_i \; x_i \; - \; b_i. \qquad (2.10)$$

Adapun variabel keputusan pada model *knapsack* biner dinotasikan dengan y_i , untuk setiap $i \in I'$ dan $j \in J_i$. Model masalah *knapsack* biner untuk setiap $i \in I'$ adalah sebagai berikut.

(PK$_i$)

Meminimumkan :

$$z_i = \sum_{j \in J_i} p_j y_i \qquad (2.11)$$

terhadap :

$$\sum_{j \in J_i} r_i \; y_i \;\; \geq \; d_i \,, \qquad (2.12)$$

$$y_i \;\; \epsilon \{0,1\} \,. \qquad (2.13)$$

Solusi optimal dari setiap masalah (PK$_i$) dinotasikan dengan y_i^* , menandakan adanya tugas yang harus dipindahkan ke agen lain dan menghasilkan solusi z_i^* yang dapat memperbaiki nilai batas bawah yang telah diperoleh sebelumnya. Nilai batas bawah yang direvisi untuk masalah penugasan yang diperumum adalah sebagai berikut.

$$L \;\; = z + \sum_{i \in I'} z_i^* \,. \qquad (2.14)$$

Jika solusi baru yang dihasilkan layak terhadap model penugasan yang diperumum maka solusi dikatakan optimal. Jika solusi baru tidak layak maka dilakukan pencabangan dari variabel yang telah ditetapkan penugasannya.

3.1 Contoh

Misalkan terdapat suatu penugasan yang terdiri dari tiga agen dan lima tugas. Adapun daftar biaya yang diperlukan ketika agen *i* melakukan tugas *j* disajikan dalam tabel berikut ini.

Daftar biaya yang diperlukan ketika agen *i* melakukan tugas *j* disajikan dalam tabel berikut ini.

Tabel 3.1 Koefisien Fungsi Objektif (c_i)

Agen	Tugas				
	1	2	3	4	5
1	14	38	*	26	14
2	49	*	20	46	49
3	*	45	16	10	*

Koefisien sumber daya yang dibutuhkan saat agen *i* melakukan tugas *j* disajikan dalam tabel berikut.

Tabel 3.2 Koefisien penugasan banyak (r_i)

Agen	Tugas				
	1	2	3	4	5
1	12	19	*	11	18
2	6	*	11	15	18
3	*	10	14	22	*

Tanda (*) pada kedua tabel di atas menunjukkan bahwa pasangan tugas-agen yang bersangkutan tidak diijinkan. Dalam contoh ini terdapat suatu pembatasan sumber daya dalam penugasan tersebut. Dan setiap agen diberikan sumber daya 28 (b_i). Tentukan pemasangan yang tepat sehingga mendapatkan hasil penugasan yang optimal.

Permasalahan di atas dapat dimodelkan ke dalam model masalah penugasan yang diperumum seperti berikut.

(OP 1)

Meminimumkan :

$$z = 14x_1 + 38x_1 + 26x_1 + 14x_1 + 49x_2 + 20x_2 + \cdots + 10x_3 \ ,$$

terhadap :

$$12x_1 + 19x_1 + 11x_1 + 18x_1 \leq 28 \ , \quad (1)$$
$$6x_2 + 11x_2 + 15x_2 + 18x_2 \leq 28 \ , \quad (2)$$
$$10x_3 + 14x_3 + 22x_3 \leq 28 \ , \quad (3)$$
$$x_1 + x_2 + x_3 = 1 \ , \quad (4)$$
$$x_1 + x_2 + x_3 = 1 \ , \quad (5)$$
$$x_1 + x_2 + x_3 = 1 \ , \quad (6)$$
$$x_1 + x_2 + x_3 = 1 \ , \quad (7)$$
$$x_1 + x_2 + x_3 = 1 \ , \quad (8)$$
$$x_i \in \{0,1\}.$$

1) Menyelesaikan model (OPR 1)

(OPR 1)

Meminimumkan :

$$z = 14x_1 + 38x_1 + 26x_1 + 14x_1 + 49x_2 + 20x_2 + \cdots + 10x_3 \ ,$$

terhadap :

$$x_1 + x_2 + x_3 = 1 \ , (1)$$
$$x_1 + x_2 + x_3 = 1 \ , (2)$$
$$x_1 + x_2 + x_3 = 1 \ , (3)$$
$$x_1 + x_2 + x_3 = 1 \ , (4)$$
$$x_1 + x_2 + x_3 = 1 \ , (5)$$
$$x_i \in \{0,1\}.$$

Diperoleh solusi $x_1 = x_1 = x_3 = x_3 = x_1 = 1$ dan nilai batas bawah awal $z = 92$.

