MEMINIMUMKAN BIAYA TRANSPORTASI MENGGUNAKAN METODE BASIS TREE

Maxsi Ary
Program Studi Sistem Informasi
Universitas BSI Bandung
Jl.Sekolah Internasional No.1-6 Antapani – Bandung, Indonesia
maxsiary@gmail.com

Abstract

Transportation problem is the linear programming problem. The main purpose is minimize the cost of distribution. In present day the solution of transportation problem are North-West Corner, Least Cost, and Vogel's Approximation (VAM). In 1946-1947 G.B Dantzig proposes the research and designed the MODI (Modified For Distribution), the method could be applied to transportation problem. The MODI method and other method that use a transportation table are adequate as long as the problem is relatively small. Once the problem becomes large, then finding the unique $\theta - loop$ *and performing the updating is difficult. This study was made to find solutions to transportation problems using basis tree. Is a literature study that the authors use to conduct research, with the source form of journal papers, articles and textbooks. The Basis Tree Transportation Problem is a specialization for transportation problem. The key idea in this method is that any basic feasible solution of transportation problem is a spanning tree of the underlying graph. Hence for each iteration, the basis is represent as a rooted spanning tree in which an arc* (i, j) *and its flow* x_{ij} *represent the basic variable* x_{ij}, *and the simplex multiplier (dual variable) are represent by node potential. Use of Nort-West Corner aimlessly total freight 1490, and the results of the iteration will be obtained a minimum total transportation cost 920. Iterations that use basis tree using five iterations. It can be noted that minimizes transportation costs using the basis tree method resulted in a difference of 570.*

Keywords: North-West Corner, Basis Tree, Transportation Problem.

1. PENDAHULUAN

Persoalan transportasi merupakan persoalan program linear, yang membahas masalah pendistribusian suatu komoditas atau produk dari sejumlah sumber (*supply*) kepada sejumlah tujuan (*destination*, *demand*), dengan tujuan meminimumkan biaya pengangkutan yang terjadi.

Selama ini persoalan transportasi dilakukan dengan menggunakan metode *North-West Corner* (NWC), *least cost*, dan *Vogel's Approximation Method* (VAM). Penulis mengguanakan metode NWC untuk menentukan solusi fisibel basis awal. Solusi ini merupakan bagian penting dalam menyelesaikan persoalan transportasi, karena sebagai langkah awal dalam *Network* Metode Simpleks. Network metode simpleks pertama kali dipelajari dalam sebuah penelitian oleh G.B Dantzig (1946-1947). Dantzig bekerja sebagai mekanik perencanaan proses angkatan udara Amerika Serikat. Sekarang penelitian itu diaplikasikan terhadap persoalan transportasi yang dikenal dengan MODI (*Modified for Distribution*)[1] (O'Connor, 2001: p.4.22). Metode MODI dan metode lainnya yang menggunakan tabel transportasi cukup memadai selama masalah relatif kecil. Permasalahan menjadi besar, sehingga mencari unik $\theta loop$ dan membentuk solusi fisibel basis yang baru menjadi sulit.

Untuk alasan ini dan alasan lain metode standar pemecahan masalah tranportasi dan masalah arus *network* telah menjadi algoritma *out-of-killer* (O'Connor, 2001, hal: 4.30), sehingga persoalan transportasi merupakan kasus yang menarik.

Pengembangan terstruktur untuk algoritma *out-of-killer* terinspirasi oleh perkembangan terbaru teknik ilmu komputer

menggunakan struktur data dan manipulasi data. Gagasan ini memberikan gambaran terstruktur sehingga persoalan yang besar (memerlukan memori yang besar) dapat disimpan pada memori komputer.

Sebagai kelanjutannya, gambaran terstruktur memiliki arti bahwa algoritma mempunyai proses kerja yang lebih sedikit, karena persoalan ditunjukkan secara terstruktur.

