KOMPUTASI MASALAH ANGKUTAN DENGAN BANTUAN KOMPUTER

SKRIPSI

Diajukan untuk memenuhi salah satu syarat
memperoleh gelar Sarjana Pendidikan
Program Studi Pendidikan Matematika

Disusun Oleh :

INDAH DWI UTAMI

NIM : 91414004
NIRM : 910052010501120004

PROGRAM STUDI PENDIDIKAN MATEMATIKA
JURUSAN PENDIDIKAN MATEMATIKA DAN ILMU PENGETAHUAN ALAM
FAKULTAS KEGURUAN DAN ILMU PENDIDIKAN
UNIVERSITAS SANATA DHARMA
YOGYAKARTA

1997

SKRIPSI

KOMPUTASI MASALAH ANGKUTAN DENGAN BANTUAN KOMPUTER

Disusun Oleh :

INDAH DWI UTAMI

NIM : 91414004
NIRM : 910052010501120004

Telah disetujui oleh :

Pembimbing I

Drs. B. Susanta

Tanggal, ... 1997

Pembimbing II

Drs. J. Eka Priyatma, M.Sc.

Tanggal, ... 1997

SKRIPSI

KOMPUTASI MASALAH ANGKUTAN DENGAN BANTUAN KOMPUTER

Yang dipersiapkan dan disusun oleh :

INDAH DWI UTAMI

NIM : 91414004

NIRM : 910052010501120004

Telah dipertahankan di depan Dewan Penguji

Pada Tanggal 24 November 1997

dan dinyatakan telah memenuhi syarat

SUSUNAN DEWAN PENGUJI

Nama Lengkap		Tanda Tangan
Ketua	: Drs. Fr. Y. Kartika Budi, M.Pd.	....
Sekretaris	: Dr. St. Suwarsono	....
Anggota	: Drs. B. Susanta	....
	Drs. J. Eka Priyatma, M.Sc.	....
	Dr. Y. Marpaung	....

Yogyakarta, ... 1997

Fakultas Keguruan dan Ilmu Pendidikan

Universitas Sanata Dharma

Dekan

Dr. Paulus Suparno, S.J., M.S.T.

KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Esa atas limpahan rahmat dan karunia-Nya sehingga skripsi yang berjudul Komputasi Masalah Angkutan dengan Bantuan Komputer ini dapat terselesaikan.

Penyusunan skripsi ini dimaksudkan untuk memenuhi salah satu syarat memperoleh gelar sarjana Pendidikan Program Studi Pendidikan Matematika di jurusan PMIPA fakultas Keguruan dan Ilmu Pendidikan Universitas Sanata Dharma Yogyakarta.

Pada kesempatan kali ini, penyusun ingin mengucapkan terima kasih kepada :

- Drs. B. Susanta selaku dosen pembimbing I yang dengan teliti dan penuh kesabaran membimbing dan memberi masukan yang sangat berharga dalam proses penyusunan skripsi ini.
- Drs. J. Eka Priyatma, M.Sc. selaku dosen pembimbing II yang selalu memberikan waktu dan penuh kesabaran membimbing dan memberi masukan yang sangat berharga sehingga pembuatan program komputer masalah angkutan dapat diselesaikan.
- Drs. Fr. Y. Kartika Budi, M.Pd. selaku ketua jurusan PMIPA.
- Drs. St. Susento, M.Si. selaku ketua Program Studi Pendidikan Matematika.

- Bapak dan Ibu dosen yang telah membimbing dan mendidik penyusun selama belajar di Universitas Sanata Dharma Yogyakarta.
- Bapak dan Ibu karyawan yang telah banyak membantu penyusun selama kuliah dan menyusun skripsi ini.
- Ibu dan Saudara-Saudara penyusun yang telah memberikan dorongan spiritual maupun material.
- Rekan-rekan mahasiswa Pendidikan Matematika 91 yang telah memberi dukungan, semangat dan doa.
- Teman-teman yang telah memberi dukungan, bantuan dan doa.

Penyusun menyadari bahwa masih terdapat kekurangan dalam skripsi ini, oleh karena itu segala masukan dan saran yang membangun akan diterima dengan senang hati.

Harapan penyusun semoga skripsi ini dapat berguna bagi para pembaca dan kemajuan Ilmu Pengetahuan.

Penyusun

DAFTAR ISI

ABSTRAK

Masalah angkutan adalah kejadian khusus dari program linear. Oleh karena itu masalah angkutan dapat diselesaikan dengan menggunakan metode simpleks maupun matriks, tetapi karena kekhususannya maka diciptakan metode angkutan yang perhitungannya lebih singkat dan mudah dibandingkan metode simpleks maupun matriks. Yang menjadi persoalan dalam masalah angkutan adalah mengalokasikan barang dari beberapa sumber ke beberapa tujuan sedemikian hingga semua kapasitas sumber terpakai dan semua kebutuhan tujuan terpenuhi. Sasaran utama masalah angkutan yaitu meminimumkan ongkos angkut total.

Ada tiga kejadian dalam masalah angkutan yaitu masalah angkutan setimbang, masalah angkutan tak setimbang yang layak dan masalah angkutan tak setimbang yang tak layak. Karena algoritma angkutan yang dibuat hanya untuk masalah angkutan setimbang maka untuk menyelesaikan masalah angkutan tak setimbang akan dikembalikan terlebih dahulu ke masalah angkutan setimbang.

Masalah yang dapat dikembalikan ke masalah angkutan antara lain adalah masalah angkutan pola maksimum dan masalah pemindahan muatan. Sedangkan masalah penugasan dan masalah penjaluran kapal tangki diselesaikan dengan metode tersendiri yang pengerjaannya lebih mudah dan cepat dibanding dengan metode angkutan. Masalah penugasan merupakan kejadian khusus masalah angkutan, setiap sumber hanya dapat mengirimkan barang ke tepat satu tujuan. Masalah pemindahan muatan dan masalah penjaluran tangki merupakan masalah angkutan yang diperluas. Dalam masalah pemindahan muatan, setiap sumber dapat berfungsi sebagai tujuan dan setiap tujuan dapat berfungsi sebagai sumber.

Untuk mempermudah kita dalam menyelesaikan masalah angkutan yang berformat besar telah dibuat program komputer yang dapat membantu kita dalam melakukan komputasi.

ABSTRACT

Transportation problem is a special case of linear programming. So this problem can be solved using simplex or matrix method, but due to its special character a simpler method it adopted, the transportation method. The problem in this model is how to allocate a certain goods from several origin to several destinations subject to the maximum capacity of each origin and the minimum demand of each destination. The objective is to minimize the total transportation cost.

There are three cases in the transportation problem, they are the balanced transportation problem, feasible unbalanced transportation problem and unfeasible unbalanced transportation problem. Since this transportation algorithm is available only for balanced transportation problem so the unbalanced transportation problem should be converted to balanced transportation problem.

Problem that can be converted to minimum pattern of transportation problem are the maximum pattern of transportation problem and the transshipment problem. The assigment problem and tanker routing problem are not converted to transportation problem, it will be solved by using a certain method that can be done easier and quicker than the transportation method, Assignment problem is a special case of the transportation problem, each origin can only send goods exactly to one destination. Transshipment problem and tanker routing problem are generalizations of the transportation problem. In the transshipment problem each origin can be considered as destination and each destination can be considered as origin.

For a large format transportation problem, computer program can help us do the computation aspects.

BAB I

PENDAHULUAN

A. ALASAN PEMILIHAN JUDUL

Masalah angkutan yang merupakan kejadian khusus dari program linear adalah salah satu bentuk matematika terapan. Masalah angkutan ini menarik perhatian dan minat penulis karena banyak masalah dalam kehidupan sehari-hari yang dapat diselesaikan dengan menggunakan metode angkutan. Komputer merupakan alat yang dapat dengan cepat membantu kita dalam menyelesaikan masalah. Komputasi dengan bantuan komputer ini juga menarik minat dan perhatian penulis karena masyarakat telah mengenal komputer untuk membantu menyelesaikan masalah. Di sini penulis akan memperdalam materi masalah angkutan dan pembuatan program komputer yang pernah didapat dalam kuliah.

B. METODE PENULISAN

Skripsi ini ditulis dengan menggunakan metode studi kepustakaan dan eksperimen pembuatan program komputer.

C. LATAR BELAKANG

Program linear merupakan bagian dari matematika terapan yang terutama menunjang usaha pemilihan atau pengambilan keputusan. Pengambilan keputusan ini mencakup bidang-bidang yang sangat luas, seperti politik, ekonomi, sosial, industri, manajemen, pertanian dan teknik sejauh masalah yang dihadapi bersifat kuantitatif. Misalnya dalam bidang ekonomi dan industri dijumpai masalah-masalah meminimumkan ongkos rata-rata, memaksimumkan laba, memaksimumkan efisiensi dan sebagainya. Contoh masalah di atas menyangkut optimisasi.

Dalam praktek banyak dijumpai masalah optimisasi berkendalakan pertidaksamaan linear dengan fungsi sasaran yang linear pula. Program linear merupakan model optimisasi masalah-masalah tersebut. Program linear merupakan model yang paling sederhana karena hanya melibatkan kendala yang linear dan fungsi sasaran yang linear pula.

Dalam masalah program linear selalu ada tiga komponen yaitu :

1. Fungsi sasaran yang akan dioptimumkan linear. Bentuknya adalah :

$$f(x) = \sum_{j=1}^{n} c_j \; x_j$$

   x_j = variabel kendala

   c_j = variabel ongkos

2. Kendala utama berbentuk linear. Bentuknya adalah :

$$\sum_{j=1}^{n} a_{ij} \; x_j \; \{ \leq \; = \; \geq \} \; b_i$$

   a_{ij} = koefisien variabel

   b_i = konstanta yang menyatakan maksimum atau minimum sumber

   Tanda $\{ \leq \; = \; \geq \}$ berarti diambil salah satu relasi $\leq$,

= atau ≥.

3. Kendala tak negatif, yang berbentuk :

$$x_j \geq 0 \ , \ j = 1,2,\ldots,n$$

Masalah angkutan merupakan salah satu model khusus dari program linear yang banyak dijumpai dalam kehidupan nyata. Oleh karena itu masalah angkutan juga mempunyai tiga komponen dasar yang terdapat dalam masalah program linear. Kekhususan dari masalah angkutan ini adalah :

1. a_{ij} (koefisien pada variabel kendala) terbatas bernilai 0 atau 1
2. Barang yang diangkut sejenis
3. Semua kendala utama berbentuk persamaan

Untuk menyelesaikan masalah angkutan berformat besar dengan cara manual akan memakan waktu yang tidak sedikit, oleh karena itu diciptakan program komputer yang dapat membantu kita dalam melakukan komputasi. Contoh program komputer yang biasa digunakan untuk menyelesaikan masalah program linear adalah LINDO dan STORM.

D. RUANG LINGKUP

Secara garis besar skripsi ini terbagi atas tiga bagian, yaitu pembukaan, isi dan penutup. Ketiga bagian tersebut secara keseluruhan diuraikan dalam tujuh bab. Pendahuluan ditulis dalam BAB I. Isi skripsi ini dimuat dalam BAB II sampai BAB VI, yang secara ringkas dapat diuraikan sebagai berikut : BAB II membicarakan Pola Baku Masalah Angkutan yang meliputi gambaran masalah angkutan, bentuk simpleks masalah angkutan, bentuk matriks masalah angkutan dan kejadian masalah angkutan. Langkah - langkah penyelesaian masalah angkutan dibahas dalam BAB III yang meliputi penentuan penyelesaian awal dengan berbagai metode, uji optimum dengan metode Stepping Stone dan metode MODI. Teori tentang aturan-aturan yang dipakai dalam masalah angkutan ditulis dalam BAB IV yang meliputi penyelesaian layak basis, sifat unimodular matriks A, lintasan tertutup metode stepping stone serta bilangan baris dan bilangan kolom MODI. Bab selanjutnya yaitu BAB V membahas masalah angkutan yang diperluas yang terdiri dari masalah penugasan, masalah pemindahan muatan dan masalah penjaluran tangki. Komputasi masalah angkutan dengan bantuan komputer ditulis dalam BAB VI. Dalam bab ini dijelaskan tentang langkah - langkah penyusunan program, input yang diperlukan program, perencanaan program, proses yang dilakukan program dan output yang dihasilkan program. Bab yang terakhir yaitu BAB VII adalah bab penutup yang berisi kesimpulan, saran dan daftar pustaka.

E. MATERI PRASYARAT

Untuk memahami skripsi ini diperlukan pengetahuan tentang program linear dan cara menjalankan TURBO BASIC.

BAB II
POLA BAKU MASALAH ANGKUTAN

A. GAMBARAN MASALAH

Suatu masalah angkutan dapat digambarkan sebagai berikut :

Ada suatu jenis barang tertentu, misalnya minyak, beras, semen dari beberapa tempat asal, misalnya depot, gudang, pabrik akan diangkut ke beberapa tempat tujuan, misalnya pasar, toko, lokasi proyek. Ongkos angkut tiap satuan barang dari tempat asal ke tempat tujuan diketahui. Suatu tempat tujuan dapat menerima barang dari satu atau lebih tempat asal. Begitu juga satu tempat asal dapat mengirim barang ke satu atau lebih tempat tujuan. Penawaran maksimum tempat asal dan kebutuhan minimum tempat tujuan harus diperhatikan. Sasaran dari masalah angkutan ini adalah menentukan banyaknya satuan barang yang harus dikirim dari masing-masing tempat asal ke masing-masing tempat tujuan sehingga ongkos angkut total menjadi minimum.

Sebuah masalah angkutan melibatkan m tempat asal / sumber $O_1, O_2, \ldots, O_m$ di mana pada masing - masing sumber berturut-turut menyediakan maksimum a_i $(i = 1,2,\ldots,m)$ satuan sejenis barang dan n tujuan $D_1, D_2, \ldots, D_n$ yang masing-masing berturut-turut membutuhkan minimum b_j $(j = 1,2,\ldots,n)$ satuan barang tersebut. Bilangan a_i dan b_j adalah bulat positif. Lambang c_{ij} menyatakan ongkos untuk mengangkut satu satuan produk dari sumber i ke tujuan j. Ongkos ini diketahui untuk tiap-tiap i dan j. Masalah yang dihadapi adalah bagaimana mengatur pola angkutan agar ongkos angkut total menjadi minimum dengan memenuhi semua kendala.

Secara umum masalah angkutan dapat dirumuskan secara matematis, yaitu :

mencari $x_{ij} \geq 0$, dengan $i = 1,2,\ldots,m$

$j = 1,2,\ldots,n$

yang memenuhi kendala

$$\sum_{j=1}^{n} x_{ij} \leq a_i \quad \text{dengan } i = 1,2,\ldots,m$$

$$\sum_{i=1}^{m} x_{ij} \geq b_j \quad \text{dengan } j = 1,2,\ldots,n$$

dan meminimumkan / memaksimumkan fungsi sasaran :

$$f = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}\, x_{ij}$$

Dari perumusan di atas terlihat bahwa masalah angkutan juga mempunyai tiga komponen dasar yang terdapat dalam masalah program linear, semua kendala maupun fungsi sasaran bersifat linear, sehingga masalah angkutan merupakan kejadian khusus dari masalah program linear.

Fungsi sasaran masalah angkutan adalah fungsi yang dioptimumkan disertai dengan seperangkat kendala pembatas. Penyelesaian yang memenuhi semua kendala yang ada disebut penyelesaian **fisibel / layak**, dan penyelesaian fisibel yang mengoptimumkan fungsi sasaran disebut **penyelesaian optimum**.

Dalam bentuk tabel, masalah angkutan dapat dinyatakan sebagai berikut :

Tabel 2.1
Tabel Masalah Angkutan

	D_1	D_2	...	D_j	...	D_n	a_i
O_1	c_{11} x_{11}	c_{12} x_{12}	...	c_{1j} x_{1j}	...	c_{1n} x_{1n}	a_1
O_2	c_{21} x_{21}	c_{22} x_{22}	...	c_{2j} x_{2j}	...	c_{2n} x_{2n}	a_2
$\vdots$	$\vdots$						$\vdots$
O_i	c_{i1} x_{i1}	c_{i2} x_{i2}	...	c_{ij} x_{ij}	...	c_{in} x_{in}	a_i
$\vdots$	$\vdots$						$\vdots$
O_m	c_{m1} x_{m1}	c_{m2} x_{m2}	...	c_{mj} x_{mj}	...	c_{mn} x_{mn}	a_m
b_j	b_1	b_2	...	b_j	...	b_n	Σa_i / Σb_j

Keterangan :

O_i : tempat asal ke i

D_j : tempat tujuan ke j

x_{ij} : jumlah barang yang harus diangkut dari tempat asal i ke tempat tujuan j.

c_{ij} : ongkos angkut tiap satuan barang dari tempat asal i ke tempat tujuan j.

a_i : banyak barang maksimum yang tersedia di tempat asal i

b_j : banyak permintaan minimum terhadap barang dari tempat tujuan j

m : banyak tempat asal / sumber

n : banyak tempat tujuan

Contoh masalah yang dapat digolongkan sebagai masalah angkutan adalah :

Contoh 2.1 :

Seorang pengusaha kue mempunyai 3 pabrik kue dan 4 toko kue. Pabrik A, B dan C setiap harinya dapat menghasilkan kue (dalam ribuan) berturut-turut sebanyak 60, 70 dan 100 ribu bungkus kue. Toko P, Q, R dan S membutuhkan kue (dalam ribuan) berturut-turut sebanyak 70, 50, 65 dan 45 ribu bungkus kue setiap harinya. Kue tersebut diantar dari masing-masing pabrik ke masing-masing toko dengan menggunakan mobil. Ongkos angkut kue dari masing-masing pabrik ke masing-masing toko (dalam ribuan rupiah) terdapat dalam tabel berikut :

	P	Q	R	S
A	40	45	30	25
B	30	25	50	20
C	55	40	35	60

Masalah yang dihadapi pengusaha tersebut adalah bagaimana menentukan jalur pengiriman kue dari pabrik ke toko agar ongkos angkut total menjadi minimum.

Contoh 2.2 :

Ada 3 pabrik semen akan mengirimkan semen dari pabrik ke 3 lokasi proyek. Semen yang diproduksi pabrik I, II, dan III berturut-turut adalah 40, 50, 30 (dalam ribuan sak) setiap harinya. Lokasi proyek I, II, dan III membutuhkan semen berturut-turut sebanyak 35, 25, dan 40 (dalam ribuan sak) setiap harinya. Semen-semen tersebut diangkut dari pabrik ke lokasi proyek dengan menggunakan truk. Ongkos angkut semen dari pabrik ke lokasi proyek (dalam ribuan rupiah) adalah sebagai berikut :

Lokasi / Pabrik	I	II	III
I	45	80	65
II	50	55	70
III	40	75	60

Bagaimana cara pengaturan pengangkutan semen tersebut agar ongkos angkut total menjadi minimum.

Contoh 2.3 :

Suatu perusahaan membutuhkan 3 orang supervisor, 8 orang operator, dan 4 orang distributor. Dalam suatu penerimaan pegawai baru terdapat 6 orang lulusan SMEA dan 9 orang lulusan SMA yang diterima. Setiap pegawai baru tersebut mempunyai hak yang sama untuk menduduki jabatan pekerjaan yang ditawarkan. Perusahaan telah menetapkan gaji bulanan yang berbeda untuk setiap lulusan pada masing-masing pekerjaan. Gaji bulanan (dalam ribuan rupiah) yang ditetapkan perusahaan adalah :

	Supervisor	Operator	Distributor
SMA	300	200	250
SMEA	275	150	225

Masalah yang dihadapi perusahaan adalah bagaimana menempatkan para pegawai baru ini pada masing-masing pekerjaan sehingga gaji bulanan total menjadi minimum.

Contoh 2.4 :

Suatu perusahaan kain sedang membuat motif suatu kain dengan 2 cara yaitu dicap dan dibatik. Perusahaan ini mempunyai 3 orang pekerja yaitu X, Y dan Z yang masing-masing mempunyai kemampuan yang berlainan dalam menggunakan kedua cara tersebut. Pekerja-pekerja tersebut dalam seminggu dapat bekerja selama 40 jam. Bahan untuk membuat cap dan membatik dapat digunakan selama 70 dan 50 jam setiap minggunya. Dengan kedua cara tersebut para pekerja setiap minggunya dapat membuat kain (dalam gulung) sebanyak :

	Dicap	Dibatik
X	10	24
Y	25	15
Z	20	20

Bagaimana cara perusahaan tersebut menempatkan masing - masing pekerja agar banyak kain total yang dihasilkan menjadi maksimum.

Contoh 2.1 dan 2.2 di atas sudah jelas merupakan masalah angkutan, sedangkan contoh 2.3 dan 2.4 tidak ada hubungannya dengan masalah angkutan dalam pengertian sehari-hari. Meskipun begitu contoh 2.3 dan 2.4 di atas dapat digolongkan sebagai masalah angkutan. Contoh 2.4 adalah contoh masalah yang memaksimumkan fungsi sasaran, sedang ketiga contoh yang lainnya adalah contoh masalah yang meminimumkan fungsi sasaran.

Dalam contoh 2.3 terdapat 2 sumber (lulusan SMA dan lulusan SMEA) dan 3 tujuan (Supervisor, Operator, Distributor). Banyaknya pegawai baru yang diterima kita anggap sebagai banyak penawaran dan banyaknya pekerja yang dibutuhkan untuk menduduki masing-masing jabatan kita anggap sebagai banyak permintaan. Gaji bulanan masing-masing lulusan pada masing-masing pekerjaan dianggap sebagai ongkos angkut dari tiap-tiap sumber ke tiap-tiap tujuan. Masalah yang dihadapi adalah bagaimana cara menempatkan masing-masing lulusan ke masing-masing pekerjaan agar gaji total pekerja menjadi minimum.

Dalam contoh 2.4 terdapat 3 sumber (Pekerja X, Y, Z) dan 2 tujuan (Dicap dan dibatik). Lamanya tiap pekerja bekerja setiap minggunya kita anggap sebagai banyak penawaran dan lamanya masing-masing cara dapat digunakan kita anggap sebagai banyak permintaan . Banyak kain yang dapat diselesaikan oleh masing-masing pekerja dalam menggunakan masing-masing cara kita anggap sebagai ongkos angkut dari tiap-tiap sumber ke tiap-tiap tujuan. Masalah yang dihadapi bagaimana menempatkan para pekerja ke masing - masing cara agar banyak kain total yang dikerjakan semua pekerja menjadi maksimum.

B. BENTUK SIMPLEKS MASALAH ANGKUTAN

Masalah angkutan merupakan masalah khusus program linear, oleh karena itu masalah angkutan dapat diselesaikan dengan menggunakan metode simpleks. Untuk membentuk tabel simpleks semua kendala harus dijadikan persamaan terlebih dahulu. Kendala yang berbentuk pertidaksamaan itu dapat diubah menjadi persamaan dengan cara :

1. Pada kendala penawaran $\sum_{j=1}^{n} x_{ij} \leq a_i$, dalam ruas kiri disisipkan S_i sedemikian hingga dipenuhi :

$$\sum_{j=1}^{n} x_{ij} + S_i = a_i \text{ dengan } S_i \geq 0$$

2. Pada kendala permintaan $\sum_{i=1}^{m} x_{ij} \geq b_j$, dalam ruas kanan disisipkan A_j sedemikian hingga dipenuhi :

$$\sum_{i=1}^{m} x_{ij} = b_j + A_j \text{ atau } \sum_{i=1}^{m} x_{ij} - A_j = b_j \text{ dengan } A_j \geq 0$$

   $-A_j$ bernilai negatif, supaya kendala permintaan menjadi basis pada kendala ini disisipkan B_j dengan $B_j \geq 0$ sedemikian hingga dipenuhi $\sum_{i=1}^{m} x_{ij} - A_j + B_j = b_j$.

S_i (i = 1,2,...,m) dan A_j (j = 1,2,...,n) disebut variabel pengetat (slack variabel), sedangkan B_j disebut variabel semu (artificial variabel). Variabel pengetat S_i menyatakan kapasitas semu dari sumber-sumber (tempat asal) O_i. Variabel pengetat A_j menyatakan kebutuhan semu dari tujuan-tujuan D_j. Koefisien ongkos untuk variabel pengetat adalah nol. Variabel semu B_j ditambahkan pada kendala permintaan dimaksudkan supaya kendala permintaan menjadi basis. Koefisien ongkos untuk variabel B_j adalah M, dengan M positif besar.

Fungsi sasaran menjadi :

$$f = \sum_i \sum_j c_{ij} \; x_{ij} + OS_i + OA_j + MB_j$$

Bentuk tabel simpleks awal untuk masalah angkutan dengan m sumber dan n tujuan adalah Tabel 2.2.

Keterangan Tabel 2.2 :

x_j = Variabel lengkap

b_i = Jumlah penawaran dari masing-masing sumber dan permintaan dari masing-masing tujuan.

c_j = Koefisien ongkos

x_i = Variabel yang menjadi basis.

c_i = Koefisien ongkos variabel basis.

$z_j = \sum_{i=1}^{m} c_i a_{ij}$

Apabila ada tabel simpleks awal untuk suatu masalah program linear berbentuk seperti tabel 2.2, maka masalah itu dapat digolongkan sebagai masalah angkutan.

Tabel 2.2
Tabel Simpleks Masalah Angkutan

	c_j	c_{11}	c_{12}	...	c_{1n}	c_{21}	c_{22}	...	c_{2n}	...	c_{m1}	c_{m2}	...	c_{mn}	0	0	...	0	0	0	..	0	M	M	...	M	
c_i	$x_i \backslash x_j$	x_{11}	x_{12}	...	x_{1n}	x_{21}	x_{22}	...	x_{2n}	...	x_{m1}	x_{m2}	...	x_{mn}	S_1	S_2	...	S_m	A_1	A_2	...	A_n	B_1	B_2	..	B_n	b_i R_i
0	S_1	1	1	...	1	0	0	...	0	...	0	0	...	0	1	0	...	0	0	0	...	0	0	0	..	0	a_1
0	S_2	0	0	...	0	1	1	...	1	...	0	0	...	0	0	1	...	0	0	0	...	0	0	0	..	0	a_2
⋮																											⋮
0	S_m	0	0	...	0	0	0	...	0	...	1	1	...	1	0	0	...	1	0	0	...	0	0	0	..	0	a_m
M	B_1	1	0	...	0	1	0	...	0	...	1	0	...	0	0	0	...	0	-1	0	...	0	1	0	..	0	b_1
M	B_2	0	1	...	0	0	1	...	0	...	0	1	...	0	0	0	...	0	0	-1	...	0	0	1	..	0	b_2
⋮																											⋮
M	B_n	0	0	...	1	0	0	...	1	...	0	0	...	1	0	0	...	0	0	0	..	-1	0	0	..	1	b_n
	Z_j																										Z

C. BENTUK MATRIKS MASALAH ANGKUTAN

Kendala yang berbentuk pertidaksamaan dapat diubah menjadi persamaan dengan menambah variabel pengetat. Dengan demikian setiap kendala utama selalu dapat diubah menjadi persamaan sehingga masalah angkutan dapat dirumuskan sebagai berikut :

mencari $x_{ij} \geq 0$ (2-5)

memenuhi kendala $\sum_{j=1}^{n} x_{ij} = a_i$ (2-6)

$\sum_{i=1}^{m} x_{ij} = b_j$ (2-7)

meminimumkan $f = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} \; x_{ij}$ (2-8)

Kendala masalah angkutan dengan m sumber dan n tujuan dapat diuraikan sebagai berikut :

$$
\begin{array}{llllll}
x_{11}+x_{12}+\ldots+x_{1n} & & & & = & a_1 \\
 & x_{21}+x_{22}+\ldots+x_{2n} & & & = & a_2 \\
 & & \vdots & & & \vdots \\
 & & & x_{m1}+x_{m2}+\ldots+x_{mn} & = & a_m \\
x_{11}+ & x_{21}+ & & x_{m1} & = & b_1 \\
x_{12}+ & x_{22}+ & & x_{m2} & = & b_2 \\
\vdots & & & & & \vdots \\
x_{1n}+ & x_{2n}+ & & x_{mn} & = & b_n
\end{array}
\qquad (2\text{-}9)
$$

Persamaan di atas dapat ditulis dalam bentuk matriks AX = B, yaitu :

$$
\begin{bmatrix}
11\ldots1 & 00\ldots0 & \ldots & 00\ldots0 \\
00\ldots0 & 11\ldots1 & \ldots & 00\ldots0 \\
\vdots & & & \vdots \\
00\ldots0 & 00\ldots0 & \ldots & 11\ldots1 \\
10\ldots0 & 10\ldots0 & \ldots & 10\ldots0 \\
01\ldots0 & 01\ldots0 & \ldots & 01\ldots0 \\
\vdots & & & \vdots \\
00\ldots1 & 00\ldots1 & \ldots & 00\ldots1
\end{bmatrix}
\begin{bmatrix}
x_{11} \\ x_{12} \\ \vdots \\ x_{1n} \\ x_{21} \\ x_{22} \\ \vdots \\ x_{mn}
\end{bmatrix}
=
\begin{bmatrix}
a_1 \\ a_2 \\ \vdots \\ a_m \\ b_1 \\ b_2 \\ \vdots \\ b_n
\end{bmatrix}
$$

Perhitungan masalah angkutan dengan menggunakan metode simpleks dan matriks akan sangat panjang dan sulit. Oleh karena itu masalah angkutan tidak akan dihitung dengan menggunakan simpleks maupun matriks. Karena masalah angkutan memiliki kekhususan tersendiri, yaitu a_{ij} (koefisien pada variabel dalam kendala) bernilai 1 atau 0, maka telah diciptakan algoritma khusus yang perhitungannya akan lebih singkat dibandingkan bila soal diselesaikan dengan metode simpleks maupun matriks.

