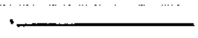
BAB II

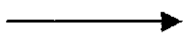
MATERI PENUNJANG

2.1. GRAPH

Beberapa definisi dan teorema tentang graph yang digunakan untuk menentukan sebuah jaringan dengan permasalahan rute terdekat yang penyelesaiannya menggunakan model transshipment, dapat dilihat pada definisi-definisi di bawah ini. Sebelumnya diberikan beberapa pengertian mengenai gambar-gambar berikut ini :

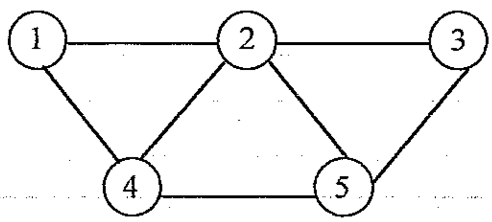
: lingkaran bernomor menggambarkan node


: garis menggambarkan edge


: garis berarah menggambarkan arc

Definisi 2.1.

Suatu graph, yang dinotasikan dengan $G\ (N,E)$, adalah himpunan berhingga titik (node) $N(G) = \{n_1, n_2, n_3, \ldots, n_n\}$ yang tidak kosong dan himpunan busur atau garis (edge), yang mungkin kosong, $E(G) = \{ n_1n_2, n_1n_3, \ldots, n_mn_n\}$.



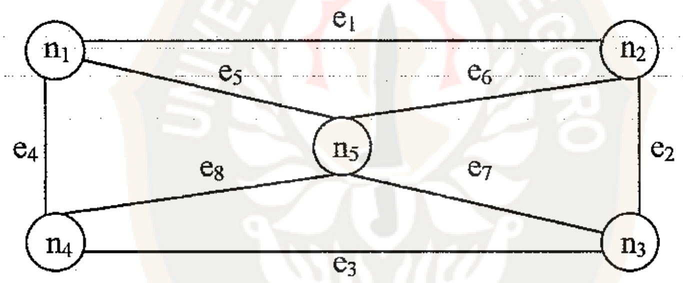
Gambar 2.1. Graph dengan 5 buah node dan 7 buah edge

Definisi 2.2.

Setiap garis pada sebuah graph menghubungkan dua buah titik yang disebut titik ujung.

Definisi 2.3.

Walk adalah deretan bergantian antara titik dan garis yang dimulai dan diakhiri dengan titik, $W : n_0, e_1, n_1, e_2, \ldots, n_{n-1}, e_n, n_n$ (n ≥ 0), dimana $e_i = n_{i-1}n_i$. Trail adalah walk dimana tidak ada garis yang diulang, sedangkan path adalah walk dengan tidak ada titik yang diulang.



Gambar 2.2. Graph lengkap dengan nama node dan edge.

Dari gambar 2.2. di atas, yang disebut dengan walk adalah W : $n_1, e_4, n_4, e_8, n_5, e_7, n_3, e_2, n_2, e_6, n_5, e_8, n_4$. Sedangkan trail adalah $n_1, e_5, n_5, e_8, n_4, e_3, n_3, e_7, n_5, e_6, n_2$ dan path adalah $n_1, e_4, n_4, e_3, n_3, e_2, n_2$.

Teorema 2.1.

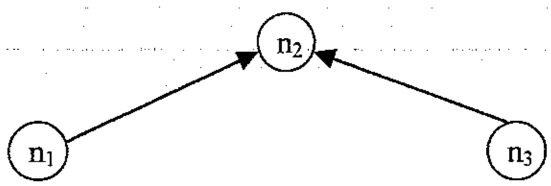
Setiap walk $n - m$ dalam sebuah graph terdapat sebuah path $n - m$.

Bukti

Misal W adalah walk $n - m$ pada sebuah graph G. Jika $n = m$, maka diperoleh jawaban yaitu path $n - n$. Sekarang kita asumsikan bahwa $n \neq m$. Andaikan $W : n = n_0, n_1, n_2, \ldots, n_n = m$. Jika tidak ada node dari G terdapat dalam W lebih dari satu kali, maka W itu sendiri adalah path $n - m$. Jika terdapat beberapa node dalam G terdapat dalam W lebih dari satu kali. Misal i dan j adalah bilangan integer positif yang berbeda dengan $i < j$. Jika hubungan $n_i, n_{i+1}, \ldots, n_{j-1}$ dihilangkan dari W, sebuah walk $n - m$, sebut W_1 yang mempunyai panjang kurang dari W. Jika tidak ada node pada G terdapat dalam W_1 lebih dari satu kali, maka W_1 adalah path $n - m$. Jika belum ditemukan kasus seperti ini, maka kita lakukan prosedur di atas, sampai akhirnya kita sampai pada path $n - m$.

Definisi 2.4.