2. TINJAUAN PUSTAKA

2.1 Persoalan Transportasi

Persoalan transportasi merupakan persoalan program linear. Persoalan ini membahas masalah pendistribusian suatu komoditas atau produk dari sejumlah sumber (supply) kepada sejumlah tujuan (destination, demand), dengan tujuan meminimumkan ongkos pengangkutan yang terjadi. Berikut formulasi dan spesialisasi persoalan transportasi.

2.1.1 Formulasi dan Spesialisasi

Misalkan $G(N,A)$ adalah *network* berarah terdiri dari himpunan berhingga *node* N, dan himpunan *arc* berarah A menghubungkan pasangan *node* pada N. Setiap *arc* $(i,j) \in A$, sebuah komoditas x_{ij}, ongkos pendistribusian per unit c_{ij}, batas bawah komoditas l_{ij} dan batas atas u_{ij}. Untuk setiap *node* $i \in A$ kita memberikan nilai integer b_i menunjukkan sumber yang tersedia atau tujuan untuk komoditas pada *node* tersebut. Jika $b_i > 0$, maka *node* i adalah *node* sumber. Jika $b_i < 0$, maka *node* i adalah *node* tujuan. Sebaliknya jika $b_i = 0$, *node* i diserahkan kepada *node transshipment*. Total jumlah komoditas sumber harus sama dengan total permintaan tujuan atau total jumlah komoditas sumber haruslah nol.

$$\sum b_i = 0, i \in N \qquad \textbf{(2. 1)}$$

Formulasi persoalan program linear:

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij} \qquad \textbf{(2. 2)}$$

Berdasarkan Pembatas:

$$\sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = b_i \qquad \textbf{(2. 3)}$$

Untuk semua $i \in A$

$$0 \le l_{ij} \le x_{ij} \le u_{ij} \qquad \textbf{(2. 4)}$$

Untuk semua $(i,j) \in A$

Formulasi persoalan program linear ditujukan untuk mengirimkan produk permintaan dari *node-node* sumber kepada *node-node* tujuan. Sedemikian sehingga pembatas tujuan (2.3), ongkos minimum (2.2), pembatas *flow bound* (2.4), haruslah terpenuhi. Pembatas tujuan juga disenut sebagai *mass balance* atau pembatas *flow balance*.

Total jumlah komoditas sumber haruslah sama dengan nol dan menjumlahkan persamaan flow balance untuk semua $i \in N$ sehingga didapatkan:

$$\sum_{i\in N}\left(\sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = b_i\right) = \sum_{i\in N} b_i = 0$$

(2. 5)

Ini artinya bahwa pembatas (2.3) adalah bergantung linear. Persoalan dalam notasi matriks adalah:

$$\min\{cx \mid Nx = b \; dan \; 0 \le l \le x \le u\}$$

Di mana N adalah matriks tetangga dari *node-node*, mempunyai baris untuk setiap *node* dan kolom untuk setiap *arc*.

Formulasi persoalan program linear sangatlah istimewa dan luas penggunaannya dan dapat digunakan pada persoalan *network* dengan model tertentu.

A. Persoalan Transshipment

Model *transshipment* adalah model transportasi yang memungkinkan dilakukan pengirim (komoditas) dengan cara tidak langsung, di mana komoditas dari suatu sumber dapat berada pada sumber lain atau tujuan lain sebelum mencapai tujuan akhir. Jadi, pada persoalan *transshipment* ini suatu sumber sekaligus dapat berperan sebagai tujuan dan, sebaliknya, suatu tujuan dapat juga berperan sebagai sumber.

Persoalan *transshipment* adalah persoalan program linear dengan *node transshipment* dan kapasitas *arc* tak terbatas. Kita membagi *node* menjadi tiga subhimpunan. N_1 adalah himpunan *node* sumber, N_2 adalah himpunan *node* tujuan, dan N_3 adalah subhimpunan *node transshipment*.