D. KEJADIAN MASALAH ANGKUTAN

Ada 3 kejadian dalam masalah angkutan, yaitu

1. $\sum_i a_i = \sum_j b_j$, total penawaran sama dengan total permintaan. Kejadian seperti ini disebut kejadian setimbang.
2. $\sum_i a_i > \sum_j b_j$, total penawaran lebih besar dari total permintaan atau dengan kata lain ada kelebihan produksi. Kejadian seperti ini disebut kejadian tidak setimbang yang layak.
3. $\sum_i a_i < \sum_j b_j$, total penawaran lebih kecil dari total permintaan atau dengan kata lain kekurangan produksi. Kejadian seperti ini disebut kejadian tidak setimbang yang tak layak.

Untuk kejadian setimbang sudah ada suatu metode untuk menyelesaikan masalah angkutan yang disebut metode angkutan, sedangkan untuk kejadian tidak setimbang akan diusahakan dikembalikan ke kejadian setimbang. Dalam masalah angkutan setimbang semua kendala penawaran dan permintaan berbentuk persamaan.

Dari perumusan matematis masalah angkutan perhatikan kendala penawaran $\sum_{j=1}^{n} x_{ij} \le a_i$, $i = 1,2,\ldots,m$. Jika ini dijumlahkan menurut i diperoleh $\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} \le \sum_{i=1}^{m} a_i$.

Perhatikan kendala permintaan $\sum_{i=1}^{m} x_{ij} \ge b_j$, $j = 1,2,\ldots,n$.

Jika kendala permintaan ini dijumlahkan menurut j diperoleh $\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} \ge \sum_{j=1}^{n} b_j$.

Untuk kejadian setimbang $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j = P$ (*)

maka $P \le \sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} \le P$.

Ini berarti bahwa $\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} = P$.

Kendala penawaran kita susun menjadi :

$$\sum_j x_{1j} \le a_1$$

$$\sum_j x_{2j} \le a_2$$

$$\vdots$$

$$\sum_j x_{mj} \le a_m$$

$$P = \sum_i a_i \qquad \text{dari } (*)$$

Karena jumlah ruas kiri sama dengan jumlah ruas kanan maka semua pertidaksamaan di atas harus berbentuk persamaan, sebab bila ada satu saja kendala yang mempunyai relasi "<", misalnya $\sum_j x_{1j} < a_1$ maka harus ada paling sedikit satu dari kendala ke-2,...,ke-m yang akan mempunyai relasi ">" dan ini melanggar kendala penawaran.

Jadi jika $\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} = \sum_{i=1}^{m} a_i$, maka kendala penawaran harus berbentuk persamaan, karena tidak mungkin ada tanda ">"

dalam kendala penawaran.

Begitu pula jika $\sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} = \sum_{j=1}^{n} b_j$, maka kendala permintaan harus berbentuk persamaan pula, karena tidak mungkin ada tanda "<" dalam kendala permintaan.

Hanya masalah angkutan setimbang yang meminimumkan fungsi sasaran yang mempunyai algoritma. Algoritma inilah yang akan kita bahas. Masalah angkutan yang tidak setimbang harus disetimbangkan terlebih dahulu dan fungsi sasaran pola maksimum nanti akan diubah ke pola minimum.

BAB III
PENYELESAIAN MASALAH ANGKUTAN

A. PERSIAPAN PENYELESAIAN

Untuk menyelesaikan masalah angkutan (setimbang) dengan metode angkutan, lebih dahulu dibuat tabel kendala lengkap dengan ongkosnya.

Perumusan masalah angkutan setimbang :

mencari $x_{ij} \geq 0$ untuk $i = 1,2,\ldots,m$,

$$j = 1,2,\ldots,n \qquad (3\text{-}1)$$

dengan kendala :

$$\sum_{j=1}^{n} x_{ij} = a_i, \quad i = 1,2,\ldots m \quad \text{(kendala penawaran)} \qquad (3\text{-}2)$$

$$\sum_{i=1}^{m} x_{ij} = b_j, \quad j = 1,2,\ldots,n \quad \text{(kendala permintaan)} \qquad (3\text{-}3)$$

dengan $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$

meminimumkan fungsi sasaran :

$$f = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}\, x_{ij} \qquad (3\text{-}4)$$

Masalah angkutan setimbang mempunyai dua kendala yaitu kendala tak negatif dan kendala pembatas (utama). Kendala utama dibagi lagi menjadi dua yaitu kendala penawaran dan kendala permintaan. Semua kendala utama masalah angkutan setimbang berbentuk persamaan. Ada m buah kendala penawaran dan n buah kendala permintaan. Begitu juga ada m sumber dan n tujuan sehingga terdapat mn variabel dalam masalah angkutan.

Bentuk tabel kendala lengkap dengan ongkosnya masalah angkutan setimbang adalah sebagai berikut :

Tabel 3.1

Tabel Kendala lengkap dengan ongkosnya

	D_1	D_2	...	D_n	a_i
O_1	c_{11}	c_{12}	...	c_{1n}	a_1
O_2	c_{21}	c_{22}	...	c_{2n}	a_2
$\vdots$	$\vdots$				$\vdots$
O_m	c_{m1}	c_{m2}	...	c_{mn}	a_m
b_j	b_1	b_2	...	b_n	$\Sigma\ a_i = \Sigma\ b_j$

Dalam tabel terdapat m baris yang menunjukkan banyak penawaran dan n kolom yang menunjukkan banyak permintaan. Ongkos ditulis pada sudut kanan atas dengan huruf kecil, dan x_{ij} yaitu besar pengiriman dari tempat asal i ke tempat tujuan j ditulis di tengah dengan huruf normal. x_{ij} bernilai nol tidak ditulis dalam tabel. Jadi kotak-kotak dalam tabel hanya memuat x_{ij} yang bernilai positif. Kotak yang bernilai $x_{ij} > 0$ disebut kotak isi, yaitu kotak yang diperoleh sebagai jalur pengiriman. Sedangkan kotak yang bernilai $x_{ij} = 0$ disebut kotak kosong, yaitu kotak yang tidak digunakan sebagai jalur pengiriman. Untuk selanjutnya kotak (i,j) ditulis K_{ij}.

1. Penyelesaian Layak Basis

Kendala penawaran dan permintaan ditulis dalam bentuk panjang menjadi :

$$
\begin{array}{l}
x_{11}+x_{12}+\ldots+x_{1n} \qquad\qquad\qquad\qquad\qquad\qquad = a_1 \\
\qquad\qquad x_{21}+x_{22}+\ldots+x_{2n} \qquad\qquad\qquad = a_2 \\
\qquad\qquad\qquad\qquad \ddots \\
\qquad\qquad\qquad\qquad\qquad x_{m1}+x_{m2}+\ldots+x_{mn} = a_m \\
x_{11}+ \qquad x_{21}+ \qquad \ldots+ \; x_{m1} \qquad\qquad = b_1 \\
\quad x_{12}+ \qquad x_{21}+ \qquad \ldots \qquad + \; x_{m2} \qquad = b_2 \\
\ddots \\
\qquad x_{1n}+ \qquad x_{2n}+ \qquad \ldots \qquad +x_{mn} = b_n
\end{array}
$$

....(3-5)

Persamaan (3-5) adalah sistem m+n persamaan dalam mn variabel, tetapi persamaan ini tidak independen. Salah satu persamaan adalah berlebih, karena persamaan tersebut dapat diperoleh dari yang lain. Untuk contoh, jika kita jumlahkan m persamaan pertama dan dikurangi jumlahan n-1 persamaan berikutnya kita peroleh :

$$x_{1n}+x_{2n}+\ldots+x_{mn} \quad = \quad a_1+a_2+\ldots+a_m-b_1-b_2-\ldots-b_{n-1}$$

Padahal $x_{1n}+x_{2n}+\ldots+x_{mn} = b_n$ ....(3-6)

Berarti persamaan (3-6) berasal dari kombinasi persamaan yang lain. Oleh karena itu persamaan (3-6) dapat dihilangkan dari (3-5). Jadi suatu basis pada (3-5) tidak akan melibatkan m+n variabel tetapi hanya m+n-1.

Kita lihat bahwa persamaan di atas dependen linear dengan cara mencari ranknya dengan eliminasi Gauss.

$$
\begin{bmatrix}
11\ldots1 & 00\ldots0 & \ldots & 00\ldots0 \\
00\ldots0 & 11\ldots1 & \ldots & 00\ldots0 \\
\vdots & & & \vdots \\
00\ldots0 & 00\ldots0 & \ldots & 11\ldots1 \\
10\ldots0 & 10\ldots0 & \ldots & 10\ldots0 \\
01\ldots0 & 01\ldots0 & \ldots & 01\ldots0 \\
\vdots & & & \vdots \\
00\ldots1 & 00\ldots1 & \ldots & 00\ldots1
\end{bmatrix}
\rightarrow
\begin{bmatrix}
11\ldots1 & 00\ldots0 & \ldots & 00\ldots0 \\
00\ldots0 & 11\ldots1 & \ldots & 00\ldots0 \\
\vdots & & & \vdots \\
00\ldots0 & 00\ldots0 & \ldots & 11\ldots1 \\
0\text{-}1..\text{-}1 & 10\ldots0 & \ldots & 10\ldots0 \\
01\ldots0 & 01\ldots0 & \ldots & 01\ldots0 \\
\vdots & & & \vdots \\
00\ldots1 & 00\ldots1 & \ldots & 00\ldots1
\end{bmatrix}
$$

$$
\begin{bmatrix}
11\ldots1 & 00\ldots0 & \ldots & 00\ldots0 \\
01\ldots0 & 01\ldots1 & \ldots & 01\ldots0 \\
\vdots & & & \vdots \\
00\ldots1 & 00\ldots1 & \ldots & 00\ldots1 \\
00\ldots0 & 11\ldots1 & \ldots & 00\ldots0 \\
\vdots & & & \vdots \\
00\ldots0 & 00\ldots0 & \ldots & 11\ldots1 \\
0\text{-}1.\text{-}1 & 10\ldots0 & \ldots & 10\ldots0
\end{bmatrix}
\rightarrow
\begin{bmatrix}
11\ldots1 & 00\ldots0 & \ldots & 00\ldots0 \\
01\ldots0 & 01\ldots1 & \ldots & 01\ldots0 \\
\vdots & & & \vdots \\
00\ldots1 & 00\ldots1 & \ldots & 00\ldots1 \\
00\ldots0 & 11\ldots1 & \ldots & 00\ldots0 \\
\vdots & & & \vdots \\
00\ldots0 & 00\ldots0 & \ldots & 11\ldots1 \\
00..\text{-}1 & 11\ldots0 & \ldots & 11\ldots0
\end{bmatrix}
$$

$$
\begin{bmatrix}
11\ldots1 & 00\ldots0 & \ldots & 00\ldots0 \\
01\ldots0 & 01\ldots0 & \ldots & 01\ldots0 \\
\vdots & & & \vdots \\
00\ldots1 & 00\ldots1 & \ldots & 00\ldots1 \\
00\ldots0 & 11\ldots1 & \ldots & 00\ldots0 \\
\vdots & & & \vdots \\
00\ldots0 & 00\ldots0 & \ldots & 11\ldots1 \\
00\ldots0 & 11\ldots1 & \ldots & 11\ldots1
\end{bmatrix}
\rightarrow
\begin{bmatrix}
11\ldots1 & 00\ldots0 & \ldots & 00\ldots0 \\
01\ldots0 & 01\ldots0 & \ldots & 01\ldots0 \\
\vdots & & & \vdots \\
00\ldots1 & 00\ldots1 & \ldots & 00\ldots1 \\
00\ldots0 & 11\ldots1 & \ldots & 00\ldots0 \\
\vdots & & & \vdots \\
00\ldots0 & 00\ldots0 & \ldots & 11\ldots1 \\
00\ldots0 & 00\ldots0 & \ldots & \text{-}1\text{-}1.\text{-}1
\end{bmatrix}
$$

$$\begin{bmatrix} 11\ldots1 & 00\ldots0 & \ldots & 00\ldots0 \\ 01\ldots0 & 01\ldots0 & \ldots & 01\ldots0 \\ \vdots & & & \vdots \\ 00\ldots1 & 00\ldots1 & \ldots & 00\ldots1 \\ 00\ldots0 & 11\ldots1 & \ldots & 00\ldots0 \\ \vdots & & & \vdots \\ 00\ldots0 & 00\ldots0 & \ldots & 11\ldots1 \\ 00\ldots0 & 00\ldots0 & \ldots & 00\ldots0 \end{bmatrix}$$

Persamaan di atas mempunyai rank m+n-1, karena banyaknya baris yang tidak semuanya nol sebanyak m+n-1. Karena rank matriks koefisien di atas kurang dari m+n maka persamaan tersebut dependen linear. Jadi dapat disimpulkan bahwa suatu penyelesaian basis paling banyak memuat m+n-1 variabel yang tidak nol.

Jika dalam pengerjaan simpleks terjadi kemerosotan maka kita dapat bekerja terus sampai didapat penyelesaian yang optimum, tetapi dalam masalah angkutan kemerosotan sangat mengganggu.

Jadi dapat disimpulkan bahwa :

Banyak kotak isi dalam tabel harus tepat m+n-1

Andaikan $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j = s$ dan penyelesaian didefinisikan dengan :

$$x_{ij} = \frac{a_i \; b_j}{s} \tag{3-7}$$

untuk i = 1,2,...,m dan j = 1,2,...,n

Jelas $x_{ij} \geq 0$ jika $a_i \geq 0$ dan $b_j \geq 0$. Sekarang substitusikan pernyataan (3-7) ke dalam (3-2) dan (3-3) maka dihasilkan :

$$\sum_{j=1}^{n} \frac{a_i\ b_j}{s} = \frac{a_i}{s} \sum_{j=1}^{n} b_j = \frac{a_i}{s}\ s = a_i$$

$$\sum_{i=1}^{m} \frac{a_i\ b_j}{s} = \frac{b_j}{s} \sum_{i=1}^{m} a_i = \frac{b_j}{s}\ s = b_j$$

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

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

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

Jadi penyelesaian yang didefinisikan dengan (3-7) adalah layak untuk (3-2) dan (3-3) dan masalah angkutan akan selalu mempunyai penyelesaian layak jika $a_i \geq 0$ dan $b_j \geq 0$.

Penyelesaian layak yaitu penyelesaian yang memenuhi semua kendala. **Penyelesaian layak basis** yaitu penyelesaian yang memenuhi semua kendala dan sudah tersusut Gauss dengan mengenolkan variabel bebas. Masalah angkutan mempunyai variabel basis sebanyak m+n-1. Jadi penyelesaian yang mempunyai $x_{ij} > 0$ sebanyak m+n-1 disebut penyelesaian layak basis masalah angkutan. Untuk lebih jelasnya mari kita lihat tabel-tabel penyelesaian masalah angkutan pada Tabel 3.2. Masalah angkutan dengan m=3 dan n=3 maka m+n-1 = 5.

Pada Tabel 3.2 (a) penyelesaian layak tetapi bukan merupakan penyelesaian basis karena semua $x_{ij} > 0$ dan semua kendala penawaran dan permintaan terpenuhi, kotak isi ada 7 lebih dari m+n-1. Pada Tabel 3.2 (b) merupakan penyelesaian basis karena kotak isi ada 5 sama dengan m+n-1, tetapi bukan merupakan penyelesaian layak, karena

Tabel 3.2 (a)

Penyelesaian layak
Bukan basis

	D_1	D_2	D_3	a_i
O_1	(8) 2	(3)	(7) 6	8
O_2	(2) 2	(9) 6	(5) 2	10
O_3	(7)	(4) 4	(2) 2	6
b_j	4	10	10	24

Tabel 3.2 (b)

Penyelesaian basis
Tidak layak

	D_1	D_2	D_3	a_i
O_1	(8)	(3) 8	(7)	8
O_2	(2) 4	(9) -4	(5) 10	10
O_3	(7)	(4) 6	(2)	6
b_j	4	10	10	24

Tabel 3.2 (c)
Penyelesaian basis
Tidak layak

	D_1	D_2	D_3	a_i
O_1	(8) 2	(3)	(7) 4	8
O_2	(2)	(9) 5	(5)	10
O_3	(7)	(4) 3	(2) 3	6
b_j	4	10	10	24

Tabel 3.2 (d)
Penyelesaian layak
Basis

	D_1	D_2	D_3	a_i
O_1	(8)	(3) 8	(7)	8
O_2	(2) 4	(9) 2	(5) 4	10
O_3	(7)	(4)	(2) 6	6
b_j	4	10	10	24

Tabel 3.2 (e)
Penyelesaian layak basis
yang merosot

	D_1	D_2	D_3	a_i
O_1	(8) 4	(3)	(7) 4	8
O_2	(2)	(9) 10	(5)	10
O_3	(7)	(4)	(2) 6	6
b_j	4	10	10	24

$x_{22} = -4 < 0$. Tabel 3.2 (c) merupakan contoh penyelesaian basis tidak layak juga. Penyelesaian basis karena kotak isi ada 5 tetapi tidak layak karena tidak memenuhi semua kendala. Tabel 3.2 (d) merupakan contoh penyelesaian layak basis, karena semua kendala penawaran dan permintaan terpenuhi, dan semua $x_{ij} > 0$ dan kotak isi ada 5 sama dengan m+n-1. Sedangkan tabel 3.1 (e) merupakan penyelesaian layak yang merosot, karena kotak isi ada 4 kurang dari m+n-1.

2. Langkah - Langkah Penyelesaian Masalah Angkutan

Untuk menyelesaikan masalah angkutan digunakan metode angkutan. Metode angkutan terdiri atas 3 langkah dasar, yaitu :

1. Menyusun Penyelesaian Layak Basis Awal yang Tidak Merosot.

   Ada 2 aturan untuk menemukan suatu penyelesaian layak basis yang tidak merosot, yaitu :

   a. <u>Setiap mengisi kotak, isikan kotak secara maksimum.</u>

   Tujuannya adalah supaya penyelesaian menjadi basis dan setiap mengisi kotak ada salah satu kolom atau baris menjadi jenuh (kendala permintaan atau penawaran terpenuhi). Akibatnya kotak - kotak yang masih kosong dalam kolom atau baris yang sudah jenuh tidak diperhitungkan lagi.

Kendala penawaran dan permintaan ditulis dalam bentuk sebagai berikut :

$$\sum_{j=1}^{n} x_{ij} = x_{i1}+x_{i2}+\ldots+x_{in} = a_i, \text{ untuk } i = 1,2,\ldots,m \tag{3-8}$$

$$\sum_{i=1}^{m} x_{ij} = x_{1j}+x_{2j}+\ldots+x_{mj} = b_j, \text{ untuk } j = 1,2,\ldots,n \tag{3-9}$$

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

Ditentukan bahwa $x_{ij} \geq 0$ untuk semua i,j. Berdasarkan (3-8) dan (3-9) didapat bahwa :

$$0 \leq x_{ij} \leq a_i \quad \text{dan} \quad 0 \leq x_{ij} \leq b_j$$

Kombinasi dari keduanya menghasilkan :

$$0 \leq x_{ij} \leq \text{minimum } (a_i, b_j)$$

untuk $i = 1,2,\ldots,m$ dan $j = 1,2,\ldots,n$

Untuk itu masing-masing x_{ij} terbatas ke bawah dengan nol dan terbatas ke atas dengan minimum (a_i, b_j). Jadi masalah angkutan selalu mempunyai penyelesaian layak yang terbatas.

Jadi jika kita pilih K_{ij} kita tetapkan nilai x_{ij} dengan

$$\boxed{x_{ij} = \text{minimum } (a_i, b_j)}$$

b. <u>Dalam mengisi kotak yang bukan terakhir, jangan sampai baris dan kolom yang dipilih jenuh bersama - sama.</u>

Tujuannya adalah supaya penyelesaian layak basis tidak merosot. Jika $a_i < b_j$ maka $x_{ij} = a_i$ dan baris i menjadi jenuh. Permintaan pada kolom j berkurang menjadi $b_j - a_i$. Jika $a_i > b_j$ maka $x_{ij} = b_j$ dan kolom j menjadi jenuh. Penawaran pada baris i berkurang menjadi $a_i - b_j$. Jika $a_i = b_j$ maka baris i dan kolom j akan jenuh bersama-sama. Hal ini mengakibatkan penyelesaian merosot. Karena kotak isi menjadi kurang dari m+n-1. Dalam pengisian tabel awal, hal ini harus dihindari dengan cara memilih urutan yang lain atau ganti metode.

2. Uji Optimum

Uji Optimum ini digunakan untuk menguji apakah penyelesaian sudah optimum atau belum. Dengan kata lain apakah fungsi sasaran sudah minimum.

3. Memperbaiki Penyelesaian jika belum Optimum

Jika dalam uji optimum didapat tabel belum optimum maka kita perbaiki tabel sampai didapat penyelesaian yang optimum.

B. PENENTUAN PENYELESAIAN AWAL

Pada prinsipnya tidak ada keharusan memilih urutan dalam mengisi alokasi kotak demi kotak untuk mendapatkan penyelesaian layak basis awal. Di bawah ini akan dipelajari beberapa metode penyusunan penyelesaian layak basis awal supaya proses perhitungan lebih mudah. Urutan langkah kerja tiap metode ini kalah kuat dibandingkan dengan aturan untuk menemukan suatu penyelesaian layak basis yang tidak merosot.

Ada beberapa metode yang dapat digunakan untuk menentukan penyelesaian awal, yaitu :

1. Metode Sudut Barat Laut
2. Metode Baris Minimum
3. Metode Kolom Minimum
4. Metode Matriks Minimum
5. Metode Vogel
6. Metode Russel

Sekarang kita pelajari metode di atas satu per satu :

1. Metode Sudut Barat Laut

Langkah-langkah pengerjaan Metode Sudut Barat Laut adalah :

a. Susun tabel kendala lengkap dengan ongkosnya.
b. Urutan pengisian dimulai dari sudut barat laut (yaitu kotak sudut kiri atas). Sebut kotak ini K_{11}.
c. Pengisian kotak diteruskan menjalar ke kanan atau ke bawah tergantung mana yang belum jenuh.

Contoh 3.1 :

Ada sejenis barang yang harus diangkut dari 4 tempat asal (O_1, O_2, O_3, O_4) ke 5 tempat tujuan $(D_1, D_2, D_3, D_4, D_5)$.

Banyaknya persediaan atau penawaran di O_1, O_2, O_3, O_4 adalah 50,40,60,30 satuan. Jumlah barang yang diangkut dari setiap tempat asal tidak boleh melebihi persediaan barang yang ada. Jumlah permintaan dari D_1, D_2, D_3, D_4, D_5 masing-masing sebesar 30,35,50,20,45 satuan. Jumlah permintaan ini harus dipenuhi. Biaya angkut per satuan barang dari tempat asal ke tempat tujuan adalah sebagai berikut :

	D_1	D_2	D_3	D_4	D_5
O_1	4	6	3	1	2
O_2	9	5	7	3	4
O_3	3	8	4	6	5
O_4	5	7	8	2	5

Bagaimana cara menentukan banyaknya satuan barang yang harus dikirim dari masing-masing tempat asal ke masing-masing tempat tujuan agar ongkos pengiriman total menjadi minimum ?.

Untuk menyelesaikan soal di atas langkah pertama kita buat tabel kendala lengkap dengan ongkosnya.
Bentuk tabel kendala lengkap dengan ongkosnya masalah di atas terdapat pada Tabel 3.3.
Selanjutnya kita mencari penyelesaian layak basis awal yang tidak merosot dengan menggunakan Metode Sudut Barat Laut dengan urutan pengisian $K_{11}, K_{12}, K_{22}, K_{23}, K_{33}, K_{34}, K_{35}, K_{45}$. Kotak isi ada 8 sama dengan m+n-1 berarti penyelesaian layak basis tidak merosot. Lihat Tabel 3.4.

Tabel 3.3
Bentuk Tabel Kendala Lengkap dengan Ongkosnya
Contoh 3.1

	D_1	D_2	D_3	D_4	D_5	
O_1	4	6	3	1	2	50
O_2	9	5	7	3	4	40
O_3	9	8	4	6	5	60
O_4	5	7	8	2	5	30
	30	35	50	20	45	180

Tabel 3.4
Metode Sudut Barat Laut

	D_1	D_2	D_3	D_4	D_5	a_i
O_1	(4) 30	(6) 20	(3)	(1)	(2)	50
O_2	(9)	(5) 15	(7) 25	(3)	(4)	40
O_3	(3)	(8)	(4) 25	(6) 20	(5) 15	60
O_4	(5)	(7)	(8)	(2)	(5) 30	30
b_j	30	35	50	20	45	180

Fungsi sasaran yang sesuai dengan penyelesaian layak basis itu adalah :

$$f = 4.30 + 6.20 + 5.15 + 7.25 + 4.25 + 6.20 + 5.15 + 5.30$$
$$= 120 + 120 + 75 + 175 + 100 + 120 + 75 + 150$$
$$= 935 \quad \blacksquare$$

Pengisian dengan metode Sudut barat laut ini belum memperhatikan ongkos, yang diperhatikan hanya urutan pengisian. Urutan pengisian dimulai dari kolom dan baris terkecil. Hal ini mempermudah bila harus diprogramkan dengan bahasa komputer, karena komputer bekerja menurut urutan dari kecil ke besar.

2. Metode Baris Minimum

Langkah-langkah pengerjaan Metode Baris Minimum adalah :

a. Susun tabel kendala lengkap dengan ongkosnya.

b. Perhatikan baris 1, kemudian pilih ongkos terkecil pada baris tersebut.

Andaikan ongkos terkecil pada baris 1 adalah c_{1r}, isikan K_{1r} secara maksimum dengan X_{1r} = min (a_1, b_r). jika ada lebih dari satu ongkos terkecil maka pilih salah satu secara sembarang.

c. Jika baris 1 belum jenuh pilih lagi ongkos terkecil yang lain pada baris 1 dan ulangi proses b.

d. Jika baris 1 sudah jenuh, pindah ke baris selanjut - nya dan ulangi proses b sesuai baris yang dipilih.

Contoh 3.2 :

Kita mencari penyelesaian layak basis awal yang tidak merosot pada contoh 3.1 dengan menggunakan metode baris minimum. Perhatikan Tabel 3.5. Urutan pengisian adalah $K_{14}, K_{15}, K_{25}, K_{22}, K_{31}, K_{33}, K_{42}, K_{43}$. Kotak isi ada 8 sama dengan m+n-1, berarti penyelesaian layak basis tersebut tidak merosot. Fungsi sasaran yang sesuai dengan penyelesaian layak basis itu adalah :

$$f = 1.20 + 2.30 + 5.25 + 4.15 + 3.30 + 4.30 + 7.10 + 8.20$$

$$= 20 + 60 + 125 + 60 + 90 + 120 + 70 + 160$$

$$= 705 \quad \blacksquare$$

Tabel 3.5
Metode Baris Minimum

	D_1	D_2	D_3	D_4	D_5	a_i
O_1	4	6	3	1 20	2 30	50
O_2	9	5 25	7	3	4 15	40
O_3	3 30	8	4 30	6	5	60
O_4	5	7 10	8 20	2	5	30
b_j	30	35	50	20	45	180

Pengisian dengan metode baris minimum ini masih memperhatikan urutan, yaitu di mulai dari baris yang paling awal dan sudah memperhatikan ongkos.

3. Metode Kolom Minimum

Langkah-langkah pengerjaan Metode Kolom Minimum adalah :

a. Susun tabel kendala lengkap dengan ongkosnya.

b. Perhatikan kolom 1, kemudian pilih ongkos terkecil pada kolom tersebut.

   Misalkan ongkos terkecil pada kolom 1 adalah c_{p1} maka isikan K_{p1} secara maksimum dengan x_{p1} = min

(a_p, b_1).

c. Jika kolom 1 belum jenuh, pilih lagi ongkos terkecil pada kolom 1 dan ulangi proses b.

d. Jika kolom 1 sudah jenuh, pindahlah ke kolom selanjutnya dan ulangi proses b sesuai kolom yang di pilih.