Suatu graph berarah (direct graph atau digraph) adalah suatu graph yang terdiri dari suatu himpunan titik-titik (node) N dan suatu himpunan garis (arc) A, yang terdiri dari pasangan terurut dari titik-titik yang ditulis sebagai (x,y) yang disebut garis berarah, yang dinotasikan dengan A = $\{(n_1,n_2), (n_1,n_3), \ldots, (n_m,n_n)\}$.



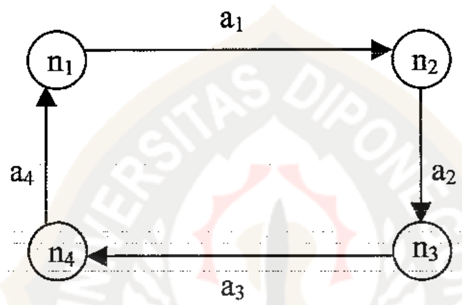
Gambar 2.3. Graph berarah

Definisi 2.5.

Setiap arc pada sebuah graph mempunyai dua buah titik, yaitu titik awal (initial endpoint) dan titik akhir (terminal endpoint).

Definisi 2.6.

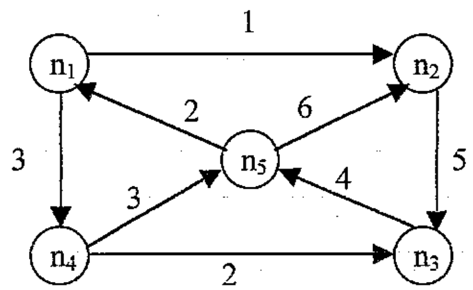
Cycle adalah walk $n_0, n_1, n_2, \ldots, n_n$, dengan n ≥ 3, dimana node awal sama dengan node akhir, $n_0 = n_n$ dan dengan n titik, $n_1, n_2, n_3, \ldots, n_n$ yang berbeda.



Gambar 2.4. Cycle, dengan urutan n_1, a_1, n_2, a_2, n_3, a_3, n_4, a_4, n_1

Definisi 2.7.

Nilai atau bobot dari suatu garis dalam graph berarah ditulis dengan lambang $l(a)$ atau $l(n_i, n_j)$.



Gambar 2.5. Graph berarah dengan bobot pada tiap barisnya.

Dengan menambahkan sebuah node sumber dan node tujuan, maka gambar di atas merupakan sebuah jaringan. Jadi dapat diketahui bahwa jaringan adalah suatu digraph yang di dalamnya terdapat sebuah node sumber dan node tujuan, serta sebuah nilai nonnegatif yang merupakan nilai masing-masing arc pada digraph tersebut.

Semua pengertian mengenai walk, trail dan path pada definisi 2.3. juga berlaku untuk graph berarah. Jika melihat gambar 2.5., maka yang disebut dengan walk adalah $n_1, n_2, n_3, n_5, n_1, n_4, n_3, n_5, n_2$. Sedangkan trail adalah $n_1, n_2, n_3, n_5, n_1, n_4, n_3$ dan path adalah n_1, n_4, n_3, n_5, n_2.

Definisi 2.8.

Nilai suatu path adalah jumlah nilai semua garis yang membentuk path tersebut. Nilai path dari titik i ke j ditulis dengan lambang $l(P_{ij})$.

Contoh : Dari gambar 2.4. diperoleh $l(P\ n_1n_4) = l(a_1) + l(a_2) + l(a_3) + l(a_4)$

Definisi 2.9.

Jarak dari titik n_i ke titik n_j, yang dinotasikan dengan $d(n_i,n_j)$, adalah nilai terkecil dari semua path $n_i - n_j$ dalam suatu digraph.

2.2. PROGRAM LINEAR

Metode Program Linear mengalokasikan sumber daya yang terbatas untuk mencapai suatu tujuan seperti memaksimalkan keuntungan atau meminimumkan biaya. Model program linear terdiri dari sebuah fungsi tujuan linear dan beberapa kendalanya.

Tujuan dari model ini adalah untuk mencapai hasil terbaik yang mungkin diperoleh dengan keterbatasan sumber daya. Hasil yang dicapai ditunjukkan sebagai maksimisasi dari beberapa ukuran, seperti profit dan penjualan, atau sebagai minimisasi, seperti biaya dan waktu.

Setelah masalah diidentifikasikan dan tujuan ditetapkan, langkah selanjutnya adalah memformulasikan model matematika yang meliputi tiga tahap sebagai berikut :

1. Tentukan variabel keputusan dan nyatakan dalam simbol matematika.
2. Membentuk fungsi tujuan yang ditunjukkan sebagai suatu hubungan linear dari variabel keputusan.
3. Menentukan semua kendala masalah dan mengekspresikan dalam persamaan atau pertidaksamaan, dan merupakan hubungan linear dengan variabel keputusan yang mencerminkan keterbatasan sumber daya masalah tersebut.