Formulasi program linearnya adalah:

$$\min \sum_{(i,j)\in A} c_{ij}x_{ij} \quad \text{.....(2. 6)}$$

Berdasarkan pembatas:

$$\sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = b_i \quad \text{...(2. 7)}$$

Untuk semua $i \in N_1, N_2$

$$\sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = 0 \quad \text{...(2. 8)}$$

Untuk semua $j \in N_3$

$x_{ij} \geq 0$; untuk semua $(i,j) \in A$

.....(2. 9)

B. Persoalan Transportasi

Kasus special dari persoalan *transshipment* di mana tidak terdapat *node transshipment* disebut dengan persoalan transportasi. *Node* disini adalah salah satu dari *node* sumber dan *node* tujuan. Ini artinya bahwa graph yang mendasari adalah *graph bipartite*, di mana N_1 adalah himpunan *node* sumber dan N_2 adalah himpunan *node* tujuan, sedemikian sehingga $N = N_1 \cup N_2$ dan himpunan *arc* didefinisikan oleh:

$$A = \{(i,j) \mid i \in N_1, j \in N_2\}$$

Secara objektif yaitu untuk merencanakan ongkos pengiriman yang terkecil dari sumber N_1 kepada tujuan N_2, di mana x_{ij} adalah banyaknya unit dikirim dari sumber i kepada tujuan j, dan c_{ij} adalah ongkos pengiriman satu unit i kepada j.

Sumber dan tujuan berturut-turut ditunjukkan oleh b_{ij}. Tidak terdapat batas atas (*uncapacitated*).

Formulasi persoalan program linear:

$$\min \sum_{(i,j)\in A} c_{ij}x_{ij} \quad \textbf{(2. 10)}$$

Berdasarkan pembatas:

$$\sum_{j:(i,j)\in A} x_{ij} = b_i; \text{ untuk semua } i \in N_1 \quad \textbf{(2. 11)}$$

$$\sum_{j:(i,j)\in A} x_{ij} = -b_i; \text{ untuk semua } i \in N_2 \quad \textbf{(2. 12)}$$

$x_{ij} \geq 0$; untuk semua $(i,j) \in A$ **(2. 13)**

2.1.2 Formulasi Persoalan Transportasi

Persoalan transportasi merupakan persoalan yang sederhana, karena alas an berikut:

a) Merupakan *graph bipartite* dengan tidak memiliki *node-node transshipment*.
b) Tidak mempunyai batas-batas kapasitas pada *arc-arc*.

Formula standar secara aljabar untuk persoalan transportasi:

$$z = \min \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \quad \textbf{(2. 14)}$$

Berdasarkan pembatas:

$$\sum_{j=1}^{n} x_{ij} = S_i, \; i = 1,2,...,m \quad \textbf{(2. 15)}$$

$$\sum_{i=1}^{m} x_{ij} = D_j, \; j = 1,2,...,n \quad \textbf{(2. 16)}$$

$x_{ij} \geq 0$; untuk semua (i,j) **(2. 17)**

di mana:

$S_i > 0, D_j > 0$ untuk semua i, j

dan $\sum_{j=1}^{n} S_i = \sum_{i=1}^{m} D_j$, dengan lain kata *total supplies = total demand*.

Kondisi terakhir ini sangat penting dan menentukan untuk menjadikan suatu masalah transportasi menjadi fisibel. Interpretasi dari persoalan transportasi:

1. Terdapat m sumber masing-masing memproduksi produk yang sama.
2. Sumber i memproduksi sejumlah S_i
3. Terdapat n tujuan masing-masing meminta produk yang sama.
4. Tujuan j memproduksi sejumlah D_j
5. x_{ij} adalah jumlah produk dikirim dari sumber i ke tujuan j dengan harga c_{ij} untuk setiap unit pengiriman.
6. Sumber tidak dapat mengirim lebih dari hasil produksinya, akibatnya terdapat pembatas *supply*. Contoh sumber ke-2:

$$\sum_{j=1}^{n} x_{2j} = S_2$$

7. Tujuan tidak dapat menerima lebih dari yang di minta, akibatnya terdapat pembatas *demand*. Contoh tujuan ke-4:

$$\sum_{i=1}^{m} x_{i4} = D_4$$

8. Jika sumber memiliki kelebihan permintaan, kita kirim semua kelebihan ke variabel bantu yang disebut variabel *dummy*. Tujuan $n+1$ dengan biaya nol.
9. Jika permintaan melebihi sumber, masalah tidak dapat diselesaikan sebagai persoalan transportasi, maka dibutuhkan formulasi yang baru.