<u>Contoh 3.3 :</u>

Kita mencari penyelesaian layak basis awal yang tidak merosot pada contoh 3.1 dengan menggunakan metode kolom minimum. Perhatikan Tabel 3.6 berikut ini :

Tabel 3.6
Metode Kolom Minimum

	D_1	D_2	D_3	D_4	D_5	a_i
O_1	(4)	(6)	(3)	(1) 20	(2) 30	50
O_2	(9)	(5) 35	(7) 5	(3)	(4)	40
O_3	(3) 30	(8)	(4) 30	(6)	(5)	60
O_4	(5)	(7)	(8) 15	(2)	(5) 15	30
b_j	30	35	50	20	45	180

Urutan pengisian $K_{31}, K_{22}, K_{33}, K_{23}, K_{43}, K_{14}, K_{15}, K_{45}$.

Kotak isi ada 8 sama dengan m+n-1, berarti penyelesaian layak basis tidak merosot. Fungsi sasaran yang sesuai dengan penyelesaian layak basis itu adalah :

$$f = 1.20 + 2.30 + 5.35 + 7.5 + 3.30 + 4.30 + 8.15 + 5.15$$

$$= 20 + 60 + 175 + 35 + 90 + 120 + 120 + 75 = 695 \quad \blacksquare$$

Pada dasarnya cara kerja metode kolom minimum ini sama dengan metode baris minimum. Metode kolom minimum juga memperhatikan urutan dan memperhitungkan ongkos.

4. Metode Matriks Minimum

Langkah-langkah pengerjaan Metode Matriks Minimum adalah :

a. Susun tabel kendala lengkap dengan ongkosnya.

b. Pilih ongkos terkecil dari tabel awal.

Misalkan ongkos terkecil yang terpilih adalah c_{ij} maka K_{ij} diisi dengan x_{ij} = min (a_i, b_j). Jika ada ongkos terkecil yang sama pilih salah satu secara sembarang.

c. Ulangi proses b sampai semua baris dan kolom jenuh.

Prinsip kerja metode matriks minimum adalah mengambil urutan pengisian dari ongkos yang paling kecil. Metode ini merupakan gabungan dari metode baris minimum dan metode kolom minimum.

Contoh 3.4 :

Kita mencari penyelesaian layak basis awal yang tidak merosot pada contoh 3.1 dengan menggunakan metode matriks minimum. Perhatikan Tabel 3.7. Urutan pengisian adalah $K_{14}, K_{15}, K_{31}, K_{33}, K_{25}, K_{22}, K_{42}, K_{43}$. Kotak isi ada 8 sama dengan m+n-1, berarti penyelesaian layak basis tidak merosot. Fungsi sasaran yang sesuai dengan penyelesaian layak basis di atas adalah :

$$f = 1.20 + 2.30 + 5.25 + 4.15 + 3.30 + 4.30 + 7.10 + 8.20$$

$$= 20 + 60 + 125 + 60 + 90 + 120 + 70 + 160$$

$$= 705 \quad ∎$$

Tabel 3.7
Metode Matriks Minimum

	D_1	D_2	D_3	D_4	D_5	a_i
O_1	4	6	3	1 20	2 30	50
O_2	9	5 25	7	3	4 15	40
O_3	3 30	8	4 30	6	5	60
O_4	5	7 10	8 20	2	5	30
b_j	30	35	50	20	45	180

5. Metode Vogel

Langkah-langkah pengerjaan Metode Vogel adalah :

a. Susun tabel kendala lengkap dengan ongkosnya

b. Hitung selisih dari dua ongkos terkecil dalam setiap baris dan dalam setiap kolom. Dengan cara demikian akan diperoleh m buah selisih dua ongkos terkecil untuk baris dan n buah selisih dua ongkos terkecil untuk kolom. Selisih dua ongkos terkecil dalam setiap baris ditulis pada kolom di sebelah kanan tabel dan selisih dari dua ongkos terkecil dalam setiap kolom ditulis pada baris di bawah tabel.

c. Pilih selisih yang terbesar di antara selisih -

selisih semua baris dan kolom. Selisih yang terbesar ini menunjukkan baris atau kolom yang terpilih. Jika ada lebih dari satu selisih terbesar dapat dipilih salah satu secara sembarang.

d. Dalam baris atau kolom yang terpilih, pilih ongkos yang paling kecil. Kotak inilah yang harus diisi. Isikan kotak terpilih secara maksimum. Jika ada lebih dari satu ongkos terkecil pilih salah satu secara sembarang.

e. Baris dan kolom yang sudah jenuh diabaikan (tidak perlu dihitung selisih dari dua ongkos terkecil lagi)

f. Ulangi proses di atas sampai semua baris dan kolom jenuh.

<u>Contoh 3.5 :</u>

Kita cari penyelesaian awal contoh 3.1 di atas dengan menggunakan metode Vogel. Perhatikan Tabel 3.8. Terjadi urutan pengisian $K_{44}, K_{15}, K_{13}, K_{33}, K_{31}, K_{22}, K_{41}, K_{21}$. Jumlah kotak isi ada 8 sama dengan m+n-1 sehingga diperoleh penyelesaian layak basis yang tidak merosot. Fungsi sasaran yang sesuai dengan penyelesaian layak basis tersebut adalah :

$$
\begin{aligned}
f &= 3.5 + 2.45 + 9.5 + 5.35 + 3.15 + 4.45 + 5.10 + 2.20 \\
&= 15 + 90 + 45 + 175 + 45 + 180 + 50 + 40 \\
&= 640 \quad \blacksquare
\end{aligned}
$$

Tabel 3.8
Metode Vogel

	D_1	D_2	D_3	D_4	D_5	a_i	
O_1	^{4}	^{6}	^{3} 5	^{1}	^{2} 45	50	1 1 $\boxed{2}$ x x x x x
O_2	^{9} 5	^{5} 35	^{7}	^{3}	^{4}	40	1 1 1 1 1 1 1 1
O_3	^{3} 15	^{8}	^{4} 45	^{6}	^{5}	60	1 1 1 1 $\boxed{2}$ x x x
O_4	^{5} 10	^{7}	^{8}	^{2} 20	^{5}	30	$\boxed{3}$ 0 0 0 0 0 0 x
b_j	30	35	50	20	45	180	
	1	1	1	1	2		
	1	1	1	x	$\boxed{2}$		
	1	1	1	x	x		
	1	1	$\boxed{3}$	x	x		
	1	1	x	x	x		
	1	$\boxed{1}$	x	x	x		
	$\boxed{1}$	x	x	x	x		
	5	x	x	x	x		

6. Metode Russel

Langkah-langkah pengerjaan Metode Russel adalah :

a. Susun tabel kendala lengkap dengan ongkosnya

b. Cari ongkos terbesar pada tiap-tiap baris dan kolom. Misal $\bar{s}_i$ adalah ongkos terbesar pada baris i ditulis di sebelah kanan tabel dan $\bar{t}_j$ adalah ongkos terbesar pada kolom j ditulis di bawah tabel.

c. Hitung $\Delta_{ij} = c_{ij} - \bar{s}_i - \bar{t}_j$
Buatlah tabel baru kemudian tulis Δ_{ij} di sudut kanan atas mengganti nilai c_{ij} dalam tabel a.

d. Pilih Δ_{ij} terkecil (jika ada Δ_{ij} terkecil yang sama pilih salah satu secara sembarang). Isikan kotak yang terpilih secara maksimum. Pengisian kotak dalam tabel mengikuti langkah-langkah pengisian metode matriks minimum.

Contoh 3.6 :

Kita mencari penyelesaian awal yang tidak merosot pada contoh 3.1 dengan menggunakan metode Russel.

Perhatikan Tabel 3.9(a) di bawah ini :

Tabel 3.9(a)
Metode Russel

	D_1	D_2	D_3	D_4	D_5	a_i	$\bar{s}_i$
O_1	4	6	3	1	2	50	6
O_2	9	5	7	3	4	40	9
O_3	3	8	4	6	5	60	8
O_4	5	7	8	2	5	30	8
b_j	30	35	50	20	45	180	
$\bar{t}_j$	9	8	8	6	5		

$\bar{s}_i$ merupakan ongkos terbesar pada baris i dan $\bar{t}_j$ merupakan ongkos terbesar pada kolom j. Kemudian kita hitung $\Delta_{ij} = c_{ij}-\bar{s}_i-\bar{t}_j$. Selanjutnya kita buat tabel baru dengan meletakkan Δ_{ij} pada sudut kanan atas menggantikan c_{ij}. Kemudian kita isi tabel tersebut dengan menggunakan metode matriks minimum. Sehingga kita dapatkan Tabel 3.9(b).

Tabel 3.9(b)
Metode Russel

	D_1	D_2	D_3	D_4	D_5	a_i
O_1	-11	-8	-11 20	-11	-9 30	50
O_2	-9	-12 35	-10	-12 5	-10	40
O_3	-14 30	-8	-12 30	-8	-8	60
O_4	-12	-9	-8	-12 15	-8 15	30
b_j	30	35	50	20	45	180

Urutan pengisian $K_{31}, K_{22}, K_{33}, K_{24}, K_{44}, K_{13}, K_{15}, K_{45}$.

Kotak isi ada 8 sama dengan m+n-1 berarti penyelesaian layak basis tidak merosot. Fungsi sasaran yang sesuai dengan penyelesaian layak basis itu adalah :

$$f = 3.20 + 2.30 + 5.35 + 3.5 + 3.30 + 4.30 + 2.15 + 5.15$$
$$= 60 + 60 + 175 + 15 + 90 + 120 + 30 + 75$$
$$= 625 \quad \blacksquare$$

Dari keenam metode penentuan penyelesaian awal yang memperhitungkan ongkos adalah metode baris minimum sampai metode Russel. Hanya metode sudut barat laut yang tidak memperhitungkan ongkos sehingga pada umumnya nilai f yang diperoleh masih lebih besar dibanding metode yang lain. Metode Vogel dan metode Russel lebih banyak memerlukan perhitungan dibanding metode matriks minimum. Penentuan penyelesaian awal dengan metode Russel dan metode matriks minimum mempunyai kesamaan, yaitu pada pengisian tabel. Dalam metode matriks minimum kita langsung memilih ongkos terkecil kemudian mengisi kotak terpilih secara maksimum. Sedangkan dalam metode Russel kita harus mencari ongkos terbesar dari tiap baris ($\bar{s}_i$) dan ongkos terbesar dari tiap kolom ($\bar{t}_j$) kemudian dihitung $\Delta_{ij} = c_{ij} - \bar{s}_i - t_j$. Setelah Δ_{ij} dihitung, isi tabel dimulai dari Δ_{ij} terkecil. Jadi metode Russel ini menuntut lebih banyak perhitungan dibanding metode matriks minimum. Masing-masing metode di atas mempunyai kelebihan dan kekurangan. Oleh karena itu tidak dapat dikatakan bahwa metode yang satu lebih baik dari metode yang lain. Untuk penyelesaian secara manual gunakanlah metode yang mudah dan cepat serta memperhitungkan ongkos. Pada umumnya metode matriks minimum lebih cepat menuju optimum. Sedangkan untuk komputasi dengan komputer dipilih metode yang sesuai dengan bahasa komputer yaitu metode yang bekerja menurut urutan dari kecil ke besar. Metode yang hanya memperhatikan urutan adalah metode sudut barat laut.

Dari semua metode yang diuraikan di atas bila ada langkah yang mengakibatkan tabel awal merosot maka pedoman langkah pengerjaan metode harus dikalahkan dan ikuti aturan untuk menentukan suatu penyelesaian layak basis yang tidak merosot.

C. UJI OPTIMUM

Untuk mengetahui apakah suatu penyelesaian layak basis sudah optimum atau belum, dengan kata lain apakah nilai f yang dihasilkan sudah minimum, maka kita harus melakukan uji optimum.

Perhatikan Tabel 3.10.(a) soal masalah angkutan sederhana dengan 2 sumber dan 2 tujuan. Penyelesaian layak basis awal disusun menurut metode sudut barat laut dan menghasilkan suatu penyelesaian layak basis yang tidak merosot dengan $x_{21} = 0$.

Tabel 3.10.(a)

	D_1	D_2	a_i
O_1	40 [3]	30 [2]	70
O_2	[1]	30 [4]	30
b_j	40	60	100

Tabel 3.10.(b)

	D_1	D_2	a_i
O_1	40−1 [3]	30+1 [2]	70
O_2	+1 [1]	30−1 [4]	30
b_j	40	60	100

Perhatikan tabel 3.10.(b), untuk mengetahui apakah tabel ini sudah optimum diadakan percobaan dengan mengisi $x_{21} = 1$ satuan. Supaya jumlah ke kanan dan ke bawah tidak berubah maka x_{22} dikurangi 1 satuan, x_{12} ditambah 1 satuan dan x_{11} dikurangi 1 satuan. Kemudian dihitung nilai f sebelum dan sesudah perubahan. Kita hitung f yang baru dikurangi f yang lama : $\Delta f = \bar{f} - f$.

$$\bar{f} = 3(40-1) + 2(30+1) + 4(30-1) + 1(0+1) = 296$$

$$f = 3(40) + 2(30) + 4(30) + 1(0) = 300$$

$$\Delta f = \bar{f} - f = 3(-1) + 2(1) + 4(-1) + 1(1) = -4$$

Nilai $\Delta f = -4$, ini berarti percobaan menunjukkan bahwa tabel awal belum optimum, karena dengan mengisi x_{21} dengan nilai positif, nilai f dapat disusutkan lagi. Contoh di atas memperlihatkan cara untuk menghitung Δf tanpa menghitung f dan $\bar{f}$. Cukup menjumlahkan ongkos-ongkosnya dengan tanda berselang - seling positif negatif.

Definisi : Ongkos kesempatan c'_{ij} adalah besarnya penyusutan nilai f yang terjadi bila dalam K_{ij} diisi satu unit alokasi.

Dari definisi dapat ditulis :

$$\boxed{c'_{ij} = -\Delta f} \qquad \text{dengan } \Delta f = \bar{f} - f$$

$\bar{f}$ adalah nilai f (ongkos angkut total) yang diperoleh bila K_{ij} diisi dengan 1 satuan alokasi.

Jadi bila $\Delta f < 0$ maka $c'_{ij} > 0$. Sebaliknya bila $\Delta f > 0$ maka $c'_{ij} < 0$, artinya bila K_{ij} diisi 1 satuan alokasi sehingga nilai f naik maka $c'_{ij} < 0$.

Dengan demikian dalam masalah angkutan dengan kasus minimum, suatu penyelesaian layak basis dianggap optimum bila semua $c'_{ij} \le 0$. Untuk suatu tabel dapat dihitung c'_{ij} untuk semua kotak kosong.

$$\boxed{\text{Tabel akan optimum jika : } c'_{ij} \le 0 \text{ untuk semua } i,j}$$

Dalam masalah angkutan yang meminimumkan fungsi sasaran, suatu penyelesaian layak basis dikatakan optimum bila semua $c'_{ij} \le 0$. Jika ada $c'_{ij} > 0$ maka penyelesaian layak basis belum optimum . Untuk kotak isi dengan sendirinya $c'_{ij} = 0$.

Dalam contoh di atas $c'_{ij} = -\Delta f = -(-4) = 4$, berarti K_{21} mempunyai kesempatan untuk menyusutkan f sebesar 4 satuan bila x_{21} diisi 1 satuan. Karena $c'_{ij} = 4 > 0$ maka penyelesaian layak basis belum optimum.

Untuk uji optimum kita akan memakai 2 metode, yaitu :

1. Metode Stepping Stone
2. Metode MODI

1. Metode Stepping Stone

Langkah-langkah pengerjaan metode Stepping Stone :

a. Mulai dengan tabel penyelesaian awal yang tidak merosot.

b. Hitung $c'_{ij} = -\Delta f$ untuk setiap kotak kosong, lebih dahulu dicari suatu lintasan tertutup yang menghubungkan kotak kosong K_{ij} dengan kotak-kotak basis. Misalkan K_{ij} adalah kotak kosong, dari K_{ij} dibuat lintasan tertutup yang menghubungkan K_{ij} dengan kotak-kotak basis, sehingga terjadi lintasan dengan urutan $K_{ij}, K_{ir}, K_{ur}, \ldots, K_{pq}, K_{pj}$. Supaya tidak mengubah nilai a_i dan b_j maka harus ada alokasi yang ditambah dan dikurangi. Kotak yang dilalui lintasan diberi tanda berselang-seling positif negatif yang berarti bahwa kotak dengan tanda positif menunjukkan bahwa alokasi kotak itu perlu ditambah, sedangkan kotak dengan tanda negatif menunjukkan bahwa alokasi kotak itu harus dikurangi. Sehingga nilai c'_{ij} dapat dihitung dengan cara :

$$c'_{ij} = c_{ir} - c_{ur} + \ldots - c_{pq} + c_{pj} - c_{ij}$$

Dalam tulisan ini nilai c'_{ij} yang diperoleh ditulis pada K_{ij} di sudut kiri atas diberi kotak kecil.

c. Jika masih ada $c'_{ij} > 0$ berarti tabel belum optimum, maka diperlukan memperbaiki tabel dengan cara mengganti satu variabel basisnya. Kotak dengan $c'_{ij} > 0$ ini menjadi calon untuk diisi, berarti x_{ij} yang sesuai akan menjadi basis. Bila ada $c'_{ij} > 0$ lebih dari satu, pilih salah satu secara sembarang.

Contoh 3.7 :

Kita uji keoptimuman Tabel 3.7 dengan metode stepping stone.

Tabel 3.7

(4)	(6)	(3)	20 (1)	30 (2)	50
(9)	25 (5)	(7)	(3)	15 (4)	40
30 (3)	(8)	30 (4)	(6)	(5)	60
(5)	10 (7)	20 (8)	(2)	(5)	30
30	35	50	20	45	180

Perhatikan Tabel 3.11 terlihat bahwa K_{13}, K_{41}, K_{44}, K_{45} mempunyai $c'_{ij} > 0$ berarti tabel belum optimum.

Tabel 3.11

Menghitung c'_{ij} dengan Metode Stepping Stone

K_{ij}	LINTASAN	∂f	c'_{ij}
K_{11}	$K_{11}, K_{15}, K_{25}, K_{22}, K_{42}, K_{43}, K_{33}, K_{31}$ +4 −2 +4 −5 +7 −8 +4 −3	1	−1
K_{12}	$K_{12}, K_{22}, K_{25}, K_{15}$ +6 −5 +4 −2	3	−3
K_{13}	$K_{13}, K_{43}, K_{42}, K_{22}, K_{25}, K_{15}$ +3 −8 +7 −5 +4 −2	−1	1
K_{21}	$K_{21}, K_{31}, K_{33}, K_{43}, K_{42}, K_{22}$ +9 −3 +4 −8 +7 −5	4	−4
K_{23}	$K_{23}, K_{22}, K_{42}, K_{43}$ +7 −5 +7 −8	1	−1
K_{24}	$K_{24}, K_{14}, K_{15}, K_{25}$ +3 −1 +2 −4	0	0
K_{32}	$K_{32}, K_{42}, K_{43}, K_{33}$ +8 −7 +8 −4	5	−5
K_{34}	$K_{34}, K_{14}, K_{15}, K_{25}, K_{22}, K_{42}, K_{43}, K_{33}$ +6 −1 +2 −4 +5 −7 +8 −4	5	−5
K_{35}	$K_{35}, K_{33}, K_{43}, K_{42}, K_{22}, K_{25}$ +5 −4 +8 −7 +5 −4	3	−3
K_{41}	$K_{41}, K_{31}, K_{33}, K_{43}$ +5 −3 +4 −8	−2	2
K_{44}	$K_{44}, K_{14}, K_{15}, K_{25}, K_{22}, K_{42}$ +2 −1 +2 −4 +5 −7	−3	3
K_{45}	$K_{45}, K_{25}, K_{22}, K_{42}$ +5 −4 +5 −7	−1	1

2. Metode MODI

Langkah-langkah metode MODI (Modified Distribution) sebagai berikut :

a. Mulai dengan tabel penyelesaian awal yang tidak merosot.

b. Tentukan bilangan baris (U_i) dan bilangan kolom (V_j) di mana U_i dan V_j dapat ditentukan dengan rumus :

$$\boxed{U_i + V_j = c_{ij}} \qquad \text{atau} \qquad U_i + V_j - c_{ij} = 0$$

Andaikan $c_{ir}, c_{iq}, c_{pq}, c_{ps}, \ldots, c_{ty}, c_{tr}$ adalah $m+n-1$ ongkos dalam penyelesaian layak basis suatu masalah angkutan dengan m sumber dan n tujuan, sehingga ongkos-ongkos tersebut dapat dinyatakan sebagai :

$$c_{ir} = U_i + V_r$$
$$c_{iq} = U_i + V_q$$
$$c_{pq} = U_p + V_q$$
$$c_{ps} = U_p + V_s$$
$$\vdots$$
$$c_{ty} = U_t + V_y$$
$$c_{tr} = U_t + V_r$$

Dalam menentukan U_i dan V_j dapat dipilih satu secara sembarang sebagai yang pertama dan diisi dengan nilai sembarang maka yang lain akan tertentu.

c. Untuk kotak kosong dapat dihitung :

$$\boxed{c'_{ij} = U_i + V_j - c_{ij}}$$

Andaikan K_{ij} adalah kotak kosong. Dibuat lintasan tertutup dari K_{ij} ke kotak basis. Misalkan kotak basis yang terlewati lintasan adalah K_{ir}, K_{pr}, K_{pq},

$\ldots, K_{ty}, K_{tj}$ maka dapat dihitung ongkos kesempatannya yaitu :

$$
\begin{aligned}
c'_{ij} &= c_{ir} - c_{pr} + c_{pq} - \ldots - c_{ty} + c_{tj} - c_{ij} \\
&= U_i + V_r - U_p - V_r + U_p - V_q - \ldots - U_t - V_y + U_t + V_j - c_{ij} \\
&= U_i + V_j - c_{ij}
\end{aligned}
$$

d. Jika masih ada $c'_{ij} > 0$ berarti tabel belum optimum. Kotak dengan $c'_{ij} > 0$ ini menjadi calon untuk diisi, berarti x_{ij} yang sesuai akan menjadi basis.

e. Bila ada lebih dari satu $c'_{ij} > 0$ dipilih salah satu secara sembarang.

Contoh 3.8 :

Kita uji keoptimuman Tabel 3.7 dengan menggunakan metode MODI. Perhatikan Tabel 3.12.

Tabel 3.12
Metode MODI

	D_1	D_2	D_3	D_4	D_5	a_i	U_i
O_1	$\boxed{-1}$ 4	$\boxed{-3}$ 6	$\boxed{1}$ 3	20^{1}	30^{2}	50	-4
O_2	$\boxed{-4}$ 9	25^{5}	$\boxed{-1}$ 7	$\boxed{0}$ 3	15^{4}	40	-2
O_3	30^{3}	$\boxed{-5}$ 8	30^{4}	$\boxed{-5}$ 6	$\boxed{-3}$ 5	60	-4
O_4	$\boxed{2}$ 5	10^{7}	20^{8}	$\boxed{3}$ 2	$\boxed{1}$ 5	30	0
b_j	30	35	50	20	45	180	
V_j	7	7	8	5	6		

Urutan menentukan U dan V :

$U_4=0 \longrightarrow V_2=7 \longrightarrow U_2=-2 \longrightarrow V_5=6 \longrightarrow U_1=-4 \longrightarrow V_4=5$

$\searrow V_3=8 \longrightarrow U_3=-4 \longrightarrow V_1=7$

Sebagai bilangan pertama kita pilih $U_4 = 0$ maka akan diperoleh U_i dan V_j yang lain. Sebenarnya dalam menentukan nilai U_4 tidak harus nol. Boleh 100 atau yang lain. Tetapi jika kita memilih bilangan yang besar kita akan kesulitan dalam menghitung nilai U_i dan V_j yang lain. Maka untuk mempermudah pekerjaan kita, kita memilih nilai U_i atau V_j yang pertama adalah nol.

D. PERBAIKAN PENYELESAIAN

Setelah adanya uji optimum dan ternyata masih ada $c'_{ij} > 0$ maka diperlukan perbaikan penyelesaian dengan cara mengganti satu variabel basisnya. Untuk itu diperlukan petunjuk :

Untuk memilih variabel yang masuk :
pilih kotak dengan $c'_{ij} > 0$ terbesar

Untuk memilih variabel yang masuk basis dipilih kotak dengan $c'_{ij} > 0$ karena pola soal meminimumkan fungsi sasaran dan dipilih yang terbesar supaya penyelesaian cepat optimum. Sesuai definisi c'_{ij} bila dipilih $c'_{ij} < 0$ nilai f akan bertambah padahal yang dicari f yang makin mendekati minimum.

Langkah-langkah penggantian variabel basis adalah :

1. Kotak kosong yang terpilih untuk diisi ditentukan dahulu lintasan tertutupnya seperti dalam metode stepping stone, kotak yang dilalui lintasan tertutup diberi

tanda berselang - seling positif negatif mulai dari kotak terpilih bertanda positif.

2. Pilih alokasi kotak bertanda negatif yang paling kecil, itulah alokasi maksimum yang dapat digeser dan masuk ke kotak terpilih. Jika dipilih alokasi kotak yang tidak terkecil maka setelah perhitungan dilakukan akan terdapat alokasi kotak bertanda negatif.
3. Lakukan penggeseran dan dengan sendirinya kotak negatif dengan alokasi terkecil akan menjadi kosong berarti dialah basis yang keluar. Sesudah ketiga langkah itu dikerjakan kita kembali ke uji optimum, demikian seterusnya sampai didapat tabel yang optimum.

Contoh 3.9 :

Diambil contoh 3.8. Di dalam Tabel 3.12 ternyata masih ada $c'_{ij} > 0$ yaitu $c'_{45} = c'_{13} = 1$, $c'_{41} = 2$ dan $c'_{44} = 3$.

Tabel 3.13

Tabel Perbaikan I

	D_1	D_2	D_3	D_4	D_5	a_i
O_1	4	$\boxed{1}$ 6	3	1; 20 (−)	2; 30 (+)	50
O_2	9	5; 25 (+)	7	3	4; 15 (−)	40
O_3	3; 30	8	4; 30	6	5	60
O_4	$\boxed{2}$ 5	7; 10 (−)	8; 20	$\boxed{3}$ 2 (+)	$\boxed{1}$ 5	30
b_j	30	35	50	20	45	180

Jadi tabel itu belum optimum, kita pilih $c'_{ij}>0$ terbesar yaitu c'_{44}. Jika ada $c'_{ij} > 0$ terbesar lebih dari satu maka dapat kita pilih salah satu secara sembarang. Perhatikan Tabel 3.13.

Supaya tidak terlalu banyak data di dalam satu kotak dalam tabel maka untuk $c'_{ij} < 0$ tidak perlu ditulis dalam tabel. Lintasan yang terjadi adalah $K_{44}, K_{42}, K_{22}, K_{25}, K_{15}, K_{14}$ dengan tanda berselang-seling positif negatif. Alokasi kotak bertanda negatif yang terkecil adalah K_{42} dengan $x_{42} = 10$, maka alokasi inilah yang digeser, didapat tabel 3.14.

Tabel 3.14
T a bel perb a ikan II

	D_1	D_2	D_3	D_4	D_5	a_i	U_i
O_1	^{4}	^{6}	$\boxed{4}$ ^{3} (+)	^{1} 10 (−)	^{2} 40	50	0
O_2	^{9}	^{5} 35	$\boxed{2}$ ^{7}	^{3}	^{4} 5	40	2
O_3	^{3} 30	^{8}	^{4} 30	^{6}	$\boxed{0}$ ^{5}	60	3
O_4	^{5}	^{7}	(−) ^{8} 20	(+) ^{2} 10	^{5}	30	1
b_j	30	35	50	20	45	180	
V_j	0	3	7	1	2		

Masih ada $c'_{ij} > 0$ yaitu $c'_{13} = 4$ dan $c'_{23} = 2$. Kita pilih $c'_{ij} > 0$ terbesar yaitu $c'_{13} = 4$. Dibuat lintasan tertutup dengan urutan $K_{13}, K_{14}, K_{44}, K_{43}$ dengan tanda berselang-seling positif negatif. Alokasi kotak bertanda negatif yang terkecil adalah K_{14} dengan $x_{14} = 10$ maka didapat tabel 3.15.

Tabel 3.15
Tabel perbaikan III

	D_1	D_2	D_3	D_4	D_5	a_i	U_i
O_1	(4)	(6)	(3) 10	(1)	(2) 40	50	0
O_2	(9)	(5) 35	(7)	(3)	(4) 5	40	2
O_3	(3) 30	(8)	(4) 30	(6)	(5)	60	3
O_4	$\boxed{2}$ (5)	$\boxed{1}$ (7)	(8) 10	(2) 20	$\boxed{2}$ (5)	30	1
b_j	30	35	50	20	45	180	
V_j	0	3	7	1	2		

Masih ada $c'_{ij} > 0$ yaitu $c'_{41} = c'_{45} = 2$, $c'_{42} = 1$, karena ada $c'_{ij} > 0$ terbesar lebih dari satu kita pilih salah satu secara sembarang misalnya c'_{41}, maka didapat tabel 3.16.