Bentuk model program linear adalah :

maksimumkan/minimumkan $z = c_1x_1 + c_2x_2 + \ldots + c_nx_n$

dengan pembatas $z = \sum_{semuabusur} c_{ij}x_{ij}$

$\forall x_j \geq 0$

Keterangan dari model tersebut adalah :

- z adalah nilai fungsi tujuan.

- x_j adalah banyaknya kegiatan j, $j = 1, 2, \ldots, n$. Hal ini berarti terdapat n variabel keputusan.
- c_j adalah sumbangan per unit kegiatan j.
- b_i adalah jumlah sumber daya ke i, $i = 1, 2, \ldots, m$. Hal ini berarti terdapat m jenis sumber daya.
- a_{ij} adalah banyaknya sumber daya i yang digunakan oleh tiap unit dari kegiatan j.

Program linear dapat diselesaikan dengan dua metode yaitu metode grafis dan metode simpleks. Metode grafis digunakan apabila persoalan program linear yang akan diselesaikan hanya mempunyai dua buah variabel. Metode simpleks digunakan untuk memecahkan persoalan program linear yang mempunyai jumlah variabel dan pambatas yang besar (≥ 2). Metode simpleks menyelesaikan masalah program linear melalui perhitungan ulang (iterasi) dimana langkah-langkah perhitungan yang sama diulang berkali-kali hingga solusi optimum dicapai. Dalam menggunakan metode simpleks, model program linear harus diubah ke dalam suatu bentuk baku atau *standard form*. Ciri-ciri dari bentuk baku model program linear adalah :

1. Semua kendala berupa persamaan dengan sisi kanan non negatif.

   Suatu kendala jenis $\leq$ dan $\geq$ dapat diubah menjadi suatu persamaan dengan menambahkan atau mengurangkan suatu variabel slack pada sisi kiri kendala.
2. Semua variabel harus merupakan variabel non negatif.
3. Fungsi tujuan dapat berupa maksimisasi atau minimisasi.

2.3. MASALAH RUTE TERDEKAT

Masalah rute terdekat berkaitan dengan penentuan garis-garis berarah yang dihubungkan dalam sebuah jaringan yang membentuk jarak terdekat antara sumber dan tujuan. Setiap arc pada jaringan tersebut, misal arc (n_i,n_j), memiliki jarak yang dinotasikan dengan $d(n_i,n_j)$ atau d_{ij}, yang merupakan jarak antara node i dengan node j. Teorema mengenai jarak dalam sebuah graph dapat dilihat pada teorema berikut ini.

Teorema 2.2.

Dimisalkan G adalah sebuah graph, maka

i. $d_{ij} \geq 0$, dan $d_{ij} = 0$ jika dan hanya jika $i = j$.

ii. $d_{ij} = d_{ji}$ untuk semua $i, j \in N(G)$

iii. $d_{ij} \leq d_{ik} + d_{kj}$ untuk semua $i, j, k \in N(G)$

Bukti

i. Jika d_{ij} bernilai ∞ atau sama dengan jumlah nilai semua garis yang membentuk path $i - j$ terdekat, maka $d_{ij} \geq 0$. Jika $d_{ij} = 0$, maka path $i - j$ terdekat tidak mempunyai garis, dengan demikian $i = j$. Jika $i = j$, maka tidak dapat membentuk sebuah garis sehingga $d_{ij} = 0$.

ii. Untuk membuktikan bahwa $d_{ij} = d_{ji}$, kita menganggap dua buah kasus, yaitu jika $d_{ij} = \infty$ dan jika $d_{ij} \neq \infty$. Pada kasus pertama sudah dapat dipastikan bahwa $d_{ji} = \infty$. Pada kasus kedua, anggap path $i - j$ terpendek, serta path $j - i$ diperoleh dari path $i - j$ terpendek dengan membalik urutan node. Karena path $j - i$ hanya merupakan pembalikan urutan node dari path $i - j$, maka hal ini berarti $d_{ij} = d_{ji}$.

iii. Misal i, j, dan k adalah node dari G. Misal P adalah path $i - k$ terpendek dan Q adalah path $k - j$ terpendek pada G. Maka P diikuti dengan Q adalah walk $i - j$, sebut W, dengan panjang $d_{ik} + d_{kj}$. Dari pengertian bahwa setiap W terdiri dari path $i - j$, maka hal ini dapat diikuti dengan pengertian $d_{ij} \leq d_{ik} + d_{kj}$.

Tujuan dari permasalah rute terdekat adalah untuk mencari rute yang memiliki jarak terdekat dari sebuah sumber ke beberapa tujuan. Namun sebagian besar aplikasi rute terdekat melibatkan kriteria biaya sebagai pengganti jarak. Sehingga dalam hal ini permasalahan rute terdekat tidak bertujuan untuk mencari jarak yang terdekat melainkan untuk mendapatkan biaya minimum. Rute-rute tersebut berpengaruh terhadap biaya pengiriman produk yang melewatinya, tetapi biaya pengiriman produk tersebut tidak dipengaruhi oleh jumlah produk yang dikirimkan.