2.1.3 Sifat-Sifat Persoalan Transportasi

Bentuk dasar dari program linear umum adalah:

$z = \min_{x \in R^n} cx$, berdasarkan pembatas $Ax = b$; $x \geq 0$

di mana b, c dan x adalah $mx1$, $1xn$ dan $nx1$ berturut-turut adalah vector, dan A adalah matrik mxn. Untuk persoalan transportasi dengan m sumber dan n tujuan, m menjadi $m+n$ dan n menjadi mn, sehingga kita dapat:

$$\begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} & c_{21} & c_{22} & \cdots & c_{2n} & \cdots & c_{m1} & c_{m2} & \cdots & c_{mn} \\ 1 & 1 & \cdots & 1 & & & & & & & & & \\ & & & & 1 & 1 & \cdots & 1 & & & & & \\ & & & & & & & & \cdots & & & & \\ & & & & & & & & & 1 & 1 & \cdots & 1 \\ 1 & & & & & 1 & & & & 1 & & & \\ & 1 & & & & & & 1 & & & 1 & & \\ & & \ddots & & & & & & \ddots & & & \ddots & \\ & & & 1 & & & & & 1 & & & & 1 \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \\ \vdots \\ x_{1n} \\ x_{21} \\ x_{22} \\ \vdots \\ x_{2n} \\ \vdots \\ x_{m1} \\ x_{m2} \\ \vdots \\ x_{mn} \end{bmatrix} = \begin{bmatrix} z \\ S_1 \\ S_2 \\ \vdots \\ S_m \\ D_1 \\ D_2 \\ \vdots \\ D_n \end{bmatrix}$$

Sumber: O'Connor, 2001: p.4.29

Gambar 2. 1 Bentuk Matrix Pembatas $Ax = b$

Matriks A adalah matrik node-arc dari graph bipartite $G = (N_1, N_2, A, c)$, di mana $|N_1| = m$, $|N_2| = n$, dan $|A| = mn$.

Setiap program linear dengan matrik A mempunyai sifat-sifat:

1. Persoalan program linear dengan mxn variabel x_{ij} dan $m+n$ pembatas.
2. Terdapat tepatnya satu pembatas yang berlebih. Pembatas tersebut boleh dihilangkan sehingga kelebihan dapat dikeluarkan.
3. Vektor basis terdiri dari $(m+n-1)$ variabel bebas.
4. Jika semua S_i dan D_j adalah bilangan bulat positif, maka semua solusi basis adalah bilangan bulat dan karenanya solusi optimum adalah bilangan bulat.
5. Matrik A dan kemungkinan vektor c adalah sangat jarang. Tentu saja matrik A mempunyai $(m+n)xmn$ elemen dan hanya $2mn$ adalah *non-zero*. Dalam pemahamannya *non-zero* ini adalah 1 dan mempunyai struktur yang sangat spesial.

Semua sifat-sifat memberikan kontribusi untuk membuat persoalan transportasi menjadi persoalan program linear yang sangat special.

2.1.4 Dual Persoalan Transportasi

Dual persoalan transportasi sangatlah penting karena kita gunakan variabel dual dalam perhitungan penurunan ongkos transportasi pada persoalan primal:

$$\max \sum_{i=1}^{m} S_i u_i + \sum_{j=1}^{n} D_j v_j \qquad \textbf{(2. 18)}$$

Berdasarkan pembatas:

$u_i + v_j \le c_{ij}$, untuk semua i, j **(2. 19)**

u_i dan v_j tertutup. **(2. 20)**

Akan ditunjukkan bagaimana pembatas dual digunakan dalam menghitung penurunan ongkos transportasi $\bar{c}_{ij} = c_{ij} - (u_i + v_j)$ untuk variabel nonbasis. Perlu diingat bahwa terdapat satu pembatas dual untuk setiap *arc* dalam *network* persoalan transportasi.