Tabel 3.16
Tabel Perbaikan IV

	D_1	D_2	D_3	D_4	D_5	a_i
O_1	(4)	(6)	(3) 10	(1)	(2) 40	50
O_2	(9)	(5) 35	(7)	(3)	(4) 5	40
O_3	(3) 20	(8)	(4) 40	(6)	(5)	60
O_4	(5) 10	(7)	(8)	(2) 20	(5)	30
b_j	30	35	50	20	45	180

Semua $c'_{ij} \leq 0$ berarti tabel sudah optimum. Penyelesaian optimum disajikan dalam bentuk tabel yaitu Tabel 3.16

dengan nilai program :

$$f = 3.10 + 2.40 + 5.35 + 4.5 + 3.20 + 4.40 + 5.10 + 2.20$$
$$= 30 + 80 + 175 + 20 + 60 + 160 + 50 + 40$$
$$= 615 \quad \blacksquare$$

E. PILIHAN PENYELESAIAN OPTIMUM

Penyelesaian optimum masalah angkutan dapat lebih dari satu. Ciri bahwa terdapat pilihan penyelesaian optimum adalah bila dalam tabel optimum terdapat kotak kosong dengan $c'_{ij} = 0$. Sesuai dengan definisi c'_{ij} kotak kosong dengan $c'_{ij} = 0$ berarti bahwa bila kotak kosong ini diisi tidak akan menambah atau mengurangi nilai program. Cara pengisian kotak dengan $c'_{ij} = 0$ tersebut yaitu dengan cara membuat lintasan tertutup seperti dalam metode stepping stone.

<u>Contoh 3.10 :</u>

Pada tabel 3.16 terdapat kotak kosong dengan $c'_{ij} = 0$ yaitu c'_{45} berarti terdapat penyelesaian optimum lebih dari satu. Perhatikan Tabel 3.17. Kita buat lintasan tertutup dimulai dari K_{45} dengan tanda berselang-seling positif negatif. Urutan lintasan tertutup adalah $K_{45}, K_{15}, K_{13}, K_{33}, K_{31}, K_{41}$ dengan alokasi kotak bertanda negatif terkecil adalah 10. Pilihan penyelesaian optimum disajikan dalam bentuk tabel yaitu Tabel 3.17 dengan nilai program :

$$f = 3.20 + 2.30 + 5.35 + 4.5 + 3.30 + 4.30 + + 2.20 + 5.10$$

$$= 60 + 60 + 175 + 20 + 90 + 120 + 40 + 50$$

$$= 615$$

Tabel 3.17
Tabel Pilihan Penyelesaian Optimum

	D_1	D_2	D_3	D_4	D_5	a_i
O_1	(4)	(6)	(3) 20	(1)	(2) 30	50
O_2	(9)	(5) 35	(7)	(3)	(4) 5	40
O_3	(3) 30	(8)	(4) 30	(6)	(5)	60
O_4	(5)	(7)	(8)	(2) 20	(5) 10	30
b_j	30	35	50	20	45	180

Ternyata nilai program yang didapat memang sama dengan nilai program penyelesaian optimum yang pertama.

F. KEMEROSOTAN

Suatu penyelesaian layak basis masalah angkutan dengan m sumber dan n tujuan terdiri atas m+n-1 buah $x_{ij}>0$. Jika banyaknya $x_{ij}>0$ kurang dari m+n-1 maka penyelesaian layak basis disebut merosot.

Kemerosotan dalam masalah angkutan tidak boleh diabaikan karena bila merosot kita tidak dapat membuat lintasan tertutup serta penentuan U_i dan V_j dalam uji optimum dengan metode MODI tidak dapat berjalan. Kemerosotan dapat terjadi baik sewaktu kita membuat tabel awal maupun dalam tabel bukan awal, yaitu pada waktu menentukan basis berikutnya.

Untuk kemerosotan pada tabel awal jalan keluarnya adalah memilih metode penentuan penyelesaian awal supaya penyelesaian tidak merosot atau dengan cara melanggar sedikit pedoman metode yang digunakan tetapi tetap mengikuti aturan untuk menemukan suatu penyelesaian layak basis. Kemerosotan pada tabel awal ini dapat terjadi apabila kita mengisi suatu kotak yang bukan terakhir baris dan kolom yang bersesuaian jenuh bersama-sama.

Contoh 3.11 :

Pada waktu pengisian tabel 3.6. seharusnya setelah kita mengisi K_{31},K_{22} kita sampai pada K_{13}. Jika kita isi K_{13} maka baris 1 dan kolom 3 akan jenuh bersama-sama karena $a_1 = b_3 = 50$, maka kotak ini dilompati dan kita lanjutkan dengan mengisi K_{33},K_{23}, K_{43},K_{14},K_{15},K_{45} agar penyelesaian tidak merosot. ∎

Sedangkan kemerosotan yang terjadi pada tabel bukan awal muncul pada waktu kita menggeser alokasi kotak bertanda negatif yang paling kecil, terdapat dua atau lebih kotak yang menjadi kosong secara bersamaan. Jika ini terjadi maka jalan keluarnya adalah mengisikan alokasi ε ke dalam salah satu kotak yang baru saja menjadi kosong. Supaya kotak isi berjumlah m+n-1 dapat dipilih kotak yang akan diisi asal dipenuhi m+n-1. Alokasi ε merupakan alokasi semu. Untuk proses selanjutnya dirumuskan sifat ε sebagai berikut :

$$a + \varepsilon = a$$

$$b - \varepsilon = b$$

$$\varepsilon - \varepsilon = 0$$

Setelah kita isi ε secukupnya sehingga kotak isi menjadi m+n-1, maka tabel tidak lagi merosot dan proses dapat dilanjutkan kembali. Dalam proses selanjutnya alokasi ε dapat hilang sehingga tabel optimum tidak memuat ε lagi tetapi ada juga ε yang tidak hilang sampai didapatkan penyelesaian optimum. Dalam laporan penyelesaian optimum kotak yang memuat ε tidak ditulis karena ε ini merupakan alokasi semu. Jadi penyelesaian optimum dalam keadaan merosot. Hal ini tidak menjadi masalah karena tabel sudah optimum dan tidak ada proses lebih lanjut.

Contoh 3.12. :

Diketahui tabel kendala lengkap dengan ongkosnya seperti terlihat pada tabel berikut :

	D_1	D_2	D_3	a_i
O_1	7	7	1	100
O_2	5	6	2	100
O_3	7	9	3	40
b_j	40	120	80	240

Masalah dengan 3 sumber dan 3 tujuan di atas merupakan masalah angkutan setimbang dengan $\Sigma\ a_i = \Sigma\ b_j = 240$. Perhatikan Tabel 3.18.

Tabel 3.18

	D_1	D_2	D_3	a_i	U_i
O_1	^{7}	20^{7}	80^{1}	100	0
O_2	40^{5}	60^{6}	^{2}	100	-1
O_3	$\boxed{1}\ ^{7}$	40^{9}	^{3}	40	2
b_j	40	120	80	240	
V_j	6	7	1		

Tabel awal disusun dengan metode matriks minimum dengan urutan pengisian $K_{13}, K_{21}, K_{22}, K_{12}, K_{32}$. Kotak isi ada 5 sama dengan m+n-1, berarti penyelesaian layak basis tidak merosot. Fungsi sasaran yang sesuai dengan penyelesaian layak basis di atas adalah :

$$f = 7.20 + 1.80 + 5.40 + 6.60 + 9.40$$
$$= 140 + 80 + 200 + 360 + 360$$
$$= 1140 \quad \blacksquare$$

Setelah c'_{ij} dihitung ternyata masih ada $c'_{ij} > 0$, yaitu $c'_{31} = 1$. Perbaikan tabel terjadi dengan menggeser alokasi sebesar 40 ke K_{31}. Tabel menjadi Tabel 3.19. Kotak isi ada 7 kurang dari m+n-1 berarti penyelesaian layak basis merosot. Hal ini disebabkan karena dengan menggeser alokasi 40 ke K_{31} maka K_{21} dan K_{32} menjadi kosong bersamaan dan ini tidak dapat dihindari. Supaya

Tabel 3.19

	D_1	D_2	D_3	a_i	U_i
O_1	^{7}	20^7	80^1	100	0
O_2	ε^5	100^6	^{2}	100	-1
O_3	40^7	^{9}	^{3}	40	2
b_j	40	120	80	240	
V_j	6	7	1		

penyelesaian menjadi layak basis, jalan keluarnya adalah kotak yang menjadi kosong bersamaan salah satu kotak diisi dengan ε. Kita pilih K_{21} untuk diisi ε, dengan demikian tabel tidak lagi merosot. Setelah K_{21} diisi ε kita lanjutkan uji optimum pada Tabel 3.19 ternyata tabel sudah optimum dan penyelesaian akhir ditulis tanpa ε. Jadi penyelesaian optimum dalam keadaan merosot, seperti yang tercantum dalam tabel 3.20.

Tabel 3.20

	D_1	D_2	D_3	a_i
O_1	^{7}	20^7	80^1	100
O_2	^{5}	100^6	^{2}	100
O_3	40^7	^{9}	^{3}	40
b_j	40	120	80	240

Nilai program yang sesuai dengan penyelesaian layak basis di atas adalah :

$$f = 7.20 + 1.80 + 6.100 + 7.40$$

$$= 140 + 80 + 600 + 280 = 1100 \blacksquare$$

G. JALUR RUSAK

Untuk mengirimkan barang pesanan dari tempat asal ke tempat tujuan tidak selamanya berjalan lancar. Dalam perjalanan menuju tempat tujuan kadangkala mendapat hambatan, misalnya :

- Satu-satunya jembatan yang biasa dilewati oleh kendaraan pengangkut untuk mengantar barang dagangan ke tempat tujuan, pada waktu sedang rusak berat sehingga tidak dapat dilewati. Jadi tidak ada pengiriman barang dari tempat asal ke tempat tujuan.
- Kereta api merupakan satu-satunya alat angkut yang dapat digunakan untuk mengangkut barang dari sumber ke tujuan. Rel yang menghubungkan sumber ke tujuan sedang rusak sehingga tidak ada jalur pengiriman barang dari sumber ke tujuan.

Hal di atas dibahas dalam masalah angkutan yang disebut **masalah jalur rusak**.

Suatu hambatan akan timbul bila satu atau lebih jalur pengangkutan rusak sehingga tidak dapat digunakan (jadi alokasi harus nol). Bila kotak yang sesuai dalam tabel di tutup, maka metode angkutan tidak berjalan. Jalan keluarnya adalah ongkos pada jalur itu diganti dengan suatu bilangan M, dengan M positif besar.

Jadi :

Bila K_{pq} rusak maka ganti c_{pq} dengan M positif besar

Bila semula $x_{pq} > 0$, maka dengan adanya koefisien M yang positif besar algoritma akan segera mengeluarkan x_{pq} dari basis sehingga bernilai nol. Dengan istilah program linear berarti bahwa x_{pq} dianggap sebagai variabel semu dalam soal berpola minimum.

Contoh 3.13 :

Diketahui tabel ketentuan seperti terlihat dalam Tabel 3.21 dan diberitahukan bahwa K_{13} rusak total sehingga tidak dapat dilewati.

Tabel 3.21
Tabel Contoh 3.13

	D_1	D_2	D_3	a_i
O_1	4	3	2	40
O_2	5	4	6	40
O_3	2	7	5	40
b_j	50	50	20	120

Karena diketahui K_{13} rusak total maka c_{13} kita ganti dengan M. Jika tabel awal diisi dengan menggunakan metode matriks minimum, maka didapat urutan pengisian K_{31}, K_{12},K_{22},K_{21},K_{23}, didapat Tabel 3.22.

Kotak isi ada 5 sama dengan m+n-1, berarti penyelesaian layak basis awal tidak merosot. Kita uji keoptimuman tabel awal dengan menggunakan metode MODI.

Pada K_{13}, c'_{13} = 5 - M adalah negatif besar, karena semua $c'_{ij} < 0$ maka tabel sudah optimum. Pada K_{11}, $c'_{11} = 0$ berarti ada pilihan penyelesaian optimum. Dibuat lintasan tertutup $K_{11}, K_{21}, K_{22}, K_{12}$ dengan tanda berselang - seling positif - negatif sehingga didapat tabel 3.23.

Tabel 3.22

Penyelesaian Contoh 3.13

	D_1	D_2	D_3	a_i	U_i
O_1	4	40^3	M	40	-1
O_2	10^5	10^4	20^6	40	0
O_3	40^2	7	5	40	-3
b_j	50	50	20	120	
V_j	5	4	6		

Tabel 3.23

Pilihan Penyelesaian

	D_1	D_2	D_3	a_i
O_1	10^4	30^3	M	40
O_2	5	20^4	20^6	40
O_3	40^2	7	5	40
b_j	50	50	20	120

Tabel 3.23 ini merupakan salah satu contoh pilihan penyelesaian optimum dengan nilai program :

$$f = 4.10 + 3.30 + 4.20 + 6.20 + 2.40$$

$$= 40 + 90 + 80 + 120 + 80 = 410 \quad \blacksquare$$

H. MASALAH ANGKUTAN TAK SETIMBANG

Di depan telah dibahas masalah angkutan yang setimbang yaitu terjadi jika penawaran total sama dengan permintaan total. Pada dasarnya masalah angkutan setimbang jarang terjadi dalam kehidupan sehari-hari, biasanya justru yang tidak setimbang yang sering terjadi. Bila terjadi ketakseimbangan maka modelnya diubah menjadi masalah angkutan yang setimbang, karena kalau masalahnya tidak setimbang, algoritma di atas tidak dapat digunakan karena algoritma yang dibuat hanya untuk masalah angkutan yang setimbang saja. Kejadian masalah angkutan tidak setimbang ada 2 macam, yaitu :

1. Kejadian penawaran total lebih dari permintaan total

Keadaan $\sum a_i > \sum b_j$, yaitu penawaran total lebih besar dari permintaan total ini merupakan masalah angkutan tidak setimbang yang ternyata layak, karena permintaan dapat dipenuhi. Masalah angkutan tak setimbang yang layak dapat dirumuskan sebagai berikut :

Mencari $x_{ij} \geq 0$ untuk $i = 1,2,\ldots,m$ (3-10)

$j = 1,2,\ldots,n$

dengan kendala :

$$\sum_{j=1}^{n} x_{ij} \leq a_i \quad , i = 1,2,\ldots,m \qquad (3\text{-}11)$$

$$\sum_{i=1}^{m} x_{ij} = b_j \quad , j = 1,2,\ldots,n \qquad (3\text{-}12)$$

meminimumkan fungsi sasaran :

$$f = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}\, x_{ij} \qquad (3\text{-}13)$$

dengan ketentuan :

$$\sum a_i > \sum b_j \qquad (3\text{-}14)$$

Kendala tujuan berbentuk persamaan dengan alasan pihak sumber bermaksud meminimumkan ongkos angkutan, sehingga permintaan yang akan dipenuhi adalah permintaan minimumnya.

Masalah angkutan tak setimbang di atas dapat disetimbangkan dengan menambah variabel - variabel kelonggaran $x_{i(n+1)} > 0$, $i = 1,2,\ldots,m$ pada kendala sumber. Dengan demikian kendala (3-11) dan (3-12) menjadi :

$$\sum_{j=1}^{n} x_{ij} + x_{i(n+1)} = a_i \quad , \; i = 1,2,\ldots,m \qquad (3\text{-}15)$$

$$\sum_{i=1}^{m} x_{ij} = b_j \quad , \; j = 1,2,\ldots,n \qquad (3\text{-}16)$$

Jika (3-15) dijumlahkan atas i dan (3-16) dijumlahkan atas j diperoleh :

$$\sum_{i=1}^{m} x_{i(n+1)} = \sum_{i=1}^{m} a_i - \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} \qquad (3\text{-}17)$$

$$\sum_{i=1}^{m} x_{i(n+1)} = \sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j = b_{n+1} \qquad (3\text{-}18)$$

Ini berarti harus ditambah suatu tujuan semu untuk menampung kelebihan persediaan barang dari sumber. Pada tabel angkutan dapat ditambahkan kolom ke-(n+1) yang disebut kolom semu, dengan $b_{n+1} = \sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j$ yang bernilai positif. Ongkos pada tujuan semu ini adalah nol, karena sesungguhnya tidak ada yang diangkut lewat jalur tersebut. Sebagai akibat diberikan pedoman dalam mengisi tabel awal :

> Anggap kolom semu tidak ada, susun tabel awal dengan salah satu metode sampai kolom-kolom asli sudah jenuh semua baru kolom semu diisi.

Contoh 3.14 :

Selesaikan soal masalah angkutan tak setimbang pada tabel 3.24.

Tabel 3.24
Masalah Angkutan Tak Setimbang

	D_1	D_2	D_3	D_4	a_i
O_1	5	6	3	2	40
O_2	6	4	2	4	40
O_3	1	7	6	3	40
b_j	30	25	20	30	120 / 105

Contoh di atas adalah masalah angkutan tak setimbang dengan $\sum a_i = 120$ lebih besar dari $\sum b_j = 105$. Masalah dapat disetimbangkan dengan menambahkan satu kolom lagi yaitu D_5 dengan $c_{i5} = 0$.

Tabel awal diisi dengan menggunakan metode matriks minimum dengan urutan pengisian $K_{31}, K_{23}, K_{14}, K_{22}, K_{12}, K_{15}, K_{35}$. Kotak isi ada 7 sama dengan m+n-1, berarti tabel awal tidak merosot. Kita tulis penyelesaian awal ini dalam Tabel 3.25.

Tabel 3.25

	D_1	D_2	D_3	D_4	D_5	a_i	U_i
O_1	^{5}	$5^{\ 6}$	^{3}	$30^{\ 2}$	$5^{\ 0}$	40	0
O_2	^{6}	$20^{\ 4}$	$20^{\ 2}$	^{4}	^{0}	40	-2
O_3	$30^{\ 1}$	^{7}	^{6}	^{3}	$10^{\ 0}$	40	0
b_j	30	25	20	30	15	120	
V_j	1	6	4	2	0		

Kita uji Tabel 3.25 dengan metode MODI didapat ada satu $c'_{ij}>0$ yaitu $c'_{13}=1$. Tabel menjadi Tabel 3.26.

Tabel 3.26

	D_1	D_2	D_3	D_4	D_5	a_i	U_i
O_1	^{5}	^{6}	$5^{\ 3}$	$30^{\ 2}$	$5^{\ 0}$	40	0
O_2	^{6}	$25^{\ 4}$	$15^{\ 2}$	^{4}	^{0}	40	-1
O_3	$30^{\ 1}$	^{7}	^{6}	^{3}	$10^{\ 0}$	40	0
b_j	30	25	20	30	15	120	
V_j	1	5	3	2	0		

Tabel 3.26 sudah optimum. Dalam penulisan penyelesaian optimum tujuan semu tidak perlu di tulis, cukup diberi keterangan tambahan tentang sisa penawaran.

Penyelesaian optimum kita sajikan dalam Tabel 3.27 dengan nilai program :

$$f = 3.5 + 2.30 + 4.25 + 2.15 + 1.30 = 235 \ \blacksquare$$

Tabel 3.27

	D_1	D_2	D_3	D_4	a_i
O_1	5	6	5 (3)	30 (2)	40
O_2	6	25 (4)	15 (2)	4	40
O_3	30 (1)	7	6	3	40
b_j	30	25	20	30	120

Keterangan : ada sisa penawaran sebesar 5 pada O_1 dan 10 pada O_3

2. Kejadian penawaran total kurang dari permintaan total

Masalah dengan $\Sigma\, a_i < \Sigma\, b_j$, yaitu penawaran total kurang dari permintaan total. Sesungguhnya masalah angkutan menjadi tak layak. Jika kita tetap ingin mencari penyelesaian maka masalahnya harus kita ubah terlebih dahulu dengan menganggap b_j di sini bukan lagi sebagai permintaan minimum, melainkan kita anggap sebagai permintaan target (sejauh mungkin permintaan dapat dipenuhi kecuali bila persediaan pada sumber sudah tidak ada lagi). Masalah angkutan tak setimbang tak layak dapat dirumuskan sebagai berikut :

mencari $x_{ij} \geq 0$ untuk $i = 1,2,\ldots,m$ (3-19)

$j = 1,2,\ldots,n$

dengan kendala :

$$\sum_{j=1}^{n} x_{ij} = a_i \quad , \; i = 1,2,\ldots,m \qquad (3\text{-}20)$$

$$\sum_{i=1}^{m} x_{ij} \leq b_j \quad , \; j = 1,2,\ldots,n \qquad (3\text{-}21)$$

meminimumkan fungsi sasaran :

$$f = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}\, x_{ij} \qquad (3\text{-}22)$$

dengan ketentuan :

$$\sum_{i=1}^{m} a_i < \sum_{j=1}^{n} b_j \qquad (3\text{-}23)$$

Untuk menyelesaikan masalah angkutan tak setimbang tak layak ini dengan algoritma angkutan, terlebih dahulu masalahnya dikembalikan ke bentuk setimbang dengan menambahkan variabel-variabel surplus $x_{(m+1)j} > 0$, $j = 1,2,\ldots,n$ pada kendala tujuan. Dengan demikian kendala (3-20) dan (3-21) menjadi :

$$\sum_{j=1}^{n} x_{ij} = a_i \quad , \; i = 1,2,\ldots,m \qquad (3\text{-}24)$$

$$\sum_{i=1}^{m} x_{ij} + x_{(m+1)j} = b_j \quad , \; j = 1,2,\ldots,n \qquad (3\text{-}25)$$

Jika (3-24) dijumlahkan atas i dan (3-25) dijumlahkan atas j diperoleh :

$$\sum_{j=1}^{n} x_{(m+1)j} = \sum_{j=1}^{n} b_j - \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} \qquad (3\text{-}26)$$

$$\sum_{i=1}^{m} x_{(m+1)j} = \sum_{j=1}^{n} b_j - \sum_{i=1}^{m} a_i = a_{m+1} \qquad (3\text{-}27)$$

Ini berarti masalah angkutan tak setimbang di atas dapat disetimbangkan dengan menambah satu baris yang di sebut sumber semu. Dengan ditambahkan sumber semu seolah-olah ada sumber baru yang memberikan penawaran sebesar kekurangannya. Pada tabel angkutan dapat ditambahkan baris ke (m+1) dengan $a_{m+1} = \Sigma\, b_j - \Sigma\, a_i$ yang bernilai positif. Ongkos angkut pada baris semu adalah nol. Dalam mengisi tabel awal baris semu diisi setelah baris - baris asli sudah jenuh semua.

Contoh 3.15 :

Selesaikan soal masalah angkutan tak setimbang di bawah ini.

Tabel 3.28

	D_1	D_2	D_3	a_i
O_1	5	6	1	20
O_2	6	4	7	30
O_3	5	2	6	30
b_j	30	35	35	80 / 100

Tabel 3.28 adalah tabel masalah angkutan tak setimbang dengan $\Sigma\, a_i = 80$ kurang dari $\Sigma\, b_j = 100$. Masalah dapat disetimbangkan dengan menambah satu baris yaitu O_4 dengan $c_{4j}=0$.

Pengisian tabel awal dilakukan dengan menggunakn metode matriks minimum dengan urutan pengisian $K_{13}, K_{32}, K_{22}, K_{21}$ (baris 1,2,3 jenuh) kemudian diisi baris 4, K_{41}, K_{43}. Banyak kotak isi ada 6 sama dengan m+n-1 berarti penyelesaian layak basis tidak merosot. Tabel awal kita tulis dalam Tabel 3.29.

Tabel 3.29

	D_1	D_2	D_3	a_i	U_i
O_1	(5)	(6)	20 (1)	20	1
O_2	25 (6)	5 (4)	(7)	30	6
O_3	(5)	30 (2)	(6)	30	4
O_4	5 (0)	(0)	15 (0)	20	0
b_j	30	35	35	100	
V_j	0	-2	0		

Sesudah dilakukan uji optimum ternyata tabel sudah optimum.

Tabel 3.30
Tabel Optimum

	D_1	D_2	D_3	a_i
O_1	(5)	(6)	20 (1)	20
O_2	25 (6)	5 (4)	(7)	30
O_3	(5)	30 (2)	(6)	30
b_j	30	35	35	100

Keterangan :
ada kekurangan pengiriman sebesar 5 satuan pada D_1 dan 15 satuan pada D_3.

Penyelesaian optimum ditulis seperti tabel 3.30 dengan nilai program :

$$f = 1.20 + 6.25 + 4.5 + 2.30$$

$$= 20 + 150 + 20 + 60 = 250 \blacksquare$$

BAB IV

TEORI MASALAH ANGKUTAN

Bab IV ini akan membahas teori pendukung untuk algoritma masalah angkutan setimbang yang telah dibahas dalam BAB III. Teori pendukung itu meliputi penyelesaian layak basis masalah angkutan, sifat Unimodular matriks A, penjelasan lintasan tertutup metode Stepping Stone serta penjelasan bilangan baris dan bilangan kolom metode MODI. Untuk pembahasan lebih lanjut persamaan (2-5) sampai (2-10) dalam Bab II kita tulis kembali dalam Bab IV ini.

Masalah angkutan setimbang dengan m sumber dan n tujuan dapat dirumuskan secara matematis sebagai berikut :

mencari $x_{ij} \geq 0$ dengan $i = 1,2,...,m$

$$j = 1,2,...,n \qquad (4\text{-}1)$$

yang memenuhi kendala :

$$\sum_{j=1}^{n} x_{ij} = a_i, \qquad i = 1,2,...,m \qquad (4\text{-}2)$$

$$\sum_{i=1}^{m} x_{ij} = b_j, \qquad j = 1,2,...,n \qquad (4\text{-}3)$$

meminimumkan fungsi sasaran :

$$f = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}\, x_{ij} \qquad (4\text{-}4)$$

dengan $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$

Kendala (4-2) dan (4-3) di atas dapat diuraikan menjadi :

$$x_{11}+x_{12}+...+x_{1n} \qquad\qquad = a_1$$

$$
\begin{array}{llllll}
 & x_{21}+x_{22}+\ldots+x_{2n} & & & & = a_2 \\
 & & \ddots & & & \\
 & & & x_{m1}+x_{m2}+\ldots+x_{mn} & & = a_m \\
x_{11}+ & x_{21}+ & \ldots & x_{m1} & & = b_1 \\
\quad x_{12}+ & \quad x_{22}+ & \ldots & \quad x_{m2} & & = b_2 \\
\quad\ddots & & & & & \\
\qquad x_{1n}+ & \qquad x_{2n}+ & \ldots & & x_{mn} & = b_n
\end{array}
\qquad (4\text{-}5)
$$

Perumusan (4-5) dapat ditulis dalam bentuk matriks $Ax = b$, dengan :

$$
A = \begin{bmatrix}
11\ldots1 & 00\ldots0 & \ldots & 00\ldots0 \\
00\ldots0 & 11\ldots1 & \ldots & 00\ldots0 \\
\vdots & & & \vdots \\
00\ldots0 & 00\ldots0 & \ldots & 11\ldots1 \\
10\ldots0 & 10\ldots0 & \ldots & 10\ldots0 \\
01\ldots0 & 01\ldots0 & \ldots & 01\ldots0 \\
\vdots & & & \vdots \\
00\ldots1 & 00\ldots1 & \ldots & 00\ldots1
\end{bmatrix}
$$

$$
x = \begin{bmatrix} x_{11} \\ \vdots \\ x_{1n} \\ x_{21} \\ \vdots \\ x_{2n} \\ \vdots \\ x_{m1} \\ \vdots \\ x_{mn} \end{bmatrix}
\qquad
b = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \\ b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}
\qquad (4\text{-}6)
$$

Jika 1_n menyatakan vektor baris dengan n komponen yang semua elemennya adalah 1.

0 menyatakan vektor baris dengan n komponen yang semua elemennya adalah 0.

I_n menyatakan matriks identitas berordo n.

maka Matriks A dapat juga ditulis menjadi :

$$\begin{array}{cl} \underbrace{\begin{bmatrix} 1_n & 0 & \cdots & 0 \\ 0 & 1_n & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & 1_n \\ I_n & I_n & \cdots & I_n \end{bmatrix}}_{mn \text{ kolom}} & \begin{array}{l} \left.\begin{array}{c} \\ \\ \\ \\ \end{array}\right\} m \text{ baris} \\ \} \; n \text{ baris} \end{array} \end{array} \qquad (4\text{-}7)$$

A. RANK MATRIKS A

Matriks A adalah matriks berordo (m+n) x (mn). Matriks A mempunyai m+n baris dan mn kolom. Pada matriks A didefinisikan P_{ij} suatu vektor kolom dengan (m+n) komponen dan hanya dua komponen pada P_{ij} yang tidak 0 melainkan 1. Sehingga P_{ij} dapat kita tulis :

$$P_{ij} = e_i + e_{m+j} \qquad (4\text{-}8)$$

dengan e_k adalah vektor satuan, huruf k menunjukkan letak bilangan 1 pada baris ke-k. Baris-baris pada A dapat dikelompokkan menjadi dua himpunan, m baris pertama merupakan koefisien kendala sumber dan n baris berikutnya merupakan koefisien kendala tujuan.