Dalam sebuah jaringan, tidak semua node saling berhubungan. Ada kalanya sebuah node dengan node yang lain tidak dihubungkan dengan sebuah arc. Arc yang menghubungkan antara node awal i dengan node akhir j dinotasikan dengan x_{ij}. Keadaan tersebut mempengaruhi koefisien dari hubungan antara node-node tersebut, dimana koefisien pada variabel x_{ij} bernilai 1, jika node i dan node j dihubungkan oleh sebuah arc dan bernilai 0, jika node i dan node j tidak dihubungkan oleh sebuah arc.

Jika digambarkan ke dalam bentuk program linear dan dengan menganggap bahwa node awal (sumber) adalah i dan node akhir (tujuan) adalah n, maka permasalahan rute terdekat mempunyai fungsi tujuan yaitu :

Minimumkan $z = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$

Dengan kendala pada tiap-tiap node adalah :

$\sum_{keluar} x_{ij} - \sum_{masuk} x_{ij} = 1$ untuk node sumber

$\sum_{keluar} x_{ij} - \sum_{masuk} x_{ij} = 0$ untuk node antara

$\sum_{keluar} x_{ij} - \sum_{masuk} x_{ij} = -1$ untuk node tujuan

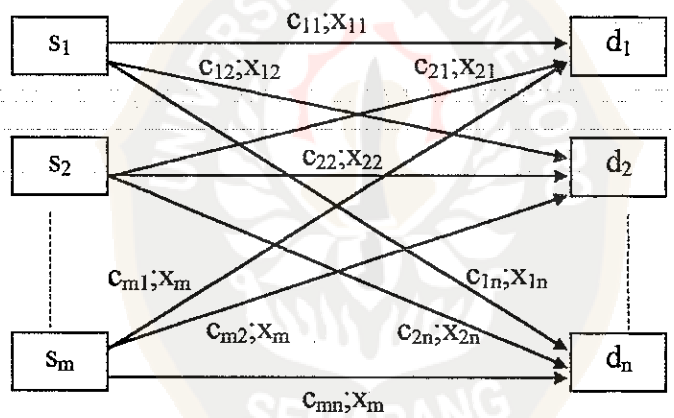
Nilai 1 dan –1 di atas merupakan jumlah rute terdekat yang keluar dari node sumber dan yang masuk ke node tujuan.

Dalam perhitungan fungsi tujuan, variabel x_{ij} bernilai 1 untuk arc yang dipakai sebagai rute dan bernilai 0 untuk arc yang tidak dipakai sebagai rute, sedangkan c_{ij} merupakan ongkos yang harus dikeluarkan apabila melewati rute tersebut. Terdapat beberapa penyelesaian masalah rute terdekat, salah satu penyelesaian yang digunakan adalah dengan menggunakan model transshipment.

2.4. MODEL TRANSPORTASI

Transportasi merupakan suatu model jaringan di mana di dalamnya terdapat suatu sistem garis-garis berarah yang menghubungkan titik-titik yang berlainan, yang berhubungan dengan pendistribusian suatu produk dari sumber ke tujuan. Model transportasi bertujuan untuk meminimalkan ongkos pengangkutan dari sejumlah sumber ke sejumlah tujuan.

Bentuk umum dari model transportasi ini dapat dilihat pada gambar distribusi suatu produk dari *m* sumber ke *n* tujuan.



Gambar 2.6. Bentuk umum Model Transportasi

Gambar distribusi produk di atas menerangkan tentang :

1. Masing-masing sumber mempunyai kapasitas s_i, $i = 1, 2, ..., m$
2. Masing-masing tujuan membutuhkan produk sebanyak d_j, $j = 1, 2, ..., n$
3. Jumlah satuan (unit) yang dikirim dari sumber i ke tujuan j sebanyak x_{ij}
4. Ongkos pengiriman per unit dari sumber i ke tujuan j adalah c_{ij}

Model transportasi dapat dirumuskan ke dalam model program linear sebagai berikut :