Untuk menghitung penurunan ongkos transportasi, kita memerlukan *simplek multipliers* atau *node potensial* u_i dan v_j untuk $i = 1,...,m$ dan $j = 1,...,n$. Jika x_{ij} adalah variabel basis, maka $u_i + v_j = c_{ij}$. Ini akan memberikan $m+n-1$ persamaan dengan $m+n$ yang tidak diketahui. Karena satu pembatas yang berlebihan, kita dapat memilih nilai tertentu untuk setiap salah satu u dan v. Pilihan $u_i = 0$, kemudian selesaikan variabel yang lain dari $m+n-1$ persamaan tersebut dengan cara substitusi.

Untuk setiap variabel nonbasis kita dapat $u_i - v_j < c_{ij}$ atau $u_i - v_j + \bar{c}_{ij} = c_{ij}$. Karenanya $\bar{c}_{ij} = c_{ij} - (u_i + v_j)$ dapat dihitung untuk semua variabel nonbasis.

2.2 Transportasi Basis Tree

Metode MODI dan metode lainnya yang menggunakan tabel transportasi cukup memadai selama permasalahan relative kecil. Jika persoalan dengan sumber > 100 dan tujuan > 100, kemudian mencari unik loop menjadi sangat sulit dan membentuk solusi fisibel basis yang baru menjadi membosankan.

2.2.1 Basis Tree

Jika kita perhatikan tabel transportasi *3 x 4* hasil aturan *North West Corner* (NWC) yang diberikan sebagai contoh seperti berikut:

Tabel 1. Solusi Fisibel Basis Awal dengan aturan NWC

	D1	D2	D3	D4	
S1	6 **60**	8 **20**	4	5	80
S2	2	9 **15**	3 **35**	8	50
S3	5	4	9 **50**	7 **40**	90
	60	35	85	40	**1490**

Terdapat sebuah korespondensi antara setiap tabel transportasi dengan *tree*. Setiap *arc* antara *node* sumber dan *node* tujuan menunjukkan variabel basis. *Arc* yang terputus (tidak dialokasikan) menunjukkan variabel nonbasis.

Adanya hubungan ini, maka setiap *network* persoalan transportasi memiliki sifat-sifat sebagai berikut:

1. $m+n$ *node*.
2. $m+n-1$ *arc*.
3. Setiap *node* dihubungkan pada *node* lain oleh satu atau lebih *arc*.
4. Tidak ada *loop*. Hanya terdapat satu rangkaian *arc-arc* antara setiap pasangan *node-node*.

Sifat-sifat ini mengarah pada *spanning tree* dalam persoalan transportasi dengan

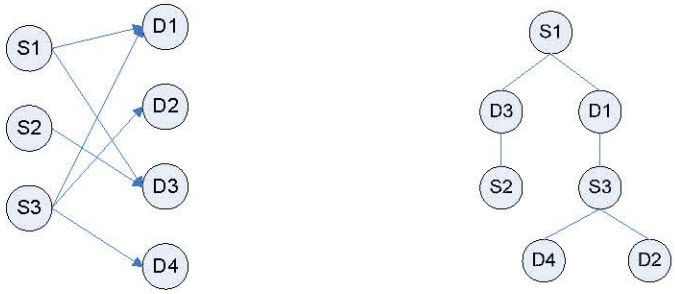
$m+n$ *node* dan $m \times n$ *arc*. Kenyataan diatas memberikan petunjuk bahwa setiap basis dapat diartikan sebagai *spanning tree*. Sehingga konsep ide ini disebut dengan *Basis Tree*.

Menambah satu *arc* ke *basis tree* akan sama dengan variabel nonbasis yang baru masuk (*entering variabel*). Menambahkan satu *arc* pada *basis tree* membentuk unik *cycle* dan mempunyai hubungan dengan unik $\theta loop$ pada solusi persoalan transportasi. *Loop* ini akan terputus dengan penghapusan *arc* yang lain dalam *loop*. Proses pengahapusan *arc* pada satu *loop* mempunyai korespondensi dengan *leaving variable* dan menjadi variabel nonbasis.