Sebagai contoh P_{ij} dapat kita ambil :

$$P_{11} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ \hline 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad P_{21} = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \\ \hline 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad P_{mn} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \hline 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix}0\\0\\\vdots\\1\end{matrix}}\right\} \text{m baris} \\ \left.\vphantom{\begin{matrix}0\\0\\\vdots\\1\end{matrix}}\right\} \text{n baris} \end{matrix}$$

Dari (4-7) tampak bahwa jumlah m baris pertama pada A hasilnya 1_{mn} dan jumlah n baris berikutnya pada A menghasilkan 1_{mn}, dengan 1_{mn} adalah vektor baris dengan mn komponen yang semua elemennya adalah 1. Jadi jumlah m baris pertama pada A dikurangi jumlah n baris selanjutnya pada A menghasilkan suatu vektor nol, m+n vektor dependen linear. Oleh karena itu rank A kurang dari m+n.

Dalam Bab III A.1. telah kita simpulkan bahwa matriks A mempunyai rank m+n-1. Di bawah ini akan kita buktikan teorema rank matriks A dengan cara lain.

<u>Teorema 4.1 :</u>

Rank matriks A adalah m+n-1

<u>Bukti :</u>

Teorema di atas akan dibuktikan dengan menemukan sub matriks bujur sangkar dalam A yang mempunyai determinan berordo m+n-1 yang tidak nol.

Susunlah matriks D dari matriks A dengan mengambil kolom ke n,2n,...,mn,1,...,n-1 dan baris ke 1,...,m+n-1 (baris selanjutnya dalam A dihilangkan), sehingga :

$$|D| = \begin{vmatrix} I_m & F \\ 0 & I_{n-1} \end{vmatrix} \quad \text{dengan} \quad F = \begin{bmatrix} 1_{n-1} \\ 0 \end{bmatrix}$$

$$|D| = \begin{vmatrix} 1 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 \\ 0 & 1 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & & & & & & & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & & & & & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 \end{vmatrix} = 1$$

Karena D matriks bujur sangkar dengan ordo m+n-1 dan $|D| \neq 0$, maka $r(A) = m+n-1$. ■

B. PENYELESAIAN LAYAK BASIS

Dalam Bab III A.1. telah dijelaskan adanya penyelesaian layak dalam masalah angkutan. Keterangan dalam Bab III tersebut akan kita pakai sebagai bukti teorema berikut ini.

Teorema 4.2 :

Masalah angkutan setimbang mempunyai penyelesaian layak.

Bukti :

Karena $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j = \lambda$, kita mempunyai penyelesaian layak $x_{ij} = a_i b_j / \lambda$ untuk semua i dan j. Masing-masing $x_{ij} \geq 0$, dan (4-2) terpenuhi karena :

$$\sum_{j=1}^{n} x_{ij} = \sum_{j=1}^{n} \frac{a_i \, b_j}{\lambda} = \frac{a_i \sum_{j=1}^{n} b_j}{\lambda} = a_i$$

dan (4-3) terpenuhi, karena :

$$\sum_{i=1}^{m} x_{ij} = \sum_{i=1}^{m} \frac{a_i \, b_j}{\lambda} = \frac{b_j \sum_{i=1}^{m} a_i}{\lambda} = b_j$$

Teorema 4.3 :

Jika a_i dan b_j dalam (4-2) dan (4-3) bernilai bulat maka penyelesaian layak basis bernilai bulat.

Bukti :

Akan dibuktikan masalah angkutan dengan m sumber dan n tujuan jika diketahui a_i dan b_j bulat maka penyelesaian layak basis akan bernilai bulat pula.

Langkah pertama algoritma angkutan adalah menyusun penyelesaian layak basis awal yang tidak merosot dengan rumus $x_{ij} = \min(a_i, b_j)$. Karena a_i dan b_j bernilai bulat maka x_{ij} pasti bernilai bulat juga.

Langkah kedua algoritma angkutan adalah menguji keoptimuman penyelesaian layak basis. Uji optimum dengan metode Stepping Stone maupun MODI hanya melibatkan ongkos angkut.

Langkah ketiga algoritma angkutan adalah memperbaiki penyelesaian bila penyelesaian layak basis belum optimum. Cara memperbaiki penyelesaian layak basis dengan membuat lintasan tertutup dimulai dari kotak dengan ongkos kesempatan positif terbesar ke kotak-kotak basis dengan tanda berselang seling positif negatif. Tanda positif negatif menunjukkan operasi penjumlahan dan pengurangan. Dalam penggeseran alokasi x_{ij} operasi yang digunakan hanyalah penjumlahan dan pengurangan. Kedua operasi ini tidak akan mempengaruhi x_{ij} menjadi tidak bulat. Jadi penyelesaian layak basis akan bernilai bulat. ■

C. SIFAT UNIMODULAR MATRIKS A

Untuk menjelaskan sifat unimodular matriks A kita tulis lagi bentuk matriks A dalam persamaan (4-6)

$$A = \begin{bmatrix} 11\ldots1 & 00\ldots0 & \ldots & 00\ldots0 \\ 00\ldots0 & 11\ldots1 & \ldots & 00\ldots0 \\ \vdots & & & \vdots \\ 00\ldots0 & 00\ldots0 & \ldots & 11\ldots1 \\ 10\ldots0 & 10\ldots0 & \ldots & 10\ldots0 \\ 01\ldots0 & 01\ldots0 & \ldots & 01\ldots0 \\ \vdots & & & \vdots \\ 00\ldots1 & 00\ldots1 & \ldots & 00\ldots1 \end{bmatrix}$$

Minor M_{ij} dari matriks A adalah determinan sub matriks bujur sangkar yang diperoleh dari matriks A dengan menghapus baris ke i dan kolom ke j.

Teorema 4.4 :

Setiap minor A hanya dapat bernilai +1, -1 atau 0.

Bukti :

Andaikan A_k adalah suatu submatriks bujur sangkar berordo k yang dibentuk dari sembarang k kolom yang berbeda dan k baris yang berbeda dari matriks A. Akan dibuktikan bahwa :

$$\text{Determinan } A_k = |A_k| = +1,\ -1 \text{ atau } 0$$

Masing-masing kolom A_k dapat memuat dua buah elemen 1, sebuah elemen 1 atau tidak memuat elemen 1 sama sekali.

1. Jika A_k memuat satu atau lebih kolom yang semua elemennya 0 maka $|A_k| = 0$.
2. Jika setiap kolom A_k memuat dua buah elemen 1, maka untuk setiap kolom A_k sebuah elemen 1 terdapat pada baris sumber dan elemen 1 yang lainnya terdapat pada baris tujuan. Selisih dari jumlah vektor-vektor baris sumber dan jumlah vektor-vektor baris tujuan berupa suatu vektor nol. Oleh karena itu vektor - vektor baris dalam A_k tidak independen linear, sehingga ada satu baris dalam A_k yang semua elemen - nya adalah nol. Akibatnya $|A_k| = 0$.
3. Jika ada paling sedikit satu kolom yang hanya memuat sebuah elemen 1 maka akan diperoleh :

$$|A_k| = \pm |A_{k-1}|$$

dengan $|A_{k-1}|$ adalah minor A berordo k-1 yang tertentu. Cara yang sama dapat dikerjakan pada A_{k-1} sehingga diperoleh $|A_{k-1}| = 0$ atau $|A_{k-1}| = \pm |A_{k-2}|$ dan seterusnya. Akhirnya akan sampai ke satu elemen yang nilainya 0, 1 atau -1.

Elemen-elemen A hanya 0 dan 1 sehingga $|A| = 0, +1$ atau -1. Jadi terbukti bahwa setiap minor A bernilai +1, -1 atau 0. Sifat ini disebut sifat unimodular matriks A. ■

Contoh 4.1 :

Carilah minor A dari masalah angkutan dengan 3 sumber dan 5 tujuan seperti tampak dalam tabel berikut :

	D_1	D_2	D_3	D_4	D_5	
O_1	3	2	1	4	2	100
O_2	2	4	1	5	3	100
O_3	1	5	2	5	4	100
	50	70	80	60	40	300

Masalah di atas adalah masalah angkutan setimbang, karena $\sum a_i = \sum b_j = 300$.

Bentuk matriks Ax = b dari soal di atas adalah :

$$\begin{bmatrix} 11111 & 00000 & 00000 \\ 00000 & 11111 & 00000 \\ 00000 & 00000 & 11111 \\ 10000 & 10000 & 10000 \\ 01000 & 01000 & 01000 \\ 00100 & 00100 & 00100 \\ 00010 & 00010 & 00010 \\ 00001 & 00001 & 00001 \end{bmatrix} \begin{bmatrix} x_{11} \\ \vdots \\ x_{15} \\ x_{21} \\ \vdots \\ x_{25} \\ x_{31} \\ \vdots \\ x_{35} \end{bmatrix} = \begin{bmatrix} 100 \\ 100 \\ 100 \\ 50 \\ 70 \\ 80 \\ 60 \\ 40 \end{bmatrix}$$

Kasus 1. Bentuk minor A dari kolom 6,7,8 dan baris 1,2, 3 adalah :

$$\begin{vmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{vmatrix} = 0$$

Kasus 2. Bentuk minor A dari kolom 6,7,11,12 dan baris 2,3,4,5 adalah :

$$\begin{vmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{vmatrix} = (-1)^{1+1} \begin{vmatrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{vmatrix} + (-1)^{1+3} \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{vmatrix}$$

$$= (-1)^{1+3} \begin{vmatrix} 1 & 1 \\ 1 & 0 \end{vmatrix} + (-1)^{1+1} \begin{vmatrix} 1 & 1 \\ 0 & 1 \end{vmatrix} +$$

$$(-1)^{1+3} \begin{vmatrix} 0 & 0 \\ 1 & 1 \end{vmatrix}$$

$$= -1 + 1 + 0 = 0$$

Kasus 3. Bentuk minor A dari kolom 6,12 dan baris 3,4 adalah :

$$\begin{vmatrix} 0 & 1 \\ 1 & 0 \end{vmatrix} = -1$$

Kasus 3. Bentuk minor A dari kolom 6,7,11 dan baris 2, 3,4 adalah :

$$\begin{vmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{vmatrix} = (-1)^{1+1} \begin{vmatrix} 0 & 1 \\ 0 & 1 \end{vmatrix} + (-1)^{1+2} \begin{vmatrix} 0 & 1 \\ 1 & 1 \end{vmatrix}$$

$$= 0 + -(-1) = +1$$

Dari kasus 1, 2 dan 3 terlihat bahwa minor A bernilai 0, 1 dan -1. ∎

D. LINTASAN TERTUTUP METODE STEPPING STONE

Untuk menjelaskan adanya lintasan tertutup metode Stepping Stone kita tulis lagi bentuk matriks A dalam persamaan (4-6).

$$A = \left[\begin{array}{cccc}
11\ldots1 & 00\ldots0 & \ldots & 00\ldots0 \\
00\ldots0 & 11\ldots1 & \ldots & 00\ldots0 \\
\vdots & & & \vdots \\
00\ldots0 & 00\ldots0 & \ldots & 11\ldots1 \\
10\ldots0 & 10\ldots0 & \ldots & 10\ldots0 \\
01\ldots0 & 01\ldots0 & \ldots & 01\ldots0 \\
\vdots & & & \vdots \\
00\ldots1 & 00\ldots1 & \ldots & 00\ldots1
\end{array}\right]$$

Untuk mencari sebuah basis dari ruang kolom sembarang matriks A ,kita gunakan operasi baris elementer pada A agar menjadi matriks eselon E. Kemudian vektor kolom dari A yang berkaitan dengan kolom dari matriks E yang memuat elemen pivot merupakan basis dari ruang kolom A.

Bentuk matriks A yang tersusut Gauss yaitu matriks E kita tulis lagi di bawah ini :

$$E = \left[\begin{array}{cccc}
11\ldots1 & 00\ldots0 & \ldots & 00\ldots0 \\
01\ldots0 & 01\ldots0 & \ldots & 01\ldots0 \\
\vdots & & & \vdots \\
00\ldots1 & 00\ldots1 & \ldots & 00\ldots1 \\
00\ldots0 & 11\ldots1 & \ldots & 00\ldots0 \\
\vdots & & & \vdots \\
00\ldots0 & 00\ldots0 & \ldots & 11\ldots1 \\
00\ldots0 & 00\ldots0 & \ldots & 00\ldots0
\end{array}\right]$$

Jadi bentuk matriks basis E´ adalah :

$$E' = \begin{bmatrix} 11\dots1 & 0 & \dots & 0 \\ 01\dots0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 00\dots1 & 0 & \dots & 0 \\ 00\dots0 & 1 & \dots & 0 \\ 00\dots0 & 01 & \dots & 0 \\ \vdots & & & \vdots \\ 00\dots0 & 0 & \dots & 1 \\ 00\dots0 & 0 & \dots & 0 \end{bmatrix} \quad \text{dan} \quad A' = \begin{bmatrix} 11\dots1 & 0 & \dots & 0 \\ 00\dots0 & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 00\dots0 & 0 & \dots & 1 \\ 10\dots0 & 1 & \dots & 1 \\ 01\dots0 & 0 & \dots & 0 \\ \vdots & & & \vdots \\ 00\dots1 & 0 & \dots & 0 \end{bmatrix}$$

Andaikan ada suatu penyelesaian layak basis untuk $Ax = b$. Misalkan B adalah matriks basis berordo m+n. Matriks B terdiri dari m+n-1 buah P_{ij} dan supaya menjadi matriks bujur sangkar perlu ditambahkan sebuah vektor semu q yang bernilai nol. Dalam penyelesaian layak basis setiap P_{ij} dalam A dapat dinyatakan sebagai kombinasi linear dari m+n-1 vektor basis dalam B. Suatu himpunan vektor-vektor basis untuk masalah angkutan akan merupakan suatu himpunan m+n-1 vektor A yang independen linear.

Berikutnya akan kita tunjukkan bahwa kombinasi linear untuk vektor-vektor yang masuk lintasan tertutup akan mempunyai tanda berselang seling positif negatif.

Kita nyatakan $P^{B}_{\alpha\beta}$ adalah suatu vektor dari suatu himpunan vektor-vektor basis yang bersesuaian dengan perubah basis $x^{B}_{\alpha\beta}$. Subskrip α,β menunjukkan letak elemen satuan dalam vektor-vektor basis. Superskrip B menunjukkan bahwa vektor tersebut adalah vektor basis. Setiap vektor P_{ij} dari A dapat ditulis sebagai kombinasi linear dari vektor basis yaitu :

$$P_{ij} = \sum_{\alpha\beta} Y_{ij}^{\alpha\beta} \; P_{\alpha\beta}^{B} \qquad (4\text{-}10)$$

dengan $\sum_{\alpha\beta}$ berarti penjumlahan untuk dikerjakan pada vektor - vektor basis. Vektor semu q tidak muncul dalam (4-10) karena nilainya nol, sehingga $Y_{ij}^{\alpha\beta} = 0$ untuk semua i,j.

Contoh 4.2 :

Berdasarkan soal contoh 4.1.

Dengan menggunakan eliminasi Gauss matriks A akan menjadi matriks E di bawah ini :

$$E = \begin{bmatrix} 11111 & 00000 & 00000 \\ 01000 & 01000 & 01000 \\ 00100 & 00100 & 00100 \\ 00010 & 00010 & 00010 \\ 00001 & 00001 & 00001 \\ 00000 & 11111 & 00000 \\ 00000 & 00000 & 11111 \\ 00000 & 00000 & 00000 \end{bmatrix}$$

Matriks A dependen linear dan m+n-1 baris independen linear. Rank E = Rank A = 7. Kolom yang tidak memuat elemen pivot diabaikan. Jadi didapat matriks E′, yaitu

$$E' = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Vektor-vektor yang memuat elemen pivot akan Independen linear. Vektor-vektor A yang bersesuaian dengan vektor-

vektor E´ adalah :

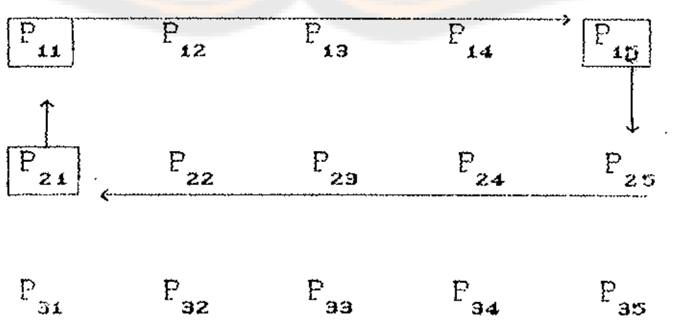
$$A' = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Sehingga di dapat matriks B sebagai berikut :

$$B = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Vektor-vektor kolom yang berasal dari B yang Independen linear adalah $P_{11}, P_{12}, P_{13}, P_{14}, P_{15}, P_{21}$, P_{31}. Setiap P_{ij} dari A dapat dinyatakan sebagai kombinasi linear dari 7 vektor basis dalam B. Vektor-vektor A yang tidak termasuk dalam vektor basis adalah $P_{22}, P_{23}, P_{24}, P_{25}$, P_{32}, P_{33}, P_{34}, P_{35}.

Dengan menggunakan gambar kita cari kombinasi linear dari salah satu vektor basis di atas, misalnya P_{25}.



Kita dapatkan kombinasi linear dari P_{25} adalah :

$$P_{25} = P_{21} - P_{11} + P_{15}$$

Berdasarkan rumus (4-10) kita perlihatkan α, β dan Y_{ij}

$$\begin{array}{c} P_{25} \\ \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \end{array} = \begin{array}{c} \begin{array}{ccc} P_{21} & P_{11} & P_{15} \end{array} \\ \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array} \begin{bmatrix} Y_{25}^{21} \\ Y_{25}^{11} \\ Y_{25}^{15} \end{bmatrix}$$

$i = 2,\ j = 5$

$\alpha = 1, 2\ ,\quad \beta = 1, 5$

Dari perhitungan matriks di atas didapat $Y_{25}^{21} = +1$, $Y_{25}^{11} = -1$, $Y_{25}^{15} = +1$. ■

Teorema 4.5 :

Setiap $Y_{ij}^{\alpha\beta}$ bernilai +1, -1 atau 0.

Bukti :

Rumus (4-10) menyatakan suatu himpunan m+n persamaan dalam m+n-1 variabel, yaitu $Y_{ij}^{\alpha\beta}$. Jika R adalah suatu himpunan yang dibentuk dari m+n-1 kolom independen linear dari A (yaitu dengan mencoret vektor semu q) maka $R\ Y_{ij}^{\alpha\beta} = P_{ij}$. Suatu baris sembarang dari R dapat dicoret dan akan menghasilkan matriks non singular. Kita ambil persamaan ke-i dari (4-10). Komponen ke-i dari P_{ij} memuat sebuah elemen 1. Persamaan - persamaan itu dapat ditulis menjadi $T\ Y_{ij} = e_{j+m-1}$ dengan e_{j+m-1} adalah vektor satuan yang terdiri atas m+n-1 komponen, dan T diperoleh dari R dengan menghapus baris i. Jika sebuah baris yang memuat sebuah elemen 1 dari P_{ij} dicoret, maka yang ada tinggal sebuah vektor satuan. T adalah matriks non singular, maka :

$$Y_{ij} = T^{-1}\ e_{j+m-1} = \gamma_{j+m-1} \qquad (4\text{-}11)$$

dengan γ_{j+m-1} adalah kolom ke (j+m-1) dari T^{-1}. Setiap komponen γ_{j+m-1} adalah minor berordo m+n-2 dari T dibagi dengan $|T|$ dan $|T| \neq 0$. $|T|$ dan minor-minor T merupakan minor A. Oleh karena itu setiap elemen γ_{j+m-1} adalah +1, -1 atau 0. Jadi setiap $Y_{ij}^{\alpha\beta}$ bernilai +1, -1 atau 0. ∎

<u>Contoh 4.3</u> :

Kita tunjukkan setiap $Y_{ij}^{\alpha\beta}$ bernilai +1, -1 atau 0.

Berdasarkan soal contoh 4.1 , kita bentuk R yaitu suatu himpunan yang dibentuk dari m+n-1 kolom independen linear dari A dengan mencoret vektor semu q.

$$R = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Baris ke-2 kita hapus akan didapat matriks T dengan ordo 7.

$$T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Matriks T mempunyai rank 7 dan $|T| = -1$. Jadi matriks T matriks non singular.

Kemudian akan ditunjukkan bahwa elemen-elemen T^{-1} hanya terdiri dari +1, −1 atau 0.

$$T^{-1} = \left[\begin{array}{ccccccc|ccccccc} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

Berdasarkan perhitungan di atas didapat T^{-1} sebagai berikut :

$$T^{-1} = \begin{bmatrix} 1 & 0 & 0 & -1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & -1 & 1 & 1 & -1 & -1 & -1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Elemen-elemen T^{-1} bernilai +1, −1 atau 0. Jadi $Y_{ij}^{\alpha\beta}$ bernilai +1, −1 atau 0. ∎

Dalam masalah angkutan $Y_{ij}^{\alpha\beta}$ selalu bernilai +1, −1 atau 0. Karena keistimewaan $Y_{ij}^{\alpha\beta}$ ini maka P_{ij} dapat ditulis menjadi :

$$P_{ij} = \sum_{\alpha\beta} (\pm)\, P^{B}_{\alpha\beta} \qquad (4\text{-}12)$$

Koefisien dari masing-masing $P^{B}_{\alpha\beta}$ adalah +1 atau -1. Tiap-tiap vektor P_{ij} dan $P^{B}_{\alpha\beta}$ berbentuk seperti (4-8), maka dalam (4-12) pasti ada sebuah $P^{B}_{\alpha\beta}$ dengan bentuk $P^{B}_{iu} = e_i + e_{m+u}$ yang koefisiennya +1 sehinggga akan ada bilangan 1 dalam komponen ke i dari P_{ij}. Untuk $u \neq j$ akan ada sebuah $P^{B}_{\alpha\beta}$ yang berbentuk $P^{B}_{vu} = e_v + e_{m+u}$ yang koefisiennya -1, sehingga elemen 1 dalam komponen ke-(m+u) dari P^{B}_{iu} akan terhapus. Jika cara ini diteruskan maka dalam beberapa langkah yang kurang dari m+n-1, akan diperoleh sebuah $P^{B}_{\alpha\beta}$ yang berbentuk $P^{B}_{wj} = e_w + e_{m+j}$ yang koefisiennya +1. Elemen 1 dalam komponen ke-w akan menghapus komponen ke-w dari vektor yang tepat sebelumnya, yang koefisiennya -1. Komponen ke-(m+j) dari P^{B}_{wj} akan menghapus komponen ke-(m+j) dari P_{ij}. Bentuk P_{ij} dalam vektor basis menjadi sederhana sehingga dapat ditulis :

$$P_{ij} = P^{B}_{iu} - P^{B}_{vu} + P^{B}_{vt} - \ \ldots\ - P^{B}_{ws} + P^{B}_{wj} \qquad (4\text{-}13)$$

Karena dalam (4-13) vektor dalam kombinasi linear kita mulai dan akhiri dengan tanda positif dan karena tanda berselang seling positif negatif maka sejumlah ganjil vektor basis yang tidak nol selalu dapat menyatakan setiap vektor dari A.

Berdasarkan (4-8), rumus (4-13) dapat ditulis menjadi :

$$\begin{aligned} P_{ij} &= P^{B}_{iu} - P^{B}_{vu} + P^{B}_{vt} - \ \ldots\ - P^{B}_{ws} + P^{B}_{wj} \\ &= e_i + e_{m+u} - e_v - e_{m+u} + e_v + e_{m+t} - \ \ldots\ - e_w - e_{m+s} + e_w + e_{m+j} \end{aligned}$$

$$= e_i + e_{m+j}$$

Sebagai contoh kita ambil P_{25} :

$$P_{25} = P_{21} - P_{11} + P_{15}$$

$$= e_2 + e_{m+1} - e_1 - e_{m+1} + e_1 + e_{m+5}$$

$$= e_2 + e_{m+5}$$

Keistimewaan $Y_{ij}^{\alpha\beta}$ juga mempermudah kita dalam menghitung ongkos kesempatan (c'_{ij}). Di bawah ini akan kita perlihatkan cara menghitung c'_{ij}.

Jika $c_{\alpha\beta}^{B}$ adalah ongkos yang bersesuaian dengan vektor basis $P_{\alpha\beta}^{B}$, maka ongkos kesempatan P_{ij} yang bukan basis yang dilambangkan dengan c'_{ij} dapat ditulis menjadi :

$$c'_{ij} = \sum_{\alpha\beta} (\pm)\, c_{\alpha\beta}^{B} - c_{ij} \qquad (4-14)$$

Untuk vektor P_{ij} dari (4-13) dapat ditulis menjadi :

$$c'_{ij} = c_{iu}^{B} - c_{vu}^{B} + c_{vt}^{B} - \ldots - c_{ws}^{B} + c_{vj}^{B} - c_{ij} \qquad (4-15)$$

Semua c'_{ij} dapat dihitung dengan mencari suatu lintasan tertutup yang menghubungkan c_{ij} dengan vektor-vektor basis $c_{\alpha\beta}^{B}$.

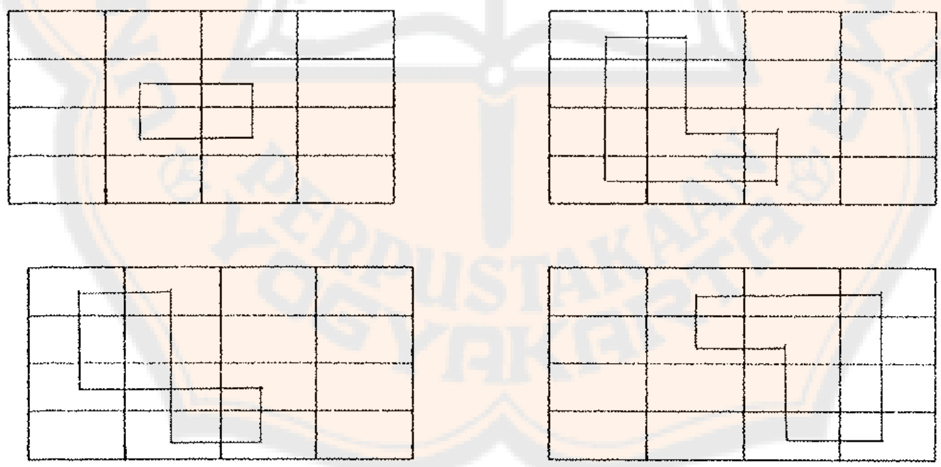
Perhitungan masalah angkutan sebagian besar dikerjakan dalam tabel masalah angkutan. Begitu juga untuk mencari ongkos kesempatan P_{ij} yang bukan basis dihitung dalam tabel dengan cara membuat lintasan tertutup.

Definisi :

Sebuah lintasan tertutup adalah suatu barisan kotak-kotak yang sedemikian hingga :

1. Tiap-tiap pasangan kotak yang berturutan terletak dalam baris yang sama atau kolom yang sama.
2. Tidak ada 3 kotak berturutan yang terletak dalam baris atau kolom yang sama.
3. Kotak pertama dan yang terakhir dari barisan ini terletak dalam baris atau kolom yang sama.
4. Tidak ada kotak yang muncul lebih dari sekali dalam barisan ini.

Contoh macam-macam bentuk lintasan tertutup :



Lintasan tertutup dimulai dari kotak yang bukan basis (kotak kosong) dihubungkan ke kotak-kotak basis. Kotak yang dilalui lintasan diberi tanda berselang-seling positif negatif. Pemberian tanda lintasan dimulai dengan tanda positif.

Penulisan P_{ij} dalam bentuk tabel akan mempermudah kita dalam menentukan suatu kombinasi linear dari vektor - vektor basis bagi P_{ij} yang bukan basis. Kita tidak perlu menghitung $Y_{ij}^{\alpha\beta}$ secara eksplisit. Nilai $Y_{ij}^{\alpha\beta}$ diperoleh dengan cara memberi tanda berselang-seling positif negatif pada kotak yang dilalui lintasan.

Seperti halnya P_{ij} untuk menghitung c'_{ij} kita hitung dengan membuat lintasan tertutup terlebih dahulu. Misalkan K_{ij} adalah kotak kosong dan $K_{ir}, K_{ur}, K_{uq}, K_{pq}, K_{pj}$ adalah kotak-kotak basis. Dari K_{ij} dibuat lintasan tertutup yang menghubungkan K_{ij} dengan kotak-kotak basis, sehingga terjadi lintasan dengan urutan $K_{ij}, K_{ir}, K_{ur}, K_{uq}, K_{pq}, K_{pj}$. Ongkos kesempatan c'_{ij} dapat dihitung dengan cara sebagai berikut :

$$c'_{ij} = c^{B}_{ir} - c^{B}_{ur} + c^{B}_{uq} - c^{B}_{pq} + c^{B}_{pj} - c_{ij} \qquad (4\text{-}16)$$

Untuk menghitung ongkos kesempatan, ongkos kotak basis yang dilalui lintasan diberi tanda berselang-seling positif negatif.