Minimumkan : $z = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$

berdasarkan pembatas : $\sum_{j=1}^{n} x_{ij} = s_i \quad , i = 1, 2, \ldots, m$

$$\sum_{i=1}^{m} x_{ij} = d_j \quad , j = 1, 2, \ldots, n$$

$x_{ij} \geq 0$ untuk semua i dan j

Kemudian dari formulasi program linear di atas dapat dibuat suatu tabel program linear yang secara umum dapat digambarkan sebagai berikut :

z	x_{11}	x_{12}	…	x_{1n}	x_{21}	…	x_{2n}	x_{m1}	x_{m2}	…	x_{mn}	solusi
1	$-c_{11}$	$-c_{12}$	…	$-c_{1n}$	$-c_{21}$	…	$-c_{2n}$	$-c_{m1}$	$-c_{m2}$	…	$-c_{mn}$	0
0	1	1	…	1	0	…	0	0	0	…	0	s_1
0	0	0	…	0	1	…	1	0	0	…	0	s_2
…	…	…	…	…	…	…	…	…	…	…	…	…
0	0	0	…	0	0	…	0	1	1	…	1	s_m
0	1	0	…	0	1	…	0	1	0	…	0	d_1
0	0	1	…	0	0	…	0	0	1	…	0	d_2
…	…	…	…	…	…	…	…	…	…	…	…	…
0	0	0	…	1	0	…	1	0	0	…	1	d_n

Tabel 2.1. Tabel program linear secara umum

Nilai *m* dan *n* dalam fungsi tujuan maupun dalam batasan supply atau demand tidak selalu sama, ada kalanya ditemukan suatu permasalahan dengan nilai *m* > *n* atau *m* < *n*. Untuk nilai *m* dan *n* yang kecil, masalah transportasi pada tabel program linear di atas dapat diselesaikan dengan menggunakan metode simplex, tetapi tidak jarang ditemukan permasalahan dengan nilai *m* dan *n* yang cukup besar, sehingga untuk menyelesaikannya membutuhkan perhitungan yang panjang, waktu yang lama dan tempat yang tidak sedikit. Oleh karena itu diperlukan suatu solusi khusus yang perhitungannya lebih efisien dibandingkan dengan metode simplex.

Dengan adanya solusi khusus tersebut, maka tabel program linear di atas tidak akan digunakan lagi. Dan dengan sifatnya yang khas, yaitu setiap koefisien pada pembatas adalah 1, maka dapat dibuat suatu tabel khusus yang dinamakan tabel transportasi. Bentuk umum dari tabel transportasi adalah sebagai berikut :

	Ke	Tujuan				
	Dari	1	2	...	n	Supply
S u	1	x_{11} c_{11}	x_{12} c_{12}	...	x_{1n} c_{1n}	s_1
m b	2	x_{21} c_{21}	x_{22} c_{22}	...	x_{2n} c_{2n}	s_2
e	...	...	...	...	...	...
r	m	x_{m1} c_{m1}	x_{m2} c_{m2}	...	x_{mn} c_{mn}	s_m
	Demand	d_1	d_2	...	d_n	

Tabel 2.2. Tabel transportasi secara umum

Keterangan dari tabel transportasi di atas adalah sumber ditulis dalam baris dan tujuan ditulis dalam kolom. Penawaran dari setiap sumber ditulis pada kolom paling kanan, sedangkan permintaan dari masing-masing tujuan ditulis pada baris paling bawah. Biaya pengangkutan per unit (c_{ij}) ditulis pada kotak kecil di bagian kanan atas dari setiap kotak. Variabel x_{ij} pada setiap kotak menyatakan jumlah barang yang didistribusikan dari sumber *i* ke tujuan *j*, variabel inilah yang nantinya akan dicari nilainya.

Model transportasi ini memiliki keseimbangan pada saat total supply sama dengan total demand, namun dalam persoalan yang sebenarnya sering terjadi jumlah supply yang tersedia lebih besar atau lebih kecil daripada jumlah yang diminta. Namun setiap persoalan transportasi dapat dibuat seimbang dengan memasukkan variabel semu (dummy). Jika jumlah demand melebihi jumlah supply, maka dibuat suatu sumber dummy yang akan mensupply kekurangan yang ada. Sebaliknya, jika jumlah supply melebihi jumlah demand, maka dibuat suatu tujuan dummy untuk menyerap kelebihan tersebut. Penambahan dummy ini dilakukan untuk menyeimbangkan jumlah penawaran dengan permintaan. Ongkos pengiriman per unit pada variabel dummy ini sama dengan nol, sehingga tidak mempengaruhi metode yang digunakan dalam penyelesaian model transportasi.

Untuk menyelesaikan masalah transportasi terdapat langkah-langkah yang harus dilakukan, yaitu :

1. Menentukan solusi awal.
2. Menentukan solusi optimum.

Untuk menentukan solusi awal, dapat dilakukan dengan beberapa metode, diantaranya Metode Pojok Kiri Atas-Pojok Kanan Bawah (North-West Corner Method), Metode Ongkos Terkecil (Least Cost Method), dan Metode Pendekatan Vogel (Vogel's Approximation Method / VAM). Sedangkan untuk menentukan solusi optimum digunakan Metode Batu

Loncatan (Stepping Stone Method) dan Metode MODI (Modified Distribution). Ketiga metode untuk menentukan solusi awal tersebut mempunyai cara yang berbeda-beda dan solusi yang nilainya berbeda-beda pula.