Dari solusi model (OPR 1), agen 1 dan 3 tidak memenuhi kendala (2.2). Maka selanjutnya menyelesaikan model *knapsack* biner (PK_i).

(PK_1)

Meminimumkan :

$$z_1 = 35y_1 + 7y_1 + 35y_1 \ ,$$

terhadap :

$$12y_1 + 19y_1 + 18y_1 \geq 21 \ ,$$
$y_{1j} \in \{0,1\}$, untuk semua *j*.

(PK_3)

Meminimumkan :

$$z_3 = 4y_3 + 16y_3$$

terhadap :

$$14y_3 + 22y_3 \geq 8 \ ,$$
$y_{3j} \in \{0,1\}$, untuk semua *j*.

Dari model-model (PK_1) dan (PK_3) di atas diperoleh solusi optimal $y_1^* = 0$, $y_1^* = 1$, $y_1^* = 1$, dan $y_3^* = 1$, $y_3^* = 0$. Dan menghasilkan nilai batas bawah yang direvisi $L = 132$.

Variabel x_1 dipilih sebagai variabel yang akan dipisahkan dengan kandidat masalah baru $x_1 = 1$ dan $x_1 = 0$. Untuk $x_1 = 1$ diperoleh model relaksasi sebagai berikut.

(OPR 2)

Meminimumkan :

$$z = 14x_1 + 26x_1 + 49x_2 + 20x_2 + \cdots + 10x_3 ,$$

terhadap :

$$x_1 + x_2 + x_3 = 1 , \quad (1)$$
$$x_2 + x_3 = 1 , \quad (2)$$
$$x_1 + x_2 + x_3 = 1 , \quad (3)$$
$$x_1 + x_2 + x_3 = 1 , \quad (4)$$
$$x_2 + x_3 = 1 , \quad (5)$$
$$x_i \in \{0,1\}.$$

Diperoleh $x_1 = x_3 = x_3 = x_3 = x_2 = 1$ dan nilai batas bawah awal $z = 134$. Dari solusi model (OPR 2), agen 3 tidak memenuhi kendala (2.2). Maka selanjutnya menyelesaikan model *knapsack* biner (PK_i).

(PK_3)

Meminimumkan :

$$z_3 = \infty y_3 + 4y_3 + 16y_3 ,$$

terhadap :

$$10y_3 + 14y_3 + 22y_3 \geq 18,$$
$$y_{3j} \in \{0,1\} \text{ , untuk semua } j.$$

Dari model (PK_3) di atas diperoleh solusi optimal $y_3^* = 0$, $y_3^* = 0$, dan $y_3^* = 1$. Dan menghasilkan nilai batas bawah yang direvisi $L = 150$. Ketika tugas 4 dipindahkan ke agen 1 maka akan diperoleh solusi yang layak terhadap model penugasan yang diperumum. Sehingga diperoleh solusi $x_1 = x_3 = x_3 = x_1 = x_2 = 1$ dengan nilai fungsi objektif adalah 150. Karena solusi pada sub masalah ini telah layak maka kandidat pada sub masalah ini adalah *fathomed.*

Untuk $x_1 = 0$ diperoleh model relaksasi sebagai berikut.

(OPR 3)

Meminimumkan :

$$z = 38x_1 + 26x_1 + 14x_1 + 49x_2 + 20x_2 + \cdots + 10x_3 ,$$

terhadap :

$$x_2 + x_3 = 1 , \quad (1)$$
$$x_1 + x_2 + x_3 = 1 , \quad (2)$$
$$x_1 + x_2 + x_3 = 1 , \quad (3)$$
$$x_1 + x_2 + x_3 = 1 , \quad (4)$$
$$x_1 + x_2 + x_3 = 1 , \quad (5)$$

$x_i \in \{0,1\}$.