E. BILANGAN BARIS DAN BILANGAN KOLOM MODI

Untuk uji optimum dengan menggunakan metode MODI, kita tentukan dahulu bilangan baris (U_i) dan bilangan kolom (V_j) dengan rumus :

$$U_i + V_j = c_{ij} \qquad (4\text{-}17)$$

sehingga ongkos pada kotak basis di atas dapat kita tulis menjadi :

$$c_{ir} = U_i + V_r$$

$$c_{ur} = U_u + V_r$$

$$c_{uq} = U_u + V_q$$

$$c_{pq} = U_p + V_q$$

$$c_{pj} = U_p + V_j$$

Ongkos kesempatan c'_{ij} dalam (4-16) dapat dihitung dengan cara lebih sederhana, yaitu :

$$\begin{aligned} c'_{ij} &= c^B_{ir} - c^B_{ur} + c^B_{uq} - c^B_{pq} + c^B_{pj} - c_{ij} \\ &= U_i + V_r - U_u - V_r + U_u + V_q - U_p - V_q + U_p + V_j - c_{ij} \\ &= U_i + V_j - c_{ij} \end{aligned}$$

Jadi c'_{ij} dapat dihitung dengan rumus :

$$c'_{ij} = U_i + V_j - c_{ij} \qquad (4\text{-}18)$$

Di bawah ini akan kita perlihatkan bahwa dalam menentukan U_i dan V_j dapat dipilih satu secara sembarang sebagai yang pertama dan diisi dengan nilai sembarang sehingga yang lain akan tertentu dan hal ini tidak mempengaruhi penghitungan c'_{ij} dengan $i = 1,2,\ldots,m$ dan $j = 1,2,\ldots,n$.

Andaikan tabel angkutan dengan kotak isi sebagai berikut : $K_{sp}, K_{sr}, K_{tq}, K_{vq}, K_{vp}$. Ongkos angkut yang bersesuaian dengan kotak tersebut adalah : $c_{sp}, c_{sr}, c_{tq}, c_{vq}, c_{vp}$. Akan kita tunjukkan bahwa dengan mengambil nilai U_i (bilangan baris MODI) secara sembarang akan dapat ditemukan c'_{ij} (ongkos kesempatan) yang sama.

Misalkan kita ambil U_t = a maka akan kita dapatkan :

$$V_q = c_{tq} - U_t = c_{tq} - a$$

$$U_w = c_{wq} - V_t = c_{wq} - c_{tq} + a$$

$$V_p = c_{wp} - U_w = c_{wp} - c_{wq} + c_{tq} - a$$

$$U_s = c_{sp} - V_p = c_{sp} - c_{wp} + c_{wq} - c_{tq} + a$$

$$V_r = c_{sr} - U_s = c_{sr} - c_{sp} + c_{wp} - c_{wq} + c_{tq} - a$$

Dengan mengambil nilai U_t secara sembarang (t juga baris sembarang) kita dapat menentukan bilangan baris dan bilangan kolom yang lain. Sekarang kita tunjukkan ongkos kesempatan kotak kosong misalnya K_{sq}, akan kita dapatkan :

$$c'_{sq} = U_s + V_q - c_{sq}$$

$$= c_{sp} - c_{wp} + c_{wq} - c_{tq} + a + c_{tq} - a - c_{sq}$$

$$= c_{sp} - c_{wp} + c_{wq} - c_{sq}$$

Dari penghitungan c'_{sq} terlihat bahwa untuk menghitung c'_{sq} hanya melibatkan ongkos angkut. Untuk sembarang nilai V_j dikerjakan secara analog. Jadi dengan mengambil nilai sembarang untuk U_i atau V_j tidak akan mempengaruhi penghitungan ongkos kesempatan, karena dalam menghitung ongkos kesempatan hanya melibatkan ongkos angkut saja.

Untuk kotak basis c'_{ij} = 0. Jika ongkos kesempatan c'_{ij} > 0 berarti tabel belum optimum.

B A B V

MASALAH ANGKUTAN YANG DIPERLUAS

Masalah yang dibahas pada bab sebelumnya adalah masalah angkutan baku berpola minimum. Untuk menyelesaikan masalah tersebut sudah disediakan algoritma angkutan. Ada masalah angkutan yang berpola maksimum. Ada masalah yang merupakan kejadian khusus dari masalah angkutan yaitu masalah penugasan. Ada juga masalah yang merupakan bentuk umum masalah angkutan yaitu masalah pemindahan muatan dan masalah penjaluran tangki. Keempat masalah tersebut dapat diselesaikan dengan menggunakan metode angkutan, dengan cara mengembalikan masalah tersebut ke masalah angkutan pola minimum. Meskipun masalah - masalah di atas dapat dikembalikan ke bentuk masalah angkutan pola minimum tetapi untuk mempermudah penyelesaiannya telah disediakan algoritma tersendiri sehingga jalannya penyelesaian menjadi ringkas. Keempat masalah tersebut kita bahas satu persatu.

A. MASALAH ANGKUTAN POLA MAKSIMUM

Tidak selamanya masalah angkutan berpola minimum. Masalah yang kita hadapi dalam kehidupan sehari-hari ada juga yang berpola maksimum. Masalah angkutan pola maksimum dapat dikembalikan ke masalah angkutan pola minimum dengan cara menentukan bilangan yang lebih besar atau sama dengan ongkos terbesar, kemudian dicari selisih antara bilangan terbesar itu dengan ongkos tiap-tiap kotak. Selanjutnya soal dapat dikerjakan dengan menggunakan metode angkutan. Setelah diperoleh penyelesaian optimum maka penyelesaian optimum maksimumnya pun sama hanya nilai fungsi sasarannya yang berbeda.

Secara matematis perumusan masalah angkutan pola maksimum adalah :

mencari $x_{ij} \geq 0$ (5-1)

yang memenuhi kendala :

$$\sum_{j=1}^{n} x_{ij} = a_i \qquad (5\text{-}2)$$

$$\sum_{i=1}^{m} x_{ij} = b_j \qquad (5\text{-}3)$$

dengan $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$

dan memaksimumkan $f = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$ (5-4)

f adalah fungsi sasaran untuk pola maksimum.

Untuk mengembalikan masalah angkutan pola maksimum ke masalah angkutan pola minimum diadakan transformasi :

$$\bar{c}_{ij} = P - c_{ij}$$

dengan P sembarang bilangan yang memenuhi $P \geq c_{ij}$ untuk setiap i dan j.

Masalah angkutan pola minimum yang sesuai adalah :

mencari $x_{ij} \geq 0$ (5-5)

yang memenuhi kendala :

$$\sum_{j=1}^{n} x_{ij} = a_i \qquad (5\text{-}6)$$

$$\sum_{i=1}^{m} x_{ij} = b_j \qquad (5\text{-}7)$$

$$\text{dengan } \sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$$

$$\text{dan meminimumkan } \bar{f} = \sum_{i=1}^{m} \sum_{j=1}^{n} \bar{c}_{ij}\, x_{ij} \qquad (5\text{-}8)$$

$\bar{f}$ adalah fungsi sasaran untuk pola minimum.

Atau $\bar{f}$ dapat dihitung dengan cara :

$$\begin{aligned}
\bar{f} &= \sum_{i=1}^{m}\sum_{j=1}^{n} (P - c_{ij})\, x_{ij} \\
&= \sum_{i=1}^{m} \sum_{j=1}^{n} P\, x_{ij} - \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}\, x_{ij} \\
&= P \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} - \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}\, x_{ij} \\
&= P \sum_{i=1}^{m} a_i - f \qquad (5\text{-}9)
\end{aligned}$$

Jadi bila untuk x_{ij} tertentu $\bar{f}$ mencapai minimum dengan sendirinya x_{ij} tersebut membuat f menjadi maksimum.

Contoh 5.1 :

Suatu perusahaan kain sedang membuat motif suatu kain dengan 2 cara yaitu dicap dan dibatik. Perusahaan ini mempunyai 3 orang pekerja yang masing-masing mempunyai kemampuan yang berlainan dalam menggunakan kedua cara tersebut. Pekerja-pekerja tersebut dalam seminggu dapat bekerja selama 30 jam. Selama seminggu cara dicap dapat digunakan selama 40 jam dan cara dibatik digunakan selama 50 jam. Banyak kain (dalam gulung) yang dapat diselesaikan oleh tiap-tiap pekerja tiap minggu dengan menggunakan kedua cara tersebut adalah :

	D_1	D_2
O_1	4	3
O_2	2	5
O_3	7	6

Tujuan perusahaan tersebut adalah memaksimumkan banyak kain yang dapat dikerjakan oleh pekerja-pekerja itu dengan kedua cara tersebut dalam seminggu.

Untuk menyelesaikan soal contoh 5.1 kita susun dahulu Tabel kendala lengkap dengan ongkosnya seperti terlihat dalam tabel 5.1.

Tabel 5.1

	D_1	D_2	a_i
O_1	(4)	(3)	30
O_2	(2)	(5)	30
O_3	(7)	(6)	30
b_j	40	50	90

Tabel 5.2

	D_1	D_2	a_i
O_1	(4) 10	(5) 20	30
O_2	(6)	(3) 30	30
O_3	(1) 30	(2)	30
b_j	40	50	90

Kita pilih P = 8, sehingga didapat transformasi :

$$\bar{c}_{ij} = 8 - c_{ij}$$

Masalah angkutan menjadi berpola minimum.

Tabel awal disusun dengan metode matriks minimum hasilnya tampak dalam tabel 5.2 dan ternyata tabel sudah optimum dengan :

$$\bar{f} = 4.10 + 5.20 + 3.30 + 1.30$$
$$= 40 + 100 + 90 + 30 = 260$$
$$f = 8.90 - 260 = 720 - 260 = 460$$

Dikembalikan ke soal asli, penyelesaian optimumnya sama dengan penyelesaian optimum soal minimum didapat tabel 5.3.

Tabel 5.3

	D_1	D_2	a_i
O_1	10 (4)	20 (3)	30
O_2	(2)	30 (5)	30
O_3	30 (7)	(6)	30
b_j	40	50	90

$$f = 4.10 + 3.20 + 5.30 + 7.30$$
$$= 40 + 60 + 150 + 210 = 460$$

Jadi kain yang dapat dikerjakan oleh ketiga pekerja tersebut dengan menggunakan cara dicap dan dibatik maksimum sebanyak 460 gulung dalam seminggu. ∎

Dengan dikembalikannya penyelesaian optimum ke soal asli maka nilai f dapat langsung dihitung dengan :

$$f = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} \; x_{ij}$$

Menyelesaikan masalah angkutan dengan memaksimumkan fungsi sasaran berarti harus dipilih ongkos angkutan yang terbesar sehingga ongkos angkut total maksimum. Jika masalah angkutan pola maksimum ini dikembalikan ke masalah angkutan pola minimum maka harus dicari bilangan yang lebih besar atau sama dengan setiap ongkos angkut. Kemudian dicari selisih antara bilangan terbesar dengan setiap ongkos angkut. Masalah angkutan menjadi berpola minimum. Dengan transformasi tersebut berarti bahwa ongkos terbesar dalam masalah angkutan pola maksimum menjadi ongkos terkecil dalam masalah angkutan pola minimum. Begitu juga sebaliknya ongkos terkecil dalam masalah angkutan pola maksimum menjadi ongkos terbesar dalam masalah angkutan pola minimum. Sehingga soal dapat diselesaikan dengan menggunakan metode angkutan. Penyelesaian optimumnya sama hanya fungsi sasaran yang berbeda.

B. MASALAH PENUGASAN

Masalah penugasan merupakan kasus khusus dari masalah angkutan. Kekhususan masalah penugasan dibandingkan dengan masalah angkutan adalah :

1. banyak sumber sama dengan banyak tujuan $(m=n)$
2. $a_i = 1$, $\forall i$, dan $b_j = 1$, $\forall j$
3. $x_{ij} \in \{0,1\}$

Akibatnya dalam satu kolom dan dalam satu baris hanya ada satu kotak yang isi. Dengan kata lain dalam satu baris dan dalam satu kolom terdapat tepat satu alokasi. Bentuk penyelesaian seperti ini disebut penyelesaian layak.

Secara umum masalah penugasan dapat dirumuskan sebagai berikut :

mencari $x_{ij} \in \{0,1\}$ (5-10)

yang memenuhi kendala :

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad (5\text{-}11)$$

$$\sum_{i=1}^{m} x_{ij} = 1 \qquad (5\text{-}12)$$

dan meminimumkan $f = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} \; x_{ij}$ (5-13)

Untuk menyelesaikan masalah penugasan telah disedia - kan 3 cara, yaitu :

1. Metode angkutan
2. Metode Enumerasi
3. Metode Hungarian

Berikut ini akan dijelaskan tentang cara kerja tiap metode di atas.

1. Metode Angkutan

Metode angkutan yang digunakan untuk menyelesaikan masalah penugasan disini seperti yang telah dibahas dalam BAB III.

Contoh 5.2 :

Ada 3 pekerjaan (O_1, O_2, O_3) yang harus diselesaikan / diproses oleh 3 mesin (D_1, D_2, D_3). Biaya untuk memproses pekerjaan / tugas tersebut (dalam ribuan rupiah) dapat dilihat dari tabel berikut ini.

	D_1	D_2	D_3
O_1	20	27	30
O_2	10	18	16
O_3	14	16	12

Setiap tugas / pekerjaan hanya dilakukan oleh satu mesin, dengan kata lain setiap mesin hanya memproses 1 tugas saja. Bagaimana cara menyelesaikan pekerjaan tersebut sehingga biaya minimum?.

Contoh 5.2 di atas dikerjakan dengan metode angkutan. Disusun tabel kendala lengkap dengan ongkosnya, didapat Tabel 5.4. Kemudian tabel awal disusun dengan menggunakan metode sudut barat laut, diperoleh Tabel 5.5.

Tabel 5.4

	D_1	D_2	D_3	a_i
O_1	20	27	30	1
O_2	10	18	16	1
O_3	14	16	12	1
b_j	1	1	1	

Tabel awal masalah penugasan selalu merosot karena jumlah kotak isi ada 3 kurang dari m+n-1 = 5. Maka harus diberi bilangan ε sebanyak 2 buah supaya tabel tidak merosot. Penempatan ε dibuat sedemikian rupa sehingga rantai penyusunan U_i dan V_j terjamin. Kemudian kita uji apakah tabel sudah optimum atau belum dengan menggunakan metode MODI, didapat Tabel 5.6.

Tabel 5.5

	D_1	D_2	D_3	a_i
O_1	20 1	27	30	1
O_2	10	18 1	16	1
O_3	14	16	12 1	1
b_j	1	1	1	

Tabel 5.6

	D_1	D_2	D_3	a_i	U_i
O_1	20 1 (−)	27 (+) ε	30	1	0
O_2	$\boxed{1}$ 10 (+)	18 (−) 1	16 ε	1	-9
O_3	14	16	12 1	1	-13
b_j	1	1	1		
V_j	20	27	25		

Setelah kita uji dengan metode MODI ternyata ada alokasi yang harus dipindahkan, di dapat Tabel 5.7.

Tabel 5.7

	D_1	D_2	D_3	a_i	U_i
O_1	20	27 1	30	1	9
O_2	10 1	18 ε	16 ε	1	0
O_3	14	16	12 1	1	-4
b_j	1	1	1		
V_j	10	18	16		

Kemudian kita uji lagi keoptimuman Tabel 5.7 dengan menggunakan metode MODI. Dalam tabel 5.7 semua $c'_{ij} \leq 0$. Karena ongkos kesempatan tidak ada yang positif berarti tabel sudah optimum.

Fungsi sasaran yang sesuai dengan penyelesaian di atas adalah :

$$f = 27 + 10 + 12$$
$$= 49$$

Jadi penyelesaian optimumnya adalah :

O_1 ditempatkan pada mesin D_2

O_2 ditempatkan pada mesin D_1

O_3 ditempatkan pada mesin D_3

dengan ongkos total penugasan adalah R_p 49.000,00. ∎

2. Metode Enumerasi

Cara kerja metode Enumerasi adalah dicari semua kemungkinan penyelesaian layak yang dapat terjadi, kemudian dihitung ongkos totalnya. Akan ada penyelesaian layak sebanyak m ! dengan m adalah banyak sumber atau tujuan. Setelah semua ongkos total diperoleh, pilih ongkos total yang paling kecil. Pola layak dengan ongkos total paling kecil ini merupakan penyelesaian optimumnya.

<u>Contoh 5.3 :</u>

Contoh 5.2 diselesaikan dengan menggunakan metode Enumerasi.

Kemungkinan pola layak yang terjadi ada 6, yaitu :

	D_1	D_2	D_3
O_1	1		
O_2		1	
O_3			1

	D_1	D_2	D_3
O_1	1		
O_2			1
O_3		1	

	D_1	D_2	D_3
O_1		1	
O_2	1		
O_3			1

	D_1	D_2	D_3
O_1		1	
O_2			1
O_3	1		

	D_1	D_2	D_3
O_1			1
O_2		1	
O_3	1		

	D_1	D_2	D_3
O_1			1
O_2	1		
O_3		1	

Tabel 5.8

Penugasan	Ongkos Total
O_1D_1, O_2D_2, O_3D_3	20 + 18 + 12 = 50
O_1D_1, O_2D_3, O_3D_2	20 + 16 + 16 = 52
O_1D_2, O_2D_2, O_3D_3	27 + 10 + 12 = 49
O_1D_2, O_2D_3, O_3D_2	27 + 16 + 14 = 57
O_1D_3, O_2D_2, O_3D_3	30 + 18 + 14 = 62
O_1D_3, O_2D_3, O_3D_2	30 + 10 + 16 = 56

Ongkos total terendah = 49

Jadi penyelesaian optimumnya adalah O_1D_2, O_2D_2, O_3D_3. ∎

3. Metode Hungarian

Sebelum kita membicarakan langkah kerja metode Hungarian kita buktikan dahulu theorema berikut :

Theorema 5.1 :

Penyelesaian optimum untuk masalah angkutan tidak akan berubah jika elemen-elemen dalam sembarang baris atau kolom dari matriks ongkos dikurangi atau ditambah dengan suatu jumlah yang sama.

Bukti :

Andaikan x_{ij} adalah variabel basis penyelesaian optimum soal asli dan f adalah ongkos total soal asli.

$$f = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} \quad x_{ij} \; , \qquad c_{ij} \geq 0$$

Andaikan r_i adalah suatu jumlah yang ditambahkan / dikurangkan pada baris i dan t_j adalah suatu jumlah yang ditambahkan / dikurangkan pada kolom j dalam tabel ongkos.

Dengan demikian dalam tabel ongkos yang ditransformasi elemen ongkos pada baris i, kolom j, yaitu c_{ij} menjadi:

$$c_{ij} + r_i + t_j$$

Ongkos total menjadi :

$$f' = \sum_{i=1}^{m} \sum_{j=1}^{n} (c_{ij} + r_i + t_j) x_{ij}$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{i=1}^{m} r_i \sum_{j=1}^{n} x_{ij} + \sum_{j=1}^{n} t_j \sum_{i=1}^{m} x_{ij} \quad (**)$$

Dari kendala penawaran diketahui $\sum_{j=1}^{n} x_{ij} = a_i$ dan dari kendala permintaan diketahui $\sum_{i=1}^{m} x_{ij} = b_j$ maka (**) dapat ditulis :

$$f' = f + \sum_{i=1}^{m} r_i a_i + \sum_{j=1}^{n} t_j b_j$$

$$f' = f + \text{konstan}$$

Terbukti bahwa penambahan suatu konstanta pada baris atau kolom dari tabel ongkos tidak mungkin mengubah penyelesaian optimum. ∎

Karena kekhususan masalah penugasan maka disediakan metode khusus yang disebut metode Hungarian. Metode Hungarian dikerjakan pada suatu tabel. Tabel ini hanya memuat ongkos saja. Supaya lebih ringkas tabel ongkos sementara dipisahkan dari tabel alokasi. Keterangan a_i dan b_j tidak diperlukan lagi.

Langkah-langkah metode Hungarian :

1. Baris-baris dan bila perlu kolom-kolom disusutkan sedemikian hingga setiap baris dan setiap kolom memuat unsur nol. Disusutkan berarti tiap baris dan tiap kolom kita kurangi dengan unsur terkecil dari tiap baris dan

tiap kolomnya.

2. Uji keoptimuman dengan cara membuat garis tegak atau mendatar untuk menutup semua nol dengan banyak garis seminim mungkin. Jika banyak garis sama dengan n (=orde matriks) maka tabel sudah memuat pola optimum. Jika banyak garis masih kurang dari n, tabel belum memuat penyelesaian optimum. Bila tabel belum optimum masuk ke langkah 3. Bila tabel sudah optimum langsung ke langkah 4.
3. Tinjau unsur-unsur yang sama sekali belum tertutup garis, lihat unsur yang terkecil didalam kelompok tersebut. Susutkan setiap baris (atau kolom) yang memuat mereka. Akan timbul unsur negatif. Karena c_{ij} tidak boleh negatif maka untuk menghapusnya, baris (atau kolom) yang memuat c_{ij} negatif ditambah dengan unsur terkecil dari tiap kolom (atau baris). Kemudian kembali ke langkah 2, kalau tabel sudah optimum langsung ke langkah 4.
4. Memilih kotak-kotak nol yang menyusun pola optimum. Caranya :
   a. Pilih kotak nol yang sendirian dalam barisnya atau kolomnya.
   b. Coret baris dan kolom kotak yang terpilih itu.
   c. Ulangi cara a dan b terhadap sisa kotak-kotak yang belum tercoret (yang belum jenuh).
   d. Jika dalam tabel ternyata tidak ada nol yang

sendirian berarti ada pilihan penyelesaian optimum.

Contoh 5.4 :

Selesaikan contoh 5.2 dengan metode Hungarian.

Langkah 1. Tiap baris dan tiap kolom disusutkan (dikurangi dengan unsur terkecil dalam tiap baris dan kolomnya) agar memuat unsur nol.

Tabel 5.9

	D_1	D_2	D_3
O_1	20	27	30
O_2	10	18	16
O_3	14	16	12
	-10	-16	-12

Tabel 5.10

	D_1	D_2	D_3	
O_1	10	11	18	-10
O_2	0	2	4	
O_3	4	0	0	

Tabel 5.11

	D_1	D_2	D_3
O_1	0	1	8
O_2	0	2	4
O_3	4	0	0

Langkah 2. Uji optimum dengan cara membuat garis tegak atau mendatar untuk menutup semua nol dengan banyak garis seminim mungkin.

Banyak garis penutup nol ada 2 kurang dari n = 3 berarti tabel belum optimum.

Tabel 5.12

	D_1	D_2	D_3
O_1	0	1	8
O_2	0	2	4
O_3	4	0	0

Langkah 3. Unsur-unsur yang sama sekali belum tertutup garis penutup nol disusutkan agar memuat unsur nol.

Tabel 5.13

	D_1	D_2	D_3
O_1	0	1	8
O_2	0	2	4
O_3	4	0	0

Tabel 5.14

	D_1	D_2	D_3
O_1	-1	0	7
O_2	-2	0	2
O_3	4	0	0
	+2		

Terdapat unsur negatif dalam tabel 5.14, untuk menghapusnya, kolom-1 ditambah dengan 2 didapat tabel 5.15.

Tabel 5.15

	D_1	D_2	D_3
O_1	1	0	7
O_2	0	0	2
O_3	6	0	0

Tabel 5.15 kita uji keoptimumannya, ternyata banyak garis penutup nol ada 3 sama dengan n berarti tabel sudah optimum.

Langkah 4. Memilih kotak-kotak nol yang menyusun pola optimum

Tabel 5.16

	D_1	D_2	D_3
O_1	1	$\boxed{0}$	7
O_2	$\boxed{0}$	0	2
O_3	6	0	$\boxed{0}$

Tabel 5.17

	D_1	D_2	D_3
O_1		27	
O_2	10		
O_3			12

Kotak-kotak nol yang menyusun pola optimum adalah O_1D_2, O_2D_1, O_3D_3, dengan ongkos total minimum adalah :

$$f = 27 + 10 + 12$$
$$= 49$$

Jadi dapat diambil kesimpulan bahwa agar biaya penugasan minimum maka :

Pekerjaan 1 harus diproses dengan mesin 2

Pekerjaan 2 harus diproses dengan mesin 1

Pekerjaan 3 harus diproses dengan mesin 3 ∎

Penyelesaian masalah penugasan dengan menggunakan metode angkutan dalam contoh 5.2, pekerjaan dianggap sebagai sumber dan mesin dianggap sebagai tujuan. Masalah penugasan yang diselesaikan dengan metode angkutan akan selalu merosot karena jumlah kotak isi akan selalu kurang dari m+n-1, sehingga kita harus menyisipkan bilangan ε agar tabel tidak merosot.

Penyelesaian masalah penugasan dengan cara enumerasi tidak efisien karena memakan banyak waktu untuk menyelesaikannya. Kita harus menentukan semua kemungkinan penyelesaian layak yang terjadi, kemudian dihitung ongkos total masing-masing kemungkinan tersebut. Setelah semua ongkos total diperoleh kita pilih ongkos total yang terkecil. Penyelesaian dengan ongkos total terkecil inilah yang merupakan penyelesaian optimum.

Meskipun masalah penugasan dapat dikembalikan ke masalah angkutan sehingga dapat diselesaikan dengan menggunakan metode angkutan, tetapi ada cara yang lebih mudah untuk menyelesaikan masalah penugasan. Metode yang paling mudah untuk menyelesaikan masalah penugasan adalah metode Hungarian.

C. MASALAH PEMINDAHAN MUATAN

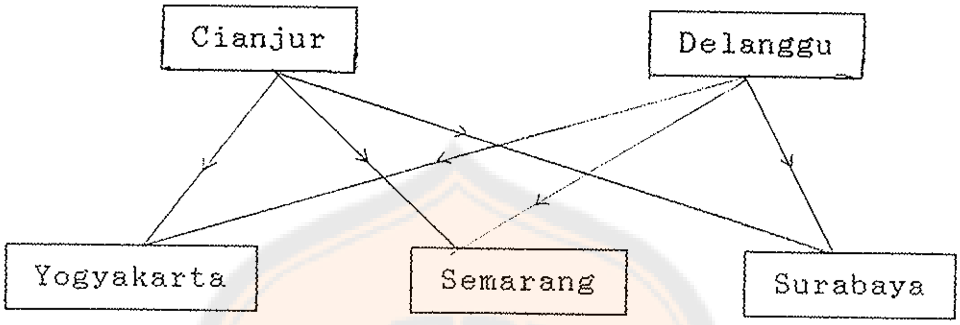
Dalam masalah angkutan kita nyatakan bahwa kegiatan sumber hanya sebagai pengirim barang dan kegiatan tujuan hanya sebagai penerima barang. Sedangkan dalam masalah pemindahan muatan, sumber dan tujuan dapat menjadi pengirim dan sekaligus penerima barang. Dalam masalah pemindahan muatan ini kita tidak hanya memperhatikan sambungan langsung yang menghubungkan sumber ke tujuan tetapi kita juga harus mempertimbangkan semua kemungkinan sambungan.

Untuk mengetahui gambaran masalah pemindahan muatan, ikuti contoh masalah berikut. Diketahui lima lokasi di

Pulau Jawa yang masuk dalam jaringan pendistribusian beras yaitu Cianjur, Delanggu, Yogyakarta, Semarang dan Surabaya. Cianjur dan Delanggu dalam memproduksi beras mempunyai kelebihan beras masing-masing sebesar 30 ton dan 25 ton, sedangkan Yogyakarta, Semarang dan Surabaya mempunyai kekurangan beras masing-masing sebesar 20 ton, 15 ton dan 20 ton. Supaya kelima lokasi tersebut dapat mencukupi kebutuhan masing-masing maka perlu adanya pembagian beras atau pemindahan beras dari lokasi yang kelebihan beras ke lokasi yang kekurangan beras. Ada banyak jalan yang menghubungkan lokasi-lokasi tersebut dengan ongkos angkut yang berbeda-beda. Oleh karena itu kita harus memperhatikan semua jalur pengiriman baik yang langsung maupun yang tidak langsung. Pengiriman langsung misalnya dari Delanggu langsung ke Semarang dan pengiriman tidak langsung misalnya dari Delanggu melewati Yogyakarta baru kemudian menuju ke Semarang. Ongkos angkut ke masing-masing lokasi tersebut diketahui. Masalah yang dihadapi adalah jalan mana yang harus dilewati agar ongkos angkut menjadi minimum.

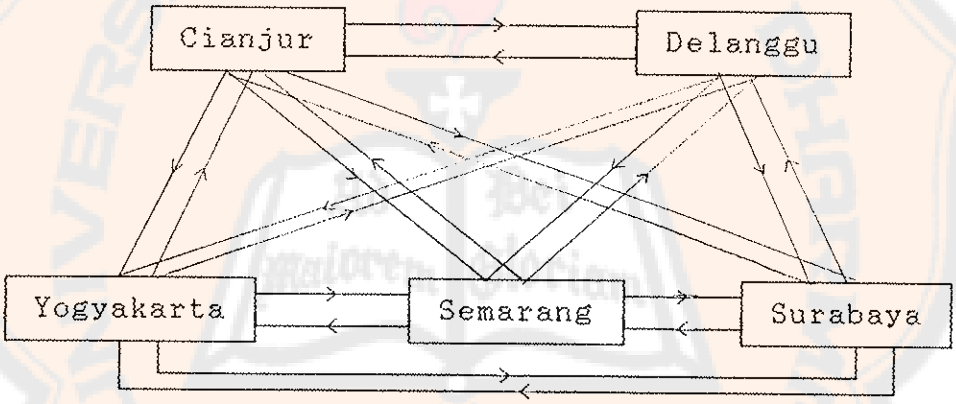
Contoh masalah di atas dapat kita gambarkan seperti terlihat dalam Gambar 5-1 dan Gambar 5-2.