Untuk menyelesaikan permasalahan kali ini digunakan Metode Ongkos Terkecil untuk menentukan solusi awal. Dan apabila persyaratan penggunaan metode batu loncatan terpenuhi, maka untuk menghitung solusi optimum digunakan Metode Batu Loncatan.

Adapun langkah-langkah dalam mengoperasikan Metode Ongkos Terkecil adalah sebagai berikut :

1. Susunlah semua supply, demand dan ongkos pendistribusian (c_{ij}) ke dalam tabel transportasi.
2. Alokasikan supply atau demand sebanyak mungkin ke kotak dengan nilai c_{ij} minimum, jika terdapat dua atau lebih c_{ij} minimum, pilih salah satu, sesuaikan supply dengan demand. Lakukan kembali langkah ini sampai semua supply atau demand terpenuhi. Sehingga dicapai solusi yang optimal.

Setelah mendapatkan solusi awal, kemudian dilakukan pengecekan terhadap hasil tersebut dengan menggunakan Metode Batu Loncatan untuk mengetahui apakah hasil dari perhitungan dengan menggunakan metode ongkos terkecil sudah optimal atau belum.

Metode batu loncatan digunakan untuk menentukan entering variable, yaitu variabel non basis atau variabel yang di dalamnya tidak terdapat alokasi supply atau demand yang menjadi tujuan perpindahan alokasi supply atau demand dari leaving variable yang merupakan variabel basis atau variabel yang di dalamnya terdapat alokasi supply atau demand sebagai asal perpindahan alokasi supply atau demand. Sebelum menentukan entering variabel dan leaving variabel, terlebih dahulu dibuat suatu loop tertutup bagi setiap variabel non basis yang berawal dan berakhir pada variabel non basis yang sama, dimana tiap sudut atau titik dari loop tersebut ditempati oleh variabel basis dalam tabel transportasi tersebut. Loop tersebut digunakan untuk memeriksa perubahan ongkos transportasi c'_{ij} pada variabel non basis, jika terdapat perpindahan alokasi supply atau demand dari variabel basis ke variabel non basis tersebut. Nilai c'_{ij} terdapat pada variabel non basis yang digunakan untuk menentukan entering variabel. Formula dari c'_{ij} adalah :

$$c'_{ij} = c_{ij} - c_{ij_1} + c_{i_1 j_1} - c_{i_1 j_2} + \ldots + c_{i_n j_n} - c_{i_n j}$$

Dengan ketentuan bahwa $i \neq i_1 \neq \ldots \neq i_n$ dan $j \neq j_1 \neq \ldots \neq j_n$, serta c_{ij} adalah sudut loop yang merupakan ongkos pada variabel non basis sedangkan $c_{ij_1}, c_{i_1 j_1}, \ldots, c_{i_n j}$ adalah sudut loop yang merupakan ongkos dari variabel basis. Jumlah sudut loop pada formula tersebut adalah $2n + 2$, dengan $n + 1$ supply dan $n + 1$ demand yang berbeda. Setelah semua nilai c'_{ij} diketahui, dapat ditentukan entering variabel yaitu variabel non basis x_{ij} dengan nilai c'_{ij} negatif yang paling minimal yang akan mengakibatkan pengurangan ongkos transportasi terbanyak. Setelah itu beri tanda positif pada variabel dengan nilai c positif dan tanda negatif pada variabel dengan nilai c negatif, sesuai dengan formula di atas. Tanda positif dan negatif pada variabel-variabel tadi menyatakan bahwa pada variabel tersebut terdapat penambahan atau pengurangan alokasi dengan ketentuan jumlah alokasi pada masing-masing baris atau kolom harus tetap. Leaving variabel dapat ditentukan dengan memilih variabel yang mempunyai tanda negatif dengan nilai alokasi yang terkecil. Pengurangan ongkos transportasi adalah sebanyak perkalian antara nilai c_{ij} dengan jumlah alokasi yang dipindahkan ke variabel tersebut. Hal ini dapat dibuktikan pada perhitungan-perhitungan berikut ini.

Misalkan terdapat dua supply (i, j) dan dua demand (k, l) dengan fungsi tujuan minimumkan $z = c_{ik}x_{ik} + c_{il}x_{il} + c_{jk}x_{jk} + c_{jl}x_{jl}$.

Dengan batasan supply $c_{ik}x_{ik} + c_{il}x_{il} = s_i$

$c_{jk}x_{jk} + c_{jl}x_{jl} = s_j$

dan batasan demand $c_{ik}x_{ik} + c_{jk}x_{jk} = d_k$

$c_{il}x_{il} + c_{jl}x_{jl} = d_l$

Dengan menggunakan metode ongkos terkecil, dimisalkan bahwa tidak terdapat alokasi supply maupun demand pada variabel x_{il}, sehingga variabel x_{il} merupakan variabel non basis dan x_{ik}, x_{jk}, x_{jl} adalah variabel basis.