Struktur setiap *basis spanning tree* menjadi lebih jelas jika mengikuti langkah-langkah sebagai berikut:

1. Pilih *node* pertama (S_1 dalam contoh ini) sebagai *root* dalam *basis tree*.
2. Gambar semua *node* dan *arc* yang lain dibawah *root*.

Dua langkah diatas digambarkan sebagai berikut:



Bipartite Graph Persoalan Transportasi Basis Tree

Gambar 2 .Bipartite Graph Persoalan Transportasi dan Basis Tree

Gambar 2 haruslah menjadi jelas bahwa semua informasi tentang solusi fisibel basis memuat dalam *basis tree*. Hanya informasi lain dibutuhkan untuk memindahkan dari satu basis ke basis berikutnya. Informasi tersebut adalah daftar *arc-arc* nonbasis.

2.2.2 Pivoting Menggunakan Basis Tree

Suatu variabel nonbasis dipilih untuk tahap *entering variable*, proses transformasi basis sekarang menjadi basis yang baru disebut *pivoting*. Akan ditunjukkan bagaimana *pivot* menggunakan *basis tree*. Dalam proses ini semua operasi aljabar digantikan oleh operasi graph. Ini tidak hanya menjelaskan semua operasi, tetapi juga semua langkah komputasi menjadi sederhana.

Langkah-langkah pivoting pada basis tree, diberikan nonbasis *arc* (i^*, j^*) dengan penurunan ongkos negatif telah dipilih dari daftar *arc*. Langkah selanjutnya dibentuk:

1. Tambahkan *arc* baru antara *node* i^* dan *node* j^*, sehingga akan membentuk unik $\theta loop$ yang ditemukan sebagai berikut:
   a. Dimulai dari *node* i^* naik keatas *tree* yaitu *root*. Ini akan memberi lintasan yang unik $i^* \rightarrow ... \rightarrow root$.
   b. Dimulai dari *node* j^* naik keatas *tree* yaitu *root*. Ini akan memberi lintasan yang unik $j^* \rightarrow ... \rightarrow root$.
   c. Dua lintasan ini akan bertemu pada *node* NCA (*Nearest Common Ancestor*) dari i^* dan j^*. Ini akan memberi lintasan yang unik $\theta loop$ $i^* \rightarrow ... \rightarrow NCA \rightarrow ... \rightarrow j^* \rightarrow i^*$.
2. Dengan menggunakan unik $\theta loop$, carilah *arc* yang akan dipindahkan dari *loop*. *Arc* disini bertanda (-) dengan alokasi arus (komoditas) yang terkecil.
3. Potong *arc* yang ditemukan pada langkah (2) dan bangun kembali *tree*.
4. Mengatur nilai komoditas pada *arc-arc* dalam unik $\theta loop$. Menghitung dual variabel u_i, v_j untuk setiap *node* pada *tree* dengan $u_1 = 0$.

Dapat dilihat bahwa operasi yang berbelit-belit dalam *pivoting* dapat terselesaikan dengan *basis tree*. Tidak ada

bagian lain dalam *network* (*basis tree*) yang direkayasa. Terjadi reduksi dalam hal komputasi dan permintaan penyimpanan dalam memori.

Walaupun demikian mengoperasikan secara aljabar senuah matrik ukuran $(m+n)xmn$ dapat kita dioperasikan dengan basis tree ukuran $m+n$. Ini merupakan kunci algoritma *network transportasi basis tree.*

3. METODE PENELITIAN

Penelitian ini menggunakan pendekatan rasional. Penelitian ini merupakan studi literature, dilakukan dengan mempelajari berbagai karya ilmiah dalam bentuk jurnal maupun buku teks, atau artikel-artikel lainnya yang berhubungan dengan bahan penelitian. Diawali dengan mempelajari *graph*, *tree*, *spanning tree*, dan persoalan transportasi. Kemudian berlandaskan materi-materi tersebut, akan dicari solusi untuk persoalan transportasi dengan menggunakan basis tree.