Jika digambarkan dengan simbol, tanda + menunjukkan sumber dan tanda - menunjukkan tujuan, maka gambar di atas menjadi seperti terlihat dalam Gambar 5-3 dan Gambar 5-4.



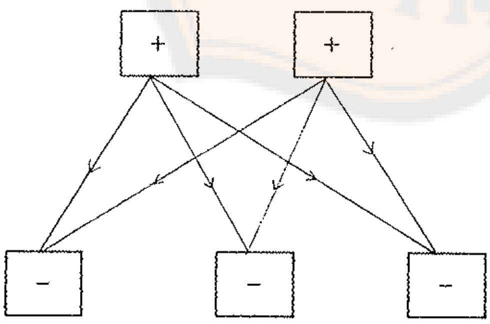
Gambar 5-1

pengiriman langsung

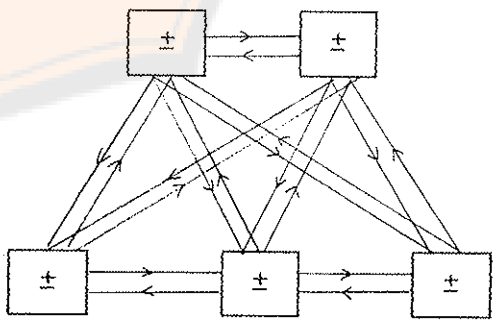


Gambar 5-2

pengiriman tidak langsung



Gambar 5-3
Pengiriman langsung



Gambar 5-4
Pengiriman tidak langsung

Secara umum masalah pemindahan muatan melibatkan sejumlah m sumber asli $(O_1, O_2, \ldots, O_m)$ dan n tujuan asli $(D_1, D_2, \ldots, D_n)$. Tiap-tiap sumber asli menghasilkan a_i unit suatu barang dan tiap-tiap tujuan asli membutuhkan b_j unit suatu barang. Diasumsikan bahwa banyaknya permintaan tujuan asli harus sama dengan banyaknya penawaran sumber asli. Barang dapat dikirim secara langsung (yaitu dari sumber asli ke tujuan asli) maupun secara tidak langsung (yaitu dari sumber asli melalui sumber dan/atau tujuan lain baru kemudian sampai ke tujuan akhir). Ongkos angkut c_{ij} untuk mengangkut satu satuan barang dari sumber i ke tujuan j diketahui. Ongkos $c_{ij} = c_{ji}$ dan ongkos $c_{ii} = 0$. Ongkos angkut c_{ii} adalah ongkos angkut ke dirinya sendiri (dari sumber i ke tujuan i itu sendiri). Akibatnya dari adanya pengiriman tidak langsung maka tujuan asli dapat menjadi sumber semu dan sumber asli dapat menjadi tujuan semu. Sumber semu ini dianggap mempunyai $a_i = 0$ dan tujuan semu dianggap mempunyai $b_j = 0$. Sumber semu kita beri nomor dari m+1,m+2,...,m+n sehingga banyaknya sumber dalam masalah pemindahan muatan ada m+n buah. Tujuan semu kita beri nomor dari 1,2,...,m dan tujuan asli kita ganti menjadi nomor m+1,m+2,...,m+n sehingga banyaknya tujuan dalam masalah pemindahan muatan ada m+n buah. Masalah yang dihadapi adalah bagaimana mengatur pola angkutan agar ongkos angkut total menjadi minimum.

Dari gambaran masalah pemindahan muatan secara umum kita dapatkan perumusan masalah pemindahan muatan sebagai berikut :

Jika diketahui bahwa :

x_{ij} = jumlah total barang yang dikirim dari sumber i ke tujuan j

t_i = banyak barang yang dipindahkan ke sumber i

t_j = banyak barang yang dipindahkan dari tujuan j

c_i = ongkos bongkar muat dan penyimpanan barang

$\sum_{i=1}^{m+n} a_i = \sum_{j=1}^{m+n} b_j$ besar permintaan sama dengan besar penawaran, maka :

Jumlah total barang yang keluar dari sumber harus sama dengan besar penawaran sumber i ditambah banyak barang yang dipindahkan ke sumber tersebut.

$$\sum_{j=1}^{m+n}{}' \; x_{ij} = a_i + t_i \;, \qquad i = 1,2,\ldots,m \qquad (5\text{-}14)$$

Jumlah total barang yang keluar dari tujuan (sumber semu) sama dengan banyak barang yang dipindahkan ke tujuan tersebut.

$$\sum_{j=1}^{m+n}{}' \; x_{ij} = t_i \;, \qquad i = m+1,m+2,\ldots,m+n \qquad (5\text{-}15)$$

Jumlah total barang yang masuk ke sumber (tujuan semu) sama dengan banyak barang yang dipindahkan dari sumber tersebut.

$$\sum_{i=1}^{m+n}{}' \; x_{ij} = t_j \;, \quad j = 1,2,\ldots,n \qquad (5\text{-}16)$$

Jumlah total barang yang masuk ke tujuan sama dengan besar permintaan tujuan ditambah banyak barang yang dipindahkan dari tujuan tersebut :

$$\sum_{i=1}^{m+n}{}' x_{ij} = b_j + t_j \ , \qquad j = m+1, m+2, \ldots, m+n \qquad (5\text{-}17)$$

Fungsi sasaran masalah pemindahan muatan adalah meminimumkan :

$$f = \sum_{j=1}^{m+n}{}' \sum_{i=1}^{m+n}{}' c_{ij}\, x_{ij} + \sum_{i=1}^{m+n} c_i\, t_i \ , \qquad (5\text{-}18)$$

Ambil batas atas pengiriman untuk nilai t_o, sehingga :

$$t_i \leq t_o \ , \qquad i = 1, 2, \ldots, m+n \qquad (5\text{-}19)$$

Sehingga didapat :

$$t_i = t_o - x_{ii} \ , \qquad i = 1, 2, \ldots, m+n \qquad (5\text{-}20)$$

dengan x adalah variabel semu tak negatif. Pengiriman tidak langsung akan mengirimkan barang dari sumber dan/atau tujuan lain agar sampai ke tujuan akhir. Barang yang diangkut dari sumber sebelum sampai ke tujuan akhir akan melalui sumber dan/atau tujuan lain. Barang yang sampai ke sumber dan/atau tujuan lain ini hanya dititipkan. Nilai t_o menyatakan jumlah total barang yang dipindahkan. Nilai t_o kita ambil :

$$t_o = \sum_{i=1}^{m+n} a_i \qquad (5\text{-}21)$$

Nilai t_o diambil bilangan yang cukup besar agar menjamin semua x_{ii} masuk penyelesaian optimum basis. x_{ii} harus masuk penyelesaian agar penyelesaian menjadi penyelesaian layak basis. Penyelesaian masalah pemindahan muatan akan menjadi layak basis apabila terdapat $x_{ij} > 0$ sebanyak $2(m+n)-1$.

Masalah pemindahan muatan dapat dijadikan bentuk masalah angkutan dengan cara mensubstitusikan (5-20) ke (5-14) sampai (5-18), diperoleh :

$$\sum_{j=1}^{m+n} x_{ij} = \begin{cases} a_i + t_o & , i = 1,2,\ldots,m \\ t_o & , i = m+1, m+2, \ldots, m+n \end{cases}$$

$$\sum_{i=1}^{m+n} x_{ij} = \begin{cases} t_o & , j = 1,2,\ldots,m \\ b_j + t_o & , j = m+1, m+2, \ldots, m+n \end{cases} \qquad (5\text{-}22)$$

Fungsi sasaran menjadi meminimumkan :

$$f = \sum_{j=1}^{m+n} \sum_{i=1}^{m+n} c_{ij}\, x_{ij} + \text{konstan} \qquad (5\text{-}23)$$

Masalah pemindahan muatan dapat diselesaikan dengan menggunakan metode angkutan. Bentuk tabel kendala lengkap dengan ongkosnya masalah pemindahan muatan dapat dilihat pada Tabel 5.18.

Penyelesaian layak basis awal masalah pemindahan muatan terdiri dari penyelesaian masalah angkutan ditambah nilai x_{ii} sepanjang diagonal tabel.

Tabel 5.18

Tabel Kendala Lengkap dengan Ongkosnya

	O_1	O_2	...	O_m	D_1	D_2	...	D_n	a_i
O_1									$a_1 + t_o$
O_2									$a_2 + t_o$
.									.
O_m									$a_m + t_o$
D_1									t_o
D_2									t_o
.									.
D_n									t_o
b_j	t_o	t_o	...	t_o	b_1+t_o	b_2+t_o	...	b_n+t_o	

Contoh 5.5 :

Suatu perusahaan logistik beras memiliki tujuh gudang beras diseluruh Indonesia. Perusahaan ini merencanakan untuk menjual beras tersebut. Ada tiga gudang beras (O_1, O_2, O_3) yang memiliki kelebihan persediaan beras masing-masing sebesar 26 ton, 10 ton, 4 ton. Sedangkan empat gudang lainnya (D_1, D_2, D_3, D_4) kekurangan beras

	O_1	O_2	O_3	D_1	D_2	D_3	D_4
O_1	0	2	1	3	2	3	1
O_2	2	0	3	4	2	1	2
O_3	1	3	0	1	5	3	2
D_1	3	4	1	0	4	1	3
D_2	2	2	5	4	0	2	1
D_3	3	1	3	1	2	0	2
D_4	1	2	2	3	1	2	0

sebesar 15 ton, 10 ton, 5 ton dan 10 ton. Biaya angkut untuk tiap jalur tertera pada tabel ongkos di atas. Bagaimanakah cara pengiriman beras tersebut agar ongkos pemindahan muatan menjadi minimum ?.

Masalah pemindahan muatan di atas diselesaikan dengan menggunakan metode angkutan. Langkah pertama dibuat tabel kendala lengkap dengan ongkosnya masalah pemindahan muatan, dengan mengambil $t_o = \sum a_i = 40$ dan $c_{ii} = 0$ didapat tabel 5.19.

Tabel 5.19

	O_1	O_2	O_3	D_1	D_2	D_3	D_4	a_i
O_1	0	2	1	3	2	3	1	66
O_2	2	0	3	4	2	1	2	50
O_3	1	3	0	1	5	3	2	44
D_1	3	3	1	0	4	1	3	40
D_2	2	2	5	4	0	2	1	40
D_3	3	1	3	1	2	0	2	40
D_4	1	2	2	3	1	2	0	40
b_j	40	40	40	55	50	45	50	320

Penyelesaian layak basis awal terdiri dari penyelesaian optimum pemindahan langsung dari sumber ke tujuan (masalah angkutan) ditambah dengan nilai 40 sepanjang diagonal. Terlebih dahulu dicari penyelesaian optimum masalah pemindahan langsung, didapat Tabel 5.20.

Tabel 5.20

	D_1	D_2	D_3	D_4	a_i	U_i
O_1	11^{3}	5^{2}	^{3}	10^{1}	26	0
O_2	^{4}	5^{2}	5^{1}	^{2}	10	0
O_3	4^{1}	^{5}	^{3}	^{2}	4	-2
b_j	15	10	5	10	40	
V_j	3	2	1	1		

Tabel 5.20 diisi dengan menggunakan metode matriks minimum. Urutan pengisian $K_{14}, K_{23}, K_{31}, K_{22}, K_{12}, K_{11}$. Jumlah kotak isi ada 6 sama dengan m+n-1. Jadi tabel tidak merosot. Kemudian tabel tersebut diuji keoptimumannya dengan menggunakan metode MODI. Kebetulan pada tabel 5.20 semua $c'_{ij} \leq 0$. Jadi tabel sudah optimum.

Tabel awal dari masalah pemindahan muatan menjadi tabel 5.21.

Tabel 5.21

	O_1	O_2	O_3	D_1	D_2	D_3	D_4	a_i	U_i
O_1	40^{0}	^{2}	$\boxed{1}\ ^{1}$	11^{3}	5^{2}	^{3}	10^{1}	66	0
O_2	^{2}	40^{0}	^{3}	^{4}	5^{2}	5^{1}	^{2}	50	0
O_3	^{1}	^{3}	40^{0}	4^{1}	^{5}	^{3}	^{2}	44	-2
D_1	^{3}	^{4}	^{1}	40^{0}	^{4}	^{3}	^{1}	40	-3
D_2	^{2}	^{2}	^{5}	^{4}	40^{0}	^{2}	^{1}	40	-2
D_3	^{3}	^{1}	^{3}	$\boxed{1}\ ^{1}$	^{2}	40^{0}	^{2}	40	-1
D_4	^{1}	^{2}	^{2}	^{3}	$\boxed{0}\ ^{1}$	^{2}	40^{0}	40	-1
b_j	40	40	40	55	50	45	50	320	
V_j	0	0	2	3	2	1	1		

Kotak isi ada 13 sama dengan 2(m+n)-1, jadi tabel tidak merosot. Tabel 5.21 diuji keoptimumannya dengan menggunakan metode MODI. Tabel 5.21 belum optimum karena masih ada nilai $c'_{ij} > 0$. Nilai $c'_{13} = c'_{64} = 1$, kita pilih secara sembarang kotak yang masuk basis, diperoleh tabel 5.22.

Tabel 5.22

	O_1	O_2	O_3	D_1	D_2	D_3	D_4	a_i	U_i
O_1	40 ^{0}	^{2}	11 ^{1}	^{3}	5 ^{2}	^{3}	10 ^{1}	66	0
O_2	^{2}	40 ^{0}	^{3}	^{4}	5 ^{2}	5 ^{1}	^{2}	50	0
O_3	^{1}	^{3}	29 ^{0}	15 ^{1}	^{5}	^{3}	^{2}	44	-1
D_1	^{3}	^{4}	^{1}	40 ^{0}	^{4}	^{3}	^{1}	40	-2
D_2	^{2}	^{2}	^{5}	^{4}	40 ^{0}	^{2}	^{1}	40	-2
D_3	^{3}	^{1}	^{3}	$\boxed{0}$ ^{1}	^{2}	40 ^{0}	^{2}	40	-1
D_4	^{1}	^{2}	^{2}	^{3}	$\boxed{0}$ ^{1}	^{2}	40 ^{0}	40	-1
b_j	40	40	40	55	50	45	50	320	
V_j	0	0	1	2	2	1	1		

Kita uji keoptimuman tabel 5.22 dengan menggunakan metode MODI. Didapat bahwa semua $c'_{ij} \leq 0$. Jadi tabel 5.22 sudah optimum, dengan ongkos pemindahan muatan sebesar :

$$f = 1.11 + 1.15 + 2.5 + 2.5 + 1.5 + 1.10$$

$$= 11 + 15 + 10 + 10 + 5 + 10 = 61$$

Kita lihat dalam tabel 5.22, O_3 memindahkan muatan sebesar 11 ton sedangkan lokasi yang lain tidak memindahkan apa-apa. Selisih antara nilai t_o dengan nilai yang berada dalam kotak sepanjang diagonal utama menyatakan jumlah masing-masing barang yang dipindahkan. Jadi cara pengiriman beras agar ongkos angkutan minimum dengan cara sebagai berikut : O_1 mengirimkan 11 ton ke O_3, 5 ton ke D_2 dan 10 ton ke D_4. O_2 mengirimkan 5 ton ke D_2 dan 5 ton lagi ke D_3. D_1,D_2,D_3 dan D_4 tidak mengirimkan apa-apa.

Tabel 5.23

	O_1	O_2	O_3	D_1	D_2	D_3	D_4	a_i
O_1	4 0 ^{0}	^{2}	11 ^{1}	^{3}	^{2}	^{3}	15 ^{1}	66
O_2	^{2}	40 ^{0}	^{3}	^{4}	5 ^{2}	^{1}	^{2}	50
O_3	^{1}	^{3}	29 ^{0}	15 ^{1}	^{5}	^{3}	^{2}	44
D_1	^{3}	^{4}	^{1}	40 ^{0}	^{4}	^{3}	^{1}	40
D_2	^{2}	^{2}	^{5}	^{4}	40 ^{0}	^{2}	^{1}	40
D_3	^{3}	^{1}	^{3}	^{1}	^{2}	40 ^{0}	^{2}	40
D_4	^{1}	^{2}	^{2}	^{3}	5 ^{1}	^{2}	35 ^{0}	40
b_j	4 0	40	40	55	50	45	50	320

Dalam tabel 5.22, $c'_{64} = c'_{75} = 0$ berarti penyelesaian optimum tidak tunggal ada pilihan penyelesaian optimum. Penyelesaian optimum yang mungkin dapat terjadi, salah satunya adalah tabel 5.23, dengan nilai program :

$$f = 1.11 + 1.15 + 2.5 + 1.5 + 1.15 + 1.5$$

$$= 11 + 15 + 10 + 5 + 15 + 5$$

$$= 61 \blacksquare$$

D. MASALAH PENJALURAN KAPAL TANGKI

Masalah penjaluran tangki merupakan perluasan dari masalah angkutan. Gambaran masalah penjaluran tangki adalah sebagai berikut :

Masalah penjaluran tangki mempunyai sejumlah sumber i = 1,2,...,m yang merupakan tempat asal suatu tangki mengi - rimkan muatannya. Muatan suatu tangki akan diserahkan ke suatu tujuan j = 1,2,...,n. Besarnya penawaran dari i dan besarnya permintaan dari j diketahui. Waktu keberangkatan tangki dari sumber i ke tujuan j dan lamanya waktu perjalanan yang harus ditempuh untuk sampai ke tujuan diketahui. Dengan diketahuinya tanggal keberangkatan tangki dan lamanya waktu perjalanan tangki itu berjalan agar sampai ke tujuan maka tanggal kedatangan tangki ke tujuan dapat diketahui. Tangki tidak dapat mengirimkan sebagian muatan ke suatu lokasi dan sebagian muatan yang lain ke lokasi yang lain. Masalah yang dihadapi adalah bagaimana merencanakan jadwal pengiriman agar jumlah tangki yang digunakan untuk mengangkut minimum.

Contoh nyata gambaran masalah penjaluran tangki di atas adalah contoh 5.6.

Contoh 5.6 :

Diketahui ada dua sumber (O_1,O_2) dan tiga tujuan (D_1,D_2,D_3). Ada pengiriman suatu tangki yang memuat suatu barang dari sumber i ke tujuan j. Jadwal keberangkatan pengiriman dari sumber i ke tujuan j diperlihatkan dalam tabel 5.25.

Tabel 5.25

	D_1	D_2	D_3
O_1	1,4,7,10	4,9	6,8
O_2	3,6,9	5,7,9	5,11

Angka dalam kotak (1,1) pada tabel 5.25 berarti pada tanggal 1,4,7,10 ada satu tangki yang berangkat dari O_1 untuk mengirimkan suatu muatan ke D_1. Dimungkinkan bahwa ada lebih dari satu tangki yang mengirimkan muatannya dari sumber i ke tujuan j pada tanggal yang sama dalam satu kotak (i,j). Misalnya ada tiga tangki yang akan berangkat dari O_2 ke D_1 pada tanggal 6, maka dalam kotak (2,1) ditulis 3,6,6,6,9.

Lamanya (waktu) perjalanan yang harus di tempuh untuk mengirimkan muatan dari sumber i ke tujuan j diketahui dalam tabel 5.26.

Tabel 5.26

	D_1	D_2	D_3
O_1	1	2	3
O_2	3	1	2

Angka 2 pada kotak (1,2) dalam tabel 5.26 berarti lamanya perjalanan dari O_1 ke D_2 adalah 2 hari. Waktu yang digunakan suatu tangki untuk mengirimkan muatan sama dengan waktu yang digunakan suatu tangki untuk kembali, misalnya dalam kotak (1,2) juga berarti tangki dari D_2 kembali ke O_1 selama 2 hari.

Dengan diketahuinya tabel 5.25 dan 5.26, kita peroleh tanggal kedatangan suatu tangki yang diperlihatkan dalam tabel 5.27.

Tabel 5.27

	D_1	D_2	D_3
O_1	2,5,8,11	6,11	9,11
O_2	6,9,12	6,8,10	7,13

Angka 2 dalam kotak (1,1) pada tabel 5.27 berarti bahwa tangki yang membawa muatan dari O_1 akan sampai ke D_1 pada tanggal 2. Angka 2 diperoleh dari menjumlahkan 1 dalam kotak (1,1) pada tabel 5.25 dengan angka 1 dalam kotak (1,1) pada tabel 5.26. Masalah yang dihadapi adalah bagaimana mengatur jadwal pengiriman agar jumlah tangki yang digunakan untuk mengangkut muatan minimum.

1. Perumusan Masalah

Jika kita lambangkan bahwa :

$n_{\alpha i}$: banyak tangki yang berangkat dari sumber i pada tanggal α.

$N_{\beta j}$: banyak tangki yang datang ke tujuan j pada tanggal β

$x_{\alpha i \beta j}$: banyak tangki yang kembali dari tujuan j pada tanggal β dan mengambil muatan dari sumber i pada tanggal α.

maka perumusan umum masalah penjaluran tangki adalah :

mencari $x_{\alpha i \beta j} \geq 0$ (5-24)

yang memenuhi kendala :

$\sum_{\alpha i} x_{\alpha i \beta j} \leq N_{\beta j}$ untuk setiap βj dengan $N_{\beta j} \neq 0$

(5-25)

$\sum_{\beta j} x_{\alpha i \beta j} \leq n_{\alpha i}$ untuk setiap αi dengan $n_{\alpha i} \neq 0$

(5-26)

dengan fungsi sasaran meminimumkan :

$$f = \sum_{\beta j} N_{\beta j} \tag{5-27}$$

Perumusan (5,24), (5-25) dan (5-26) merupakan suatu masalah angkutan. Oleh karena itu masalah penjaluran tangki dapat diselesaikan dengan metode angkutan. Dalam masalah angkutan $n_{\alpha i}$ menyatakan a_i dan $N_{\beta j}$ menyatakan b_j. Agar masalah di atas menjadi masalah angkutan setimbang maka diperlukan variabel semu $S_{\beta j}$ dan $\sigma_{\alpha i}$.

Masalah angkutan setimbang berarti :

$$\sum_{\beta j} N_{\beta j} = \sum_{\alpha i} n_{\alpha i} \tag{5-28}$$

Kita nyatakan sumber semu dengan $S_{\beta j}$, sehingga (5-25) menjadi persamaan :

$$\sum_{\alpha i} x_{\alpha i \beta j} + S_{\beta j} = N_{\beta j} \tag{5-29}$$

(5-29) dijumlahkan menurut βj didapatkan :

$$\sum_{\beta j} \sum_{\alpha i} x_{\alpha i \beta j} + \sum_{\beta j} S_{\beta j} = \sum_{\beta j} N_{\beta j} \tag{5-30}$$

Kita nyatakan tujuan semu dengan $\sigma_{\alpha i}$ sehingga (5-26) menjadi persamaan :

$$\sum_{\beta j} x_{\alpha i \beta j} + \sigma_{\alpha i} = n_{\alpha i} \tag{5-31}$$

(5-31) dijumlahkan atas αi didapatkan :

$$\sum_{\alpha i} \sum_{\beta j} x_{\alpha i \beta j} + \sum_{\alpha i} \sigma_{\alpha i} = \sum_{\alpha i} n_{\alpha i} \tag{5-32}$$

Jika kita nyatakan $y = \sum_{\alpha i} \sum_{\beta j} x_{\alpha i \beta j}$ maka (5-30) dan (5-32) menjadi :

$$y + \sum_{\beta j} S_{\beta j} = \sum_{\beta j} N_{\beta j} \qquad (5\text{-}33)$$

$$y + \sum_{\alpha i} \sigma_{\alpha i} = \sum_{\alpha i} n_{\alpha i} \qquad (5\text{-}34)$$

Variabel semu $S_{\beta j}$ menyatakan jumlah tangki yang berakhir di j pada tanggal β dan variabel semu $\sigma_{\alpha i}$ menyatakan jumlah tangki yang berasal dari i pada tanggal α. Dari (5-33) dan (5-34) kita dapatkan :

$$\sum_{\alpha i} \sigma_{\alpha i} = \sum_{\beta j} S_{\beta j} = \text{jumlah total tangki}$$

Fungsi sasaran yang semula meminimumkan variabel semu akan sama dengan memaksimumkan y. Fungsi sasaran menjadi :

$$\text{memaksimumkan} \quad y = \sum_{\alpha i} \sum_{\beta j} x_{\alpha i \beta j}$$

Walaupun masalah penjaluran tangki dapat dikembalikan ke masalah angkutan tetapi masalah penjaluran tangki tidak diselesaikan dengan metode angkutan melainkan dengan metode tersendiri yang disebut metode penjaluran. Berikut ini akan dijelaskan metode penjaluran lewat contoh.

Contoh 5.7 :

Mencari $n_{\alpha i}$ dan $N_{\beta j}$ serta rute perjalanan tangki contoh 5.6. Berdasarkan Tabel 5.25 kita akan mencari $n_{\alpha i}$ yaitu banyak tangki yang berangkat dari i pada tanggal α. Tabel yang menunjukkan $n_{\alpha i}$ ini diperlihatkan dalam tabel 5.28.

Tabel 5.28

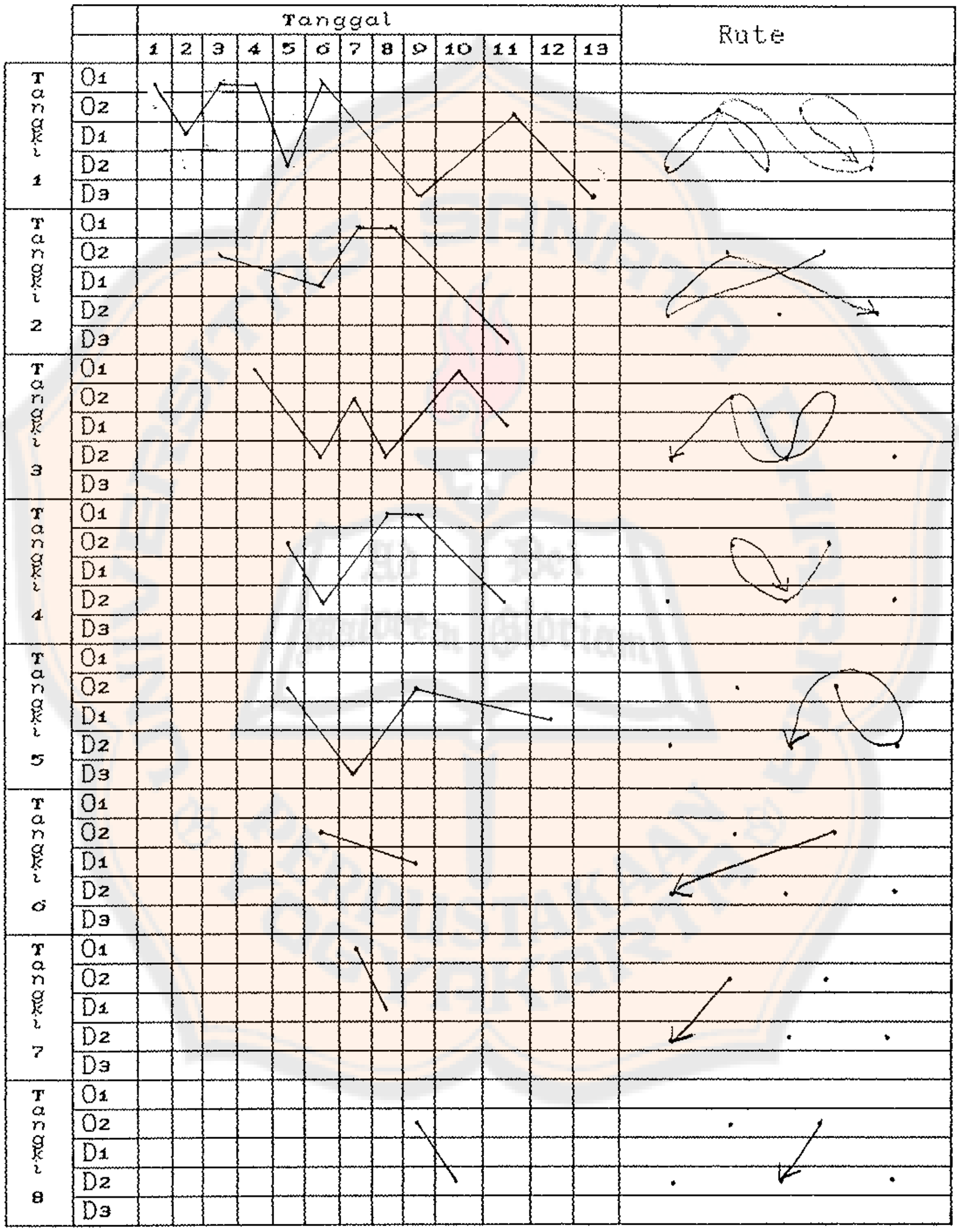
α	i	$n_{\alpha i}$
1	1	1
3	2	1
4	1	2
5	2	2
6	1	1
6	2	1
7	1	1
7	2	1
8	1	1
9	1	1
9	2	2
10	1	1
11	2	1

Tabel 5.29

β	j	$N_{\beta j}$
2	1	1
5	1	1
6	1	1
6	2	2
7	3	1
8	1	1
8	2	1
9	1	1
9	3	1
10	2	1
11	1	1
11	2	1
11	3	1
12	1	1
13	2	1

Berdasarkan tabel 5.27 kita cari $N_{\beta j}$ yaitu banyak tangki yang datang ke j pada tanggal β. Tabel yang menunjukkan $N_{\beta j}$ diperlihatkan dalam tabel 5.29. Pada tabel 5.25 angka 1 (baris 1) hanya muncul satu kali, ini berarti hanya ada satu tangki yang mengirimkan muatan dari O_1 pada tanggal 1, sehingga kita tulis n_{11} = 1 dalam tabel 5.28. Angka 5 pada baris 2 muncul dua kali, ini berarti pada tanggal 5 ada dua tangki yang mengirimkan muatan dari sumber 2, sehingga kita tulis n_{52} = 2. Pada tabel 5.27 angka 2 di kolom 1 muncul satu kali, ini berarti pada tanggal 2 ada satu tangki yang datang ke tujuan 1, sehingga kita tulis N_{21} = 1 dalam Tabel 5.29. Dari Tabel 5.25 dan Tabel 5.26 kita dapat menggambar rute perjalanan tangki seperti diperlihatkan dalam Gambar 5.5.