Setelah itu dicari nilai perubahan ongkos transportasi dari variabel non basis dengan rumus : $c'_{il} = c_{il} - c_{ik} + c_{jk} - c_{jl}$. Karena hanya ada satu nilai perubahan ongkos transportasi dari satu variabel non basis, maka dengan menganggap bernilai negatif, nilai c'_{il} merupakan nilai dimana terdapat entering variabel, yaitu x_{il}.

Setelah itu dicari leaving variabel yang merupakan nilai minimal dari jumlah alokasi yang dipindahkan pada variabel basis yang memiliki nilai c

negatif, yaitu variabel x_{ik} dan x_{jl}. Anggap nilai minimal adalah variabel x_{ik}, sehingga diperoleh nilai perubahan pada masing-masing variabel adalah :

$x'_{il} = x_{il} + x_{ik}$

$x'_{ik} = x_{ik} - x_{ik}$

$x'_{jk} = x_{jk} + x_{ik}$

$x'_{jl} = x_{jl} - x_{ik}$

Dari variabel tersebut apabila dimasukkan ke dalam fungsi tujuan adalah :

$z' = c_{ik}x'_{ik} + c_{il}x'_{il} + c_{jk}x'_{jk} + c_{jl}x'_{jl}$

$z' = c_{ik}(x_{ik} - x_{ik}) + c_{il}(x_{il} + x_{ik}) + c_{jk}(x_{jk} + x_{ik}) + c_{jl}(x_{jl} - x_{ik})$

$z' = c_{ik}x_{ik} - c_{ik}x_{ik} + c_{il}x_{il} + c_{il}x_{ik} + c_{jk}x_{jk} + c_{jk}x_{ik} + c_{jl}x_{jl} - c_{jl}x_{jk}$

$z' = c_{ik}x_{ik} + c_{il}x_{il} + c_{jk}x_{jk} + c_{jl}x_{jl} - c_{ik}x_{ik} + c_{il}x_{ik} + c_{jk}x_{ik} - c_{jl}x_{jk}$

$z' = c_{ik}x_{ik} + c_{il}x_{il} + c_{jk}x_{jk} + c_{jl}x_{jl} + x_{ik}(c_{il} - c_{ik} + c_{jk} - c_{jl})$

$z' = z + x_{ik}c'_{il}$

Karena c'_{il} bernilai negatif, maka persamaan di atas menjadi :

$z' = z - x_{ik}c'_{il}$

Dari rumus terakhir ini dapat diketahui bahwa dengan menggunakan metode batu loncatan, ongkos transportasi dapat berkurang sebanyak $x_{ik}c'_{il}$.

Berikut ini akan diberikan suatu contoh penggunakan metode ongkos terkecil dalam menentukan solusi awal dan metode batu loncatan untuk menentukan solusi optimum.

Misalkan terdapat suatu permasalahan transportasi sebagai berikut :

min $z = 6x_{11} + x_{12} + 10x_{13} + 3x_{21} + 12x_{22} + 4x_{23} + 11x_{31} + 8x_{32} + 5x_{33}$

dengan kendala sumber $x_{11} + x_{12} + x_{13} = 10$

$x_{21} + x_{22} + x_{23} = 15$

$x_{31} + x_{32} + x_{33} = 10$

dan kendala tujuan $x_{11} + x_{21} + x_{31} = 10$

$x_{12} + x_{22} + x_{32} = 5$

$x_{13} + x_{23} + x_{33} = 20$

Kemudian dibuat suatu tabel transportasi sebagai berikut :

	1	2	3	
1	6	1	10	10
2	3	12	4	15
3	11	8	5	10
	10	5	20	

Tabel 2.3. Contoh penggunaan tabel transportasi

Karena tabel transportasi tersebut telah memiliki keseimbangan, maka permasalahan transportasi dapat langsung diselesaikan dengan menggunakan metode ongkos terkecil, seperti pada tabel transportasi di bawah ini.

	1	2	3	
1	6	1 5	10 5	10
2	3 10	12	4 5	15
3	11	8	5 10	10
	10	5	20	

Tabel 2.4. Contoh penyelesaian tabel transportasi

Sehingga diperoleh ongkos transportasi sebesar :

z = 1 x 5 + 5 x 10 + 3 x 10 + 4 x 5 + 5 x 10 = 155

Ongkos transportasi tersebut merupakan solusi awal dari penyelesaian masalah transportasi. Setelah itu menentukan perubahan ongkos transportasi c'_{ij} pada semua variabel non basis.