Diperoleh $x_2 = x_1 = x_3 = x_3 = x_1 = 1$ dan nilai batas bawah awal $z = 127$. Dari solusi model (OPR 3), agen 1 dan 3 tidak memenuhi kendala (2.2). Maka selanjutnya menyelesaikan model *knapsack* biner (PK_i).

(PK_1)

Meminimumkan :

$$z_1 = 7y_1 + 35y_1 ,$$

terhadap :

$$19y_1 + 18y_1 \geq 9 ,$$

$$y_{1j} \in \{0,1\} \text{ , untuk semua } j.$$

(PK_3)

Meminimumkan :

$$z_3 = 4y_3 + 16y_3 ,$$

terhadap :

$$14y_3 + 22y_3 \geq 8 ,$$

$$y_{3j} \in \{0,1\} \text{ , untuk semua } j.$$

Dari model (PK_1) dan (PK_3) maka diperoleh solusi optimal $y_1^* = 0$, $y_1^* = 1$, dan $y_3^* = 1$, $y_3^* = 0$. Dan menghasilkan nilai batas bawah yang direvisi $L = 138$.

Variabel x_1 dipilih sebagai variabel yang akan dipisahkan. Untuk $x_1 = 1$ diperoleh model relaksasi sebagai berikut.

(OPR 4)

Meminimumkan :

$$z = 26x_1 + 14x_1 + 49x_2 + 20x_2 + \cdots + 10x_3 ,$$

terhadap :

$$x_2 + x_3 = 1 , (1)$$
$$x_2 + x_3 = 1 , (2)$$
$$x_1 + x_2 + x_3 = 1 , \quad (3)$$
$$x_1 + x_2 + x_3 = 1 , \quad (4)$$
$$x_1 + x_2 + x_3 = 1 , \quad (5)$$

$$x_i \in \{0,1\}.$$

Diperoleh $x_2 = x_3 = x_3 = x_3 = x_1 = 1$ dan nilai batas bawah awal $z = 134$. Dari solusi model (OPR 4), agen 3 tidak memenuhi kendala (2.2). Maka selanjutnya menyelesaikan model *knapsack* biner (PK_i).

(PK$_3$)

Meminimumkan :

$$z_3 = \infty y_3 \;\; + 4y_3 \;\; + 16y_3 \;\; ,$$

terhadap :

$$10y_3 \;\; + 14y_3 \;\; + 22y_3 \;\; \geq \; 18 \,,$$

$$y_{3j} \in \{0,1\} \text{ , untuk semua } j.$$

Dari model (PK$_3$) di atas maka diperoleh solusi optimal $y_3^* = 0$, $y_3^* = 0$, dan $y_3^* = 1$. Dan menghasilkan nilai batas bawah yang direvisi $L = 150$. Ketika tugas 4 dipindahkan ke agen 2 maka akan diperoleh solusi yang layak terhadap model penugasan yang diperumum. Sehingga diperoleh solusi $x_2 = x_3 = x_3 = x_2 = x_1 = 1$ dengan nilai fungsi objektif adalah 170. Solusi pada sub masalah ini telah layak maka kandidat pada sub masalah ini adalah *fathomed*.

Untuk $x_1 = 0$ diperoleh model relaksasi sebagai berikut.

(OPR 5)

Meminimumkan :

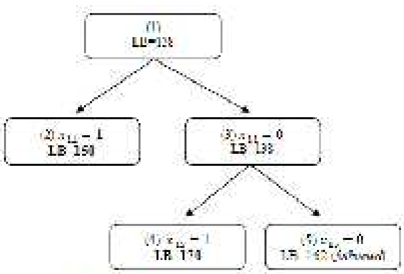
$z = 38x_1 \;\; + 26x_1 \;\; + 49x_2 \;\; + 20x_2 \;\; + \cdots + 10x_3 \;\; ,$

terhadap :

$$
\begin{aligned}
&x_2 + x_3 = 1 \,, && (1)\\
&x_1 + x_2 + x_3 = 1 \,, && (2)\\
&x_1 + x_2 + x_3 = 1 \,, && (3)\\
&x_1 + x_2 + x_3 = 1 \,, && (4)\\
&x_2 + x_3 = 1 \,, && (5)\\
&x_i \in \{0,1\}.
\end{aligned}
$$