4. HASIL DAN PEMBAHASAN

Pembahasan berikut adalah metode basis tree akan digunakan untuk mencari solusi nilai minimum pada contoh persoalan transportasi yang diberikan. Berikut adalah contoh persoalan diberikan tiga perusahaan (S_i) dan empat gudang (D_j) dengan data komoditas *supply* dan permintaan komoditas sebagai berikut:

$$S_1 = 80, S_2 = 50, S_3 = 90$$

$$D_1 = 60, D_2 = 35, D_3 = 85, D_4 = 40$$

Data ongkos transportasi pada tabel berikut:

Tabel 4. 1 Ongkos Transportasi 3 x 4

	D1	D2	D3	D4	
S1	6	8	4	5	80
S2	2	9	3	8	50
S3	5	4	9	7	90
	60	35	85	40	**220**

Iterasi Pertama

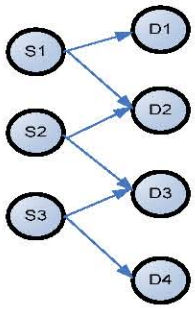
Langkah Pertama:

Dengan menggunakan *North-West Corner* diperoleh tabel berikut.

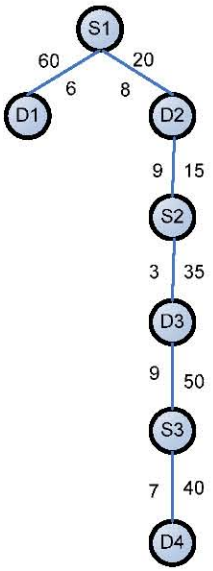
Tabel 4. 2 Jumlah Komoditas Hasil NWC

	D1	D2	D3	D4	
S1	6 60	8 20	4	5	80
S2	2	9 15	3 35	8	50
S3	5	4	9 50	7 40	90
	60	35	85	40	**1490**

Dari tabel 4.2 buatlah solusi fisibel basis awal dengan S_1 sebagai *root*.



Gambar 4. 1 Bipartite Graph Persoalan Transportasi



Gambar 4. 2 Basis Tree

Total Ongkos = (60.6)+(20.8)+(15.9)+(35.3)+(50.9)+(40.7)= 1490

Langkah Kedua:

Menghitung dual variabel u_i, v_j pada variabel basis:

$$u_1 + v_1 = 6$$
$$u_1 + v_2 = 8$$
$$u_2 + v_2 = 9$$
$$u_2 + v_3 = 3$$
$$u_3 + v_3 = 9$$
$$u_3 + v_4 = 7$$

Jika $u_1 = 0$, maka secara simultan diperoleh:

$$u_2 = 1, u_3 = 7, v_1 = 6, v_2 = 8, v_3 = 2, v_4 = 0$$

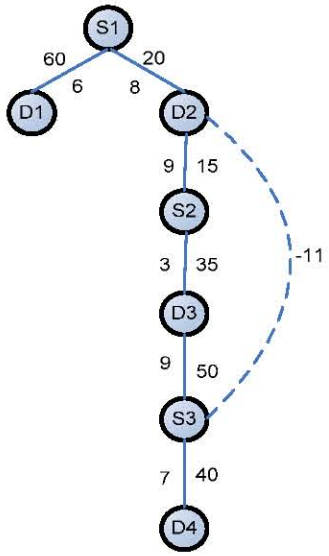
Penurunan ongkos transportasi pada variabel nonbasis diperoleh:

$$\overline{c}_{ij} = c_{ij} - (u_i + v_j)$$
$$\overline{c}_{13} = 4 - (0 + 2) = 2$$
$$\overline{c}_{14} = 5 - (0 + 0) = 5$$
$$\overline{c}_{21} = 2 - (1 + 6) = -5$$
$$\overline{c}_{24} = 8 - (1 + 0) = 7$$
$$\overline{c}_{31} = 5 - (7 + 6) = -8$$
$$\overline{c}_{32} = 4 - (7 + 8) = -11$$

Langkah ketiga:

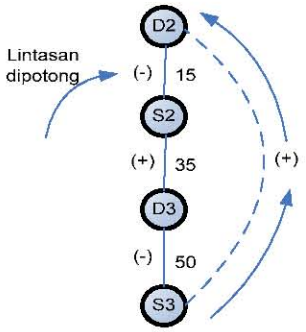
Memilih variabel nonbasis, yaitu penurunan ongkos transportasi paling negative. Variabel tersebut adalah $x_{i^* j^*} = x_{32} = -11$

Pemilihan variabel nonbasis ini untuk menentukan entering variabel dan unik $\theta - loop$ sebagai berikut:



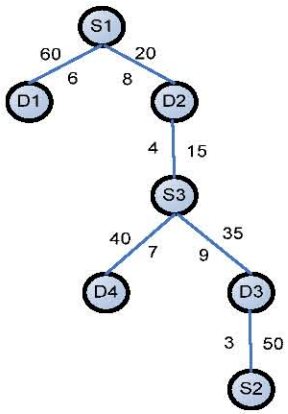
Gambar 4. 3 Unik Loop

Langkah keempat: Pada langkah ini gunakan pivoting untuk menentukan leaving variabel.



Gambar 4. 4 Proses Pivoting

Hasil pivoting diperoleh, maka selanjutnya adalah membangun kembali basis tree dan menghitung total ongkos transportasi.



Gambar 4. 5 Bangun Kembali Basis Tree

Total Ongkos = (60.6)+(20.8)+(15.4)+(40.7)+(35.9)+(50.3)= 1325

Iterasi akan berlanjut sampai tidak ditemukan lagi variabel nonbasis pada penurunan ongkos transportasi bernilai negaif.

Hasil optimum ditemukan sampai iterasi kelima dengan total ongkos minimum 920.

5. KESIMPULAN

Persoalan program linear dari algoritma simpleks untuk solusi persoalan transportasi ditemukan oleh G.B Dantzig pada tahun 1947. Konsep metode simpleks dapat diaplikasikan pada persoalan transportasi. Metode basis tree untuk aplikasi persoalan transportasi adalah salah satu alternatif untuk solusi persoalan transportasi yang lebih efisien.

Algoritma transportasi basis tree memiliki tiga prosedur utama:

1. Mencari solusi fisibel basis awal
2. Memilih arc nonbasis
3. Proses pivoting

DAFTAR PUSTAKA

Aho, A.V., Hopcroft, J.E., dan Ulman, J.D., 1987, *Data Structures And Algorithms*, Canada: Addison-Wesley.

Ary, Maxsi, 2005, *Meminimumkan Biaya Transportasi Komponen Elektrik Pesawat Telepon Jenis PTE-991 Di PT.INTI Menggunakan Metode Basis Tree*, Tugas Akhir, tidak diterbitkan, Bandung: Jurusan Matematika Unisba.

Bak, S., Greer, A., dan Mitra, S., (2010). *Hybrid Cyberphysical System Verification With Simplex Using Discrete Abstraction*. Retrieved 6 2, 2010, from http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5465972

Bieling, J., Peschlow, P., dan Martini, P., (2010). *An Efficient GPU Implementation of the Revised Simplex Method*. Retrieved 6 2, 2010, from http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5470831

Deo, Narsing, 1989, *Graph Theory With Applications To Engineering And Computer Science*, New Delhi: Prentice-Hall.

Dimyati, Tjutju, Tarliah, dan Dimyati, Ahmad, 1992, *Operation Research: Model-Model Pengambilan Keputusan*, Bandung: Sinar Baru.

Li, Lingyun, Huang, Z., Da, Qingli, dan Hu, Jinsong, (2008). *A New Method Based on Goal Progamming for Solving Transportation Problem with Fuzzy Cost*. Retrieved 6 2, 2010, from http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4554047

O'Connor, Derek, R., 2001, *Algorithms And Data Structures*, Retrieved 1 2, 2010, (http://www.derekroconnor.net/home/MMS406/Trees.pdf).

Wilson, J., Robin, dan John, J., Watkins, 1990, *Graph An Introductory Approach*, Canada: John Wiley & Sons Inc.