$c'_{11} = c_{11} - c_{13} + c_{23} - c_{21} = 6 - 10 + 4 - 3 = -3$

$c'_{22} = c_{22} - c_{23} + c_{13} - c_{12} = 12 - 4 + 10 - 1 = 17$

$c'_{31} = c_{31} - c_{33} + c_{23} - c_{21} = 11 - 5 + 4 - 3 = 7$

$c'_{32} = c_{32} - c_{33} + c_{13} - c_{12} = 8 - 5 + 10 - 1 = 12$

Kemudian menentukan entering variabel, yaitu x_{11}. Beri tanda positif pada variabel x_{11} dan x_{23} serta tanda negatif pada variabel x_{13} dan x_{21}, sehingga dapat ditentukan leaving variabel, yaitu x_{13} karena mempunyai jumlah alokasi yang lebih kecil daripada alokasi pada variabel x_{21}, seperti terlihat pada tabel transportasi berikut ini.

	1	2	3	
1	+ 6 -3	1 5	- 10 5	10
2	- 3 10	12 17	+ 4 5	15
3	11 7	8 12	5 10	10
	10	5	20	

Tabel 2.5. Tabel transportasi dengan perubahan ongkos transportasi pada masing-masing variabel non basis.

Setelah itu lakukan perpindahan alokasi dari leaving variabel ke entering variabel dengan memperhatikan jumlah alokasi pada masing-masing baris dan kolom harus tetap. Pada variabel yang bertanda positif terdapat penambahan jumlah alokasi, begitu juga sebaliknya. Sehingga tabel transportasinya menjadi seperti di bawah ini.

	1	2	3	
1	6 5	1 5	10	10
2	3 5	12	4 10	15
3	11	8	5 10	10
	10	5	20	

Tabel 2.6. Tabel transportasi setelah dilakukan metode batu loncatan

Lakukan lagi pencarian nilai perubahan ongkos transportasi pada variabel non basis di atas.

$c'_{13} = c_{13} - c_{11} + c_{21} - c_{23} = 10 - 6 + 3 - 4 = 3$

$c'_{22} = c_{22} - c_{21} + c_{11} - c_{12} = 12 - 3 + 6 - 1 = 14$

$c'_{31} = c_{31} - c_{33} + c_{23} - c_{21} = 11 - 5 + 4 - 3 = 7$

$c'_{32} = c_{32} - c_{33} + c_{23} - c_{21} + c_{11} - c_{12} = 8 - 5 + 4 - 3 + 6 - 1 = 9$

Karena sudah tidak ditemukan c'_{ij} yang bernilai negatif, maka penyelesaian di atas adalah penyelesaian optimal dari permasalahan transportasi. Jadi dengan penyelesaian menggunakan metode ongkos terkecil dan metode batu loncatan, didapatkan ongkos transportasi sebesar :

$z = 6 \times 5 + 1 \times 5 + 3 \times 5 + 4 \times 10 + 5 \times 10 = 140.$

Ongkos transportasi ini berkurang 15 jika dibandingkan dengan solusi awal, dimana jumlah 15 ini merupakan perkalian antara perubahan ongkos transportasi sebesar -3 dengan jumlah perpindahan alokasi yang besarnya 5.

Salah satu prosedur alternatif dari penggunaan model transportasi ini adalah model transshipment. Model transshipment ini merupakan model transportasi yang di dalamnya terdapat node transshipment, dimana pengiriman unit dari sumber melewati node ini sebelum sampai ke tujuan. Dengan kata lain untuk mengubah menjadi model transshipment, maka pada model transportasi perlu ditambahkan suatu node transshipment, dimana pada node ini terdapat arc masuk dan arc keluar. Fungsi tujuan dari model transshipment sama dengan model transportasi, yaitu :

Minimumkan $z = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$ atau $z = \sum_{semuabusur} c_{ij}x_{ij}$

Pada model transshipment, selain kendala node sumber dan node tujuan yang sama dengan model transportasi, perlu ditambahkan pula kendala pada node transshipment sebagai berikut :

$$\sum_{j=1}^{n} x_{ij} = \sum_{i=1}^{m} x_{ij} \quad \text{atau} \quad \sum_{keluar} x_{ij} = \sum_{masuk} x_{ij}$$

Sehingga fungsi tujuan untuk model transshipment adalah :

Minimumkan $z = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$ atau $z = \sum_{semuabusur} c_{ij}x_{ij}$

Dengan kendala untuk tiap node adalah :

Node sumber i $\quad \sum_{j=1}^{n} x_{ij} = s_i$ atau $\sum_{keluar} x_{ij} = s_i$

Node tujuan j $\quad \sum_{i=1}^{m} x_{ij} = d_j$ atau $\sum_{masuk} x_{ij} = d_j$

Node transshipment $\quad \sum_{j=1}^{n} x_{ij} = \sum_{i=1}^{m} x_{ij}$ atau $\sum_{keluar} x_{ij} = \sum_{masuk} x_{ij}$

dengan $x_{ij} \geq 0$ untuk semua nilai $i = 1, 2, \ldots, m$ dan $j = 1, 2, \ldots, n$.