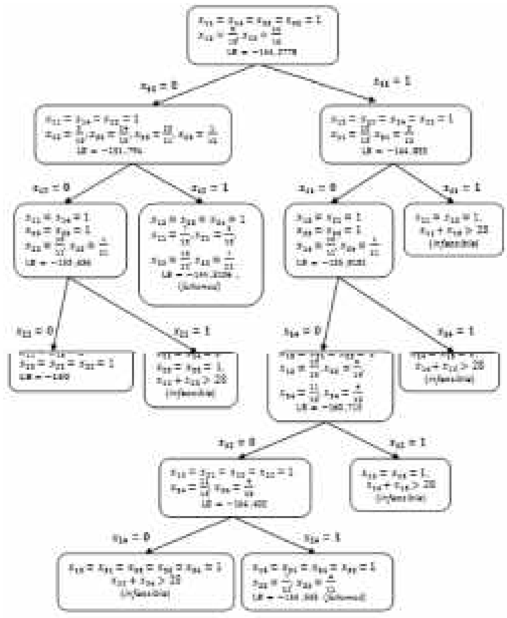
Diperoleh $x_2 = x_1 = x_3 = x_3 = x_2 = 1$ dan nilai batas bawah awal $z = 162$. Dari solusi model (OPR 5), agen 3 tidak memenuhi kendala (2.2). Sub masalah ini *fathomed*.

Dalam sub masalah ini dipilih solusi optimal $x_1 = x_3 = x_3 = x_1 = x_2 = 1$ dengan nilai fungsi objektif adalah 150. Selengkapnya pohon *branch-and-bound* dari masalah di atas diilustrasikan pada gambar berikut.



Gambar 3.1 Pohon *branch-and-bound* yang direvisi

Jika masalah ini diselesaikan dengan algoritma *branch-and-bound* biasa, maka pohon *branch-and-bound* yang dihasilkan adalah sebagai berikut.



Gambar 3.2 Pohon *branch-and-bound* biasa

4. KESIMPULAN

Berdasarkan pembahasan yang telah dilakukan, maka diperoleh kesimpulan bahwa menyelesaikan masalah penugasan yang diperoleh dengan menggunakan algoritma *branch-and-bound* yang direvisi diperoleh dengan langkah-langkah sebagai berikut.

1) Memodelkan masalah penugasan ke dalam model umum masalah penugasan yang diperumum
2) Mengidentifikasi pemasangan agen-tugas dengan biaya terkecil untuk setiap tugas sehingga menghasilkan batas bawah awal
3) Diidentifikasi agen yang melanggar kendala pembatasan sumber daya
4) Untuk setiap agen yang melanggar kendala, masing-masing agen dibuatkan model *knapsack* biner.
5) Solusi dari model *knapsack* biner menunjukkan perlunya dilakukan penugasan kembali.

- Jika terdapat satu tugas yang harus dipindahkan, maka dilakukan penugasan ulang secara langsung. Dan solusi menjadi kandidat solusi optimal.

- Jika terdapat lebih dari satu tugas yang harus dipindahkan, maka dilakukan pencabangan pada variabel yang telah ditetapkan penugasannya.

Dan telah ditunjukkan bahwa algoritma *branch-and-bound* yang direvisi dapat mengurangi banyaknya percabangan masalah dibandingkan dengan algoritma *branch-and-bound* biasa.

5. DAFTAR PUSTAKA

[1] Hillier, F. S., & Lieberman G. J. (2001). *Introduction to operation research.* Edisi ke 7. New York: McGraw-Hill Companies.

[2] Putri, E.A. (2014). *Aplikasi pengambilan keputusan dalam persoalan penugasan multikriteria*. (Skripsi). Universitas Pendidikan Indonesia, Bandung.

[3] Ross, G.T., & Soland, R.M. (1975). A branch and bound algorithm for the generalized assigment problem. *Mathematical Programming, 8 (1),* hlm. 91-103.

[4] Winston, W.L. (2003). *Operation research: Application and algorithms.* Edisi ke 4. USA: Duxbury Press.