Gambar 5.5



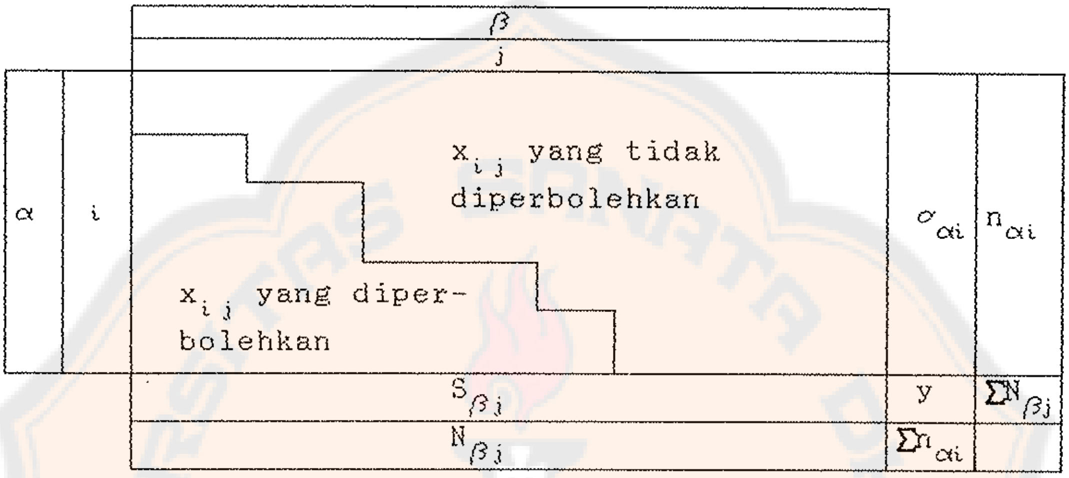
Dari Gambar 5.5 ternyata ada 8 tangki yang digunakan untuk mengirimkan muatan. Tangki 1 berangkat dari O_1

pada tanggal 1 mengirimkan muatan ke D_1, datang ke D_1 pada tanggal 2. Tangki kembali lagi ke O_1 pada tanggal 3, karena pada tanggal 3 tidak ada pengiriman dari O_1 maka tangki beristirahat dan baru ada pengiriman lagi pada tanggal 4 dari O_1 ke D_2 sampai di D_2 pada tanggal 5. Tangki kembali lagi ke O_1 untuk mengambil muatan, selanjutnya dikirimkan ke D_3, sampai di D_3 pada tanggal 9. Kemudian tangki mengambil muatan di O_2 dan mengirimkan muatan ke D_3, sampai di D_3 pada tanggal 13. Tangki 1 berakhir pada tanggal 13 karena setelah tanggal 13 tidak ada pengiriman lagi. Cara yang sama dilakukan terhadap tangki 2, tangki 3, sampai semua tangki yang tersedia digunakan untuk memenuhi semua permintaan. Dari gambar 5.5 ada 8 tangki yang digunakan untuk pengiriman.

Untuk menyelesaikan masalah penjaluran tangki diperlukan suatu tabel seperti yang diperlihatkan dalam Tabel 5.30. Baris untuk sumber semu dan kolom untuk tujuan semu terdiri dari $S_{\beta j}$ dan $\sigma_{\alpha i}$. Kita tempatkan $n_{\alpha i}$ dalam lajur vertikal dan $N_{\beta j}$ dalam lajur horisontal. Tanggal keberangkatan α diletakkan pada kolom paling kiri dan tanggal kedatangan β diletakkan pada baris paling atas. Sumber i diletakkan sesudah tanggal keberangkatan dan tujuan j diletakkan dibawah baris tanggal kedatangan. Garis demarkasi membatasi daerah x_{ij} yang diperbolehkan dan daerah

x_{ij} yang tidak diperbolehkan.

Tabel 5.30



Contoh 5.8 :

Penyelesaian masalah penjaluran tangki contoh 5.7 kita kerjakan dalam tabel masalah penjaluran tangki. Kita dapatkan tabel 5.31.

Tabel 5.31

		2	5	6	6	7	8	8	9	9	10	10	11	11	12	13		
		1	1	1	2	3	1	2	1	3	2	1	2	3	1	3		
1	1																1	1
3	2																1	1
4	1	1															1	2
5	2																2	2
6	1		1															1
6	2																1	1
7	1																1	1
7	2				1													1
8	1			1														1
9	1				1													1
9	2					1											1	2
10	1							1										1
11	2									1								1
							1		1		1	1	1	1	1	1	8	16
		1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	16	

Tangki 1 mulai berangkat dari sumber 1 dengan membawa muatan pada tanggal 1, ini berarti $\sigma_{11} = 1$, kita letakkan angka 1 dalam kotak σ_{11} pada tabel masalah penjaluran tangki. Tangki 1 ini datang ke D_1 pada tanggal 2 dan kembali ke O_1 pada tanggal 3 dan beristirahat selama 1 hari dan siap akan mengirimkan muatan lagi pada tanggal 4. Dalam tabel 5.31 kita letakkan angka 1 pada β = 2 dan j=1 dan pada baris α = 4 dan i = 1 (x_{4121} = 1). Teruskan cara ini sampai tangki 1 berhenti. Ulangi cara di atas untuk semua tangki yang lain sehingga kita dapatkan :

$$\sum_{\alpha i} \sigma_{\alpha i} = \sum_{\beta j} S_{\beta j} = 8$$

Tabel 5.32

		2	5	6	6	7	8	8	9	9	10	10	11	11	12	13		
		1	1	1	2	3	1	2	1	3	2	1	2	3	1	3		
1	1																1	1
3	2																1	1
4	1	1															1	2
5	2																2	2
6	1		1														0	1
6	2																1	1
7	1																1	1
7	2		0		1													1
8	1		0	1														1
9	1		0		1													1
9	2		0			1											1	2
10	1		0					1										1
11	2		0							1								1
							1		1		1	1	1	1	1	1	8	16
		1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	16	

Suatu penyelesaian optimum didapatkan setelah beberapa perhitungan perhitungan yang diperlihatkan dalam tabel 5.32, tabel 5.33 dan tabel 5.34. Variabel y menjadi

10 berarti tangki yang digunakan dalam penjaluran ada 8 tangki. Dari sini terlihat bahwa tangki yang digunakan lebih sedikit.

Tabel 5.33

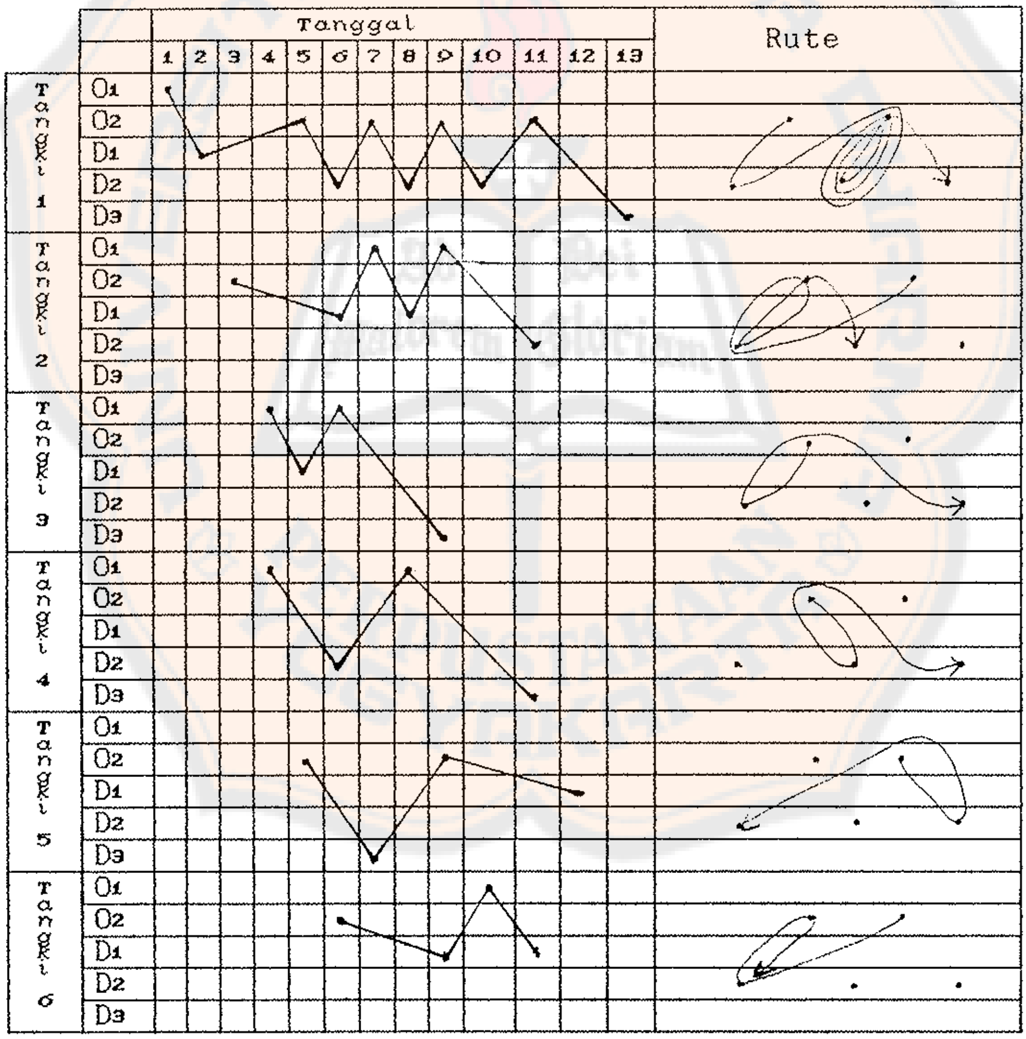
		2	5	6	6	7	8	8	9	9	10	10	11	11	12	13		
		1	1	1	2	3	1	2	1	3	2	1	2	3	1	3		
1	1																1	1
3	2																1	1
4	1																2	2
5	2	1															1	2
6	1																1	1
6	2																1	1
7	1			1													0	1
7	2				1													1
8	1				1													1
9	1						1											1
9	2		1					1										2
10	1					1											0	1
11	2									1								1
									1		1	1	1	1	1	1	9	16
		1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	16	

Tabel 5.34

		2	5	6	6	7	8	8	9	9	10	10	11	11	12	13		
		1	1	1	2	3	1	2	1	3	2	1	2	3	1	3		
1	1																1	1
3	2																1	1
4	1																2	2
5	2	1															1	2
6	1		1															1
6	2																1	1
7	1			1														1
7	2				1													1
8	1				1													1
9	1						1											1
9	2					1		1										2
10	1								1									1
11	2										1							1
										1		1	1	1	1	1	10	16
		1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	16	

Persoalan sekarang yaitu bagaimana menterjemahkan penyelesaian ini ke bentuk gerakan tangki. Kita gunakan cara yang sama pada waktu kita membentuk penyelesaian awal. Kita cari $\sigma_{\alpha i}$ yang tidak nol misalnya σ_{11} berarti ada tangki yang berangkat dari O_1 pada tanggal 1 dan pergi ke D_1. Lama perjalanan dari O_1 ke D_1 adalah 1 hari berarti tangki ini sampai di D_1 pada tanggal 2. Kita lihat kolom 2,1 ke bawah (β = 2, j = 1). Kita cari $x_{\alpha,i,2,1}$ yang tidak nol. Kita lihat dari tabel 5.34 bahwa $x_{5,2,2,1}$ = 1, tangki kembali ke O_2 untuk mengambil muatan pada tanggal 5. Dari Tabel 5.25 dan Tabel 5.26 kita lihat bahwa tangki dari O_2 pada tanggal 5 mengirimkan muatan ke D_2 dengan lama perjalanan 1 hari. Jadi tangki sampai di D_2 pada tanggal 6. Kita lihat kolom 6,2 (β = 6, j = 2) ke bawah, kita cari $x_{\alpha,i,6,2}$ yang tidak nol. Ada dua kemungkinan yaitu pada baris 7,2. Dari O_2 pada tanggal 7 tangki mengirimkan muatannya ke D_2 sampai di D_2 tanggal 8. Cari lagi $x_{\alpha,i,8,2}$ yang tidak nol, ditemukan $x_{9,2,8,2}$ = 1. Pengiriman diteruskan lagi pada tanggal 9 dari O_2 ada dua kemungkinan yaitu ke D_1 dan D_2 misal kita pilih ke D_2 berarti tangki sampai di D_2 tanggal 10 dicari lagi $x_{\alpha,i,10,2}$ yang tidak nol, ditemukan $x_{11,2,10,2}$ = 1. Pada tanggal 11 dari O_2 ada pengiriman ke D_3 pada tanggal 13. Setelah tanggal 13 tidak ada pengiriman lagi berarti rute berhenti sampai di sini. Ulangi prosedur ini untuk semua $\sigma_{\alpha i}$ yang tidak nol, kita dapatkan gerakan tangki seperti yang diperlihatkan dalam gambar 5.6. Rute gerakan tangki tersebut tidak tunggal. Kita dapat menemukan rute lain yang tidak mempengaruhi jumlah total tangki. Penyelesaian optimum disajikan dalam bentuk gambar yaitu Gambar 5.6.

Gambar 5.6



BAB VI
KOMPUTASI MASALAH ANGKUTAN DENGAN KOMPUTER

Dalam bab sebelumnya, kita telah membahas tentang gambaran masalah angkutan, algoritma angkutan dan penyelesaian masalah angkutan. Untuk menyelesaikan masalah angkutan dengan banyak sumber dan tujuan sedikit kita bisa menyelesaikannya dengan cara manual, tetapi untuk menyelesaikan masalah angkutan berformat besar dengan cara manual akan memakan waktu yang tidak sedikit. Oleh karena itu diciptakan program komputer yang dapat membantu kita dalam melakukan komputasi. Dalam bab VI ini kita akan membahas komputasi masalah angkutan dengan bantuan komputer.

Bahasa komputer yang akan digunakan untuk membuat program masalah angkutan dalam bab VI ini adalah Turbo BASIC.

A. LANGKAH - LANGKAH PENYUSUNAN PROGRAM

Tahap pertama dalam menyiapkan sebuah program komputer ialah menentukan sifat dan masalah yang akan diselesaikan dengan komputer, serta memeriksa input dan output yang ada. Tahap kedua adalah mengembangkan cara penyelesaian yang meliputi langkah - langkah pengolahan yang harus dilakukan, sehingga bentuk input akan dikeluarkan sebagai output sesuai yang diharapkan.

Secara rinci langkah-langkah penyusunan program adalah :

1. Pemahaman masalah yang akan kita programkan, terdiri dari :
   a. Input yang dimiliki program
   b. Proses yang harus dilakukan
   c. Output yang dihasilkan
2. Merencanakan langkah proses (flowcharting)
3. Membuat program
4. Pengujian program

B. PEMAHAMAN MASALAH YANG AKAN DIPROGRAMKAN

Dalam kehidupan sehari-hari syarat utama agar kita dapat menyelesaikan persoalan dengan baik ialah harus memahami masalah yang sedang dihadapi, setelah kita memahami, barulah kita memikirkan bagaimana cara menyelesaikan persoalan tersebut.

Masalah yang akan kita programkan dengan Turbo BASIC yaitu masalah angkutan baik yang setimbang maupun yang tidak setimbang. Sasaran utama masalah angkutan yang diprogramkan adalah meminimumkan dan memaksimumkan ongkos angkut total.

Input/masukan yang diperlukan agar komputasi dengan komputer ini dapat dijalankan adalah :

- banyak sumber
- banyak tujuan
- besar penawaran dari masing-masing sumber

- besarnya permintaan dari masing-masing tujuan
- ongkos angkut dari tiap sumber ke tiap tujuan
- bentuk fungsi sasaran dan jenis ekstremnya

Urutan pengerjaan komputasi masalah angkutan dengan komputer mengikuti 3 langkah pengerjaan masalah angkutan, yaitu :

1. Mencari penyelesaian layak basis awal yang tidak merosot.
2. Menguji keoptimuman penyelesaian layak basis.
3. Memperbaiki penyelesaian bila belum optimum.

Hasil komputasi dengan komputer yang kita inginkan adalah menemukan penyelesaian layak basis yang optimum dan besar ongkos angkut total yang diperoleh.

C. MERENCANAKAN LANGKAH PROSES

Sebelum menyusun program terlebih dahulu yang harus dilakukan ialah menyusun instruksi dengan menggunakan simbol-simbol. Bagan dengan simbol-simbol ini disebut flowchart. Flowchart untuk masing-masing subrutin dalam program ini selengkapnya dapat dilihat dalam lampiran. Berikut ini kita uraikan jalannya flowchart.

Masalah yang kita programkan yaitu masalah angkutan setimbang dan masalah angkutan tidak setimbang. Sebelum kita menyelesaikan masalah yang akan kita hadapi perlu kita tanyakan terlebih dahulu hal-hal yang harus diketahui oleh komputer, yaitu banyak sumber, banyak tujuan, besar penawaran dari tiap sumber, besar permintaan dari tiap tujuan, ongkos angkut dari tiap-tiap sumber ke tiap-tiap tujuan dan jenis ekstrem fungsi sasarannya. Untuk menanyakan hal-hal tersebut kita membuat *subrutin masukan*.

Untuk masalah angkutan yang tidak setimbang perlu disetimbangkan terlebih dahulu dengan menggunakan *subrutin setimbangkan*. Untuk masalah angkutan yang memaksimumkan fungsi sasaran diubah ke masalah angkutan yang meminimumkan fungsi sasaran dengan *subrutin maksimum*.

Langkah pertama

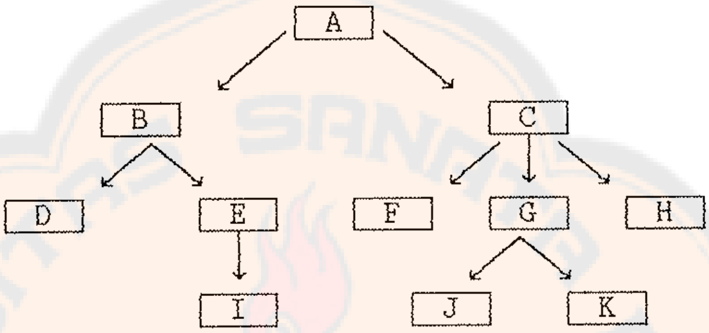
Setelah soal masalah angkutan diketahui oleh komputer, kita mulai menyelesaikan masalah tersebut. Langkah pertama yang harus kita lakukan adalah menentukan penyelesaian layak basis awal yang tidak merosot. Untuk menentukan penyelesaian awal masalah angkutan kita mempunyai enam metode, tetapi dalam pembuatan program ini kita ambil salah satu metode yang paling mudah diterjemahkan oleh komputer yaitu metode sudut barat laut. Penyelesaian awal yang merosot tidak dapat dikerjakan dengan program ini. Dalam program untuk menentukan penyelesaian awal ini kita tulis dalam *subrutin awal*.

Setelah kita dapatkan penyelesaian awal kita periksa apakah penyelesaian awal yang didapat sudah layak atau belum. Jika dalam penyelesaian awal jumlah $x_{ij} \neq 0$ kurang m+n-1 maka penyelesaian awal disebut merosot. Kita tidak dapat menyelesaikan masalah ini sehingga komputer akan memberi pesan kepada kita untuk mengganti soal lain. Jika dalam penyelesaian awal jumlah $x_{ij} \neq 0$ sama dengan m+n-1 maka penyelesaian awal tidak merosot. Proses kita lanjutkan untuk mencari fungsi sasaran yang didapat. Penghitungan ini kita tulis dalam *subrutin sasaran*.

Langkah Kedua

Untuk menguji keoptimuman penyelesaian layak basis kita gunakan metode MODI. Untuk mencari bilangan baris dan bilangan kolom dengan MODI didasarkan pada pencarian $x_{ij} \neq 0$ dalam baris atau kolom yang sama. Urutan letak $x_{ij} \neq 0$ ini berbentuk suatu pohon. Sebuah pohon merupakan himpunan titik yang mempunyai hubungan ibarat orang tua. Semua titik dalam pohon ini mempunyai orang tua yang tunggal yang disebut titik akar. Titik akar ini menjadi titik awal suatu pohon. Untuk lebih jelas kita lihat bagan pohon pada gambar 6.1.

Berdasarkan gambar 6.1, yang menjadi titik akar pohon adalah A. A mempunyai dua anak yaitu B dan C. B mempunyai 2 anak yaitu D dan E, sedangkan C mempunyai 3 anak yaitu F, G dan H. D tidak mempunyai anak, sedangkan E mempunyai 1 anak yaitu I. F dan H tidak mempunyai anak, sedangkan G mempunyai 2 anak yaitu J dan K. Dari pohon tersebut terlihat bahwa tiap orang tua dapat mempunyai lebih dari satu anak tetapi tiap anak hanya mempunyai 1 orang tua.



Gb. 6.1 Pohon

Contoh 6.1 :

Kita mencari letak $x_{ij} \neq 0$ dalam baris atau kolom yang sama masalah di bawah ini.

Diketahui masalah angkutan dengan 3 sumber dan 3 tujuan dengan penyelesaian layak basis tabel berikut ini :

40^{4}	^{5}	$\boxed{1}$ ^{6}	40	0
10^{3}	30^{4}	$\boxed{4}$ ^{2}	40	-1
^{6}	20^{1}	20^{3}	40	-4
50	50	20		
4	5	7		

Misalkan sebagai titik awal kita ambil kotak yang mempunyai ongkos kesempatan maksimum. Dalam tabel di atas kotak yang mempunyai ongkos kesempatan maksimum adalah titik (2,3). Titik (2,3) mempunyai 2 anak yaitu (2,1) dan (2,2). Titik (2,1) mempunyai 1 anak yaitu (1,1) dan titik (2,2) juga mempunyai 1 anak yaitu (3,2). Titik (1,1) tidak mempunyai anak tetapi titik (3,2) masih mempunyai anak yaitu titik (3,3). Titik (3,3) merupakan titik terakhir karena titik (3,3) tidak mempunyai anak. Oleh karena itu diperoleh suatu pohon seperti gambar 6.2.

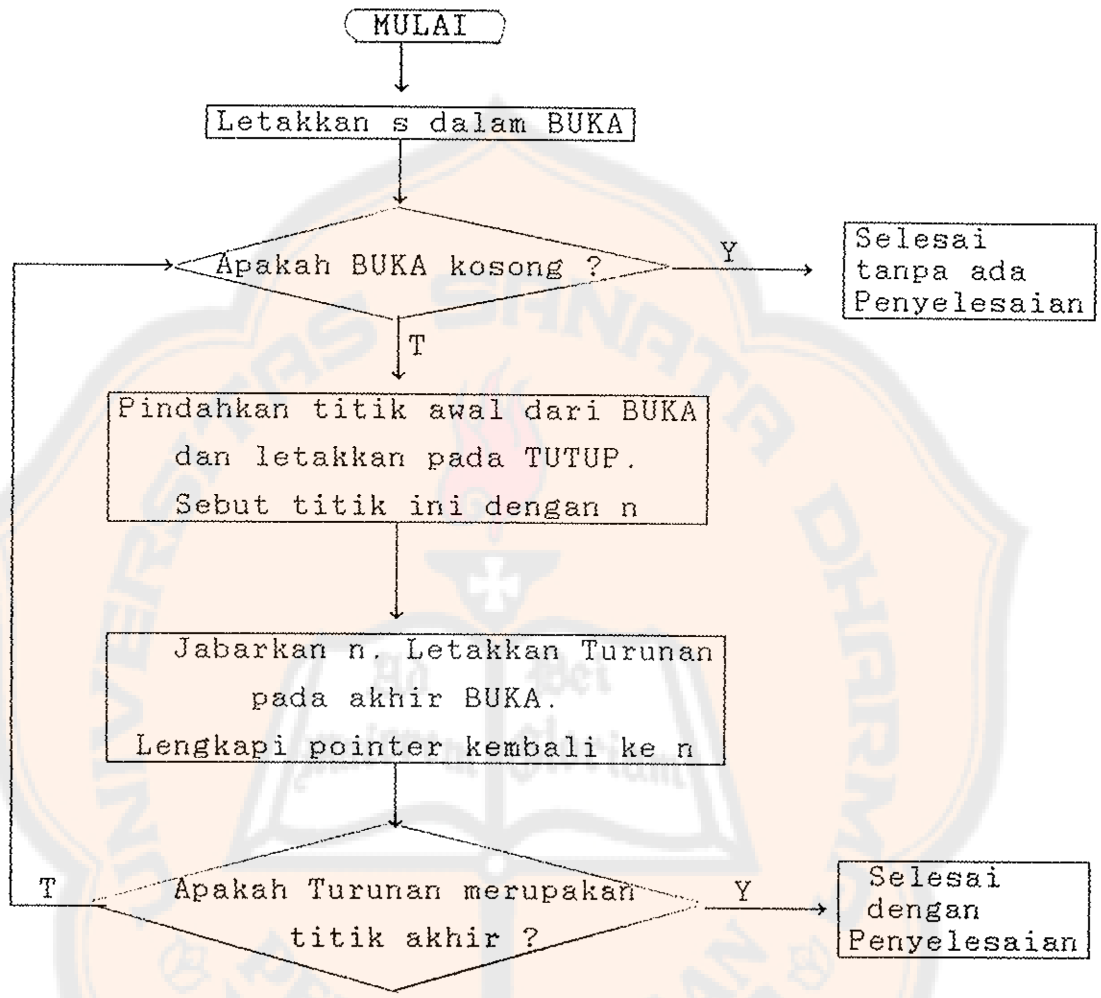
$$\begin{tikzcd}
 & (2,3) \arrow[dl] \arrow[dr] & \\
(2,1) \arrow[d] & & (2,2) \arrow[d] \\
(1,1) & & (3,2) \arrow[d] \\
 & & (3,3)
\end{tikzcd}$$

Gb. 6.2 Pohon urutan letak $x_{ij} \neq 0$

Dalam pembuatan program dengan menggunakan bahasa BASIC kita tidak dapat membuat suatu pohon, tetapi kita dapat menelusuri letak urutan titik-titik dalam pohon tersebut dengan menggunakan **metode breadth-first**.

Algoritma singkat untuk mengunjungi semua titik dari suatu pohon dengan menggunakan metode breadth-first terdiri dari 5 langkah berikut ini :

1. Letakkan titik awal pada array yang disebut array BUKA.
2. Jika array BUKA kosong maka selesai tanpa ada penyelesaian. Jika array BUKA tidak kosong maka lanjutkan ke langkah berikutnya.



Gb. 6.3. Flowchart Metode Breadth-First

3. Pindahkan titik pertama dari array BUKA dan letakkan pada array yang disebut array TUTUP. Sebut titik ini dengan n.
4. Jabarkan titik n, sehingga didapat semua turunan dari titik n. Jika tidak ada turunan maka ulangilah langkah 2. Letakkan turunan pada akhir array BUKA dan berikan pointer (penunjuk) yang mengarah ke n.
5. Jika salah satu turunan adalah titik akhir maka metode

selesai dijalankan dan diperoleh suatu penyelesaian dengan cara menelusuri kembali pointernya. Jika turunan tersebut bukan titik akhir maka ulangi proses dari langkah 2.

Algoritma metode breadth-first di atas mengasumsikan bahwa titik awal dan titik akhir tidak sama. Flowchart metode Breadth-First dapat dilihat pada Gambar 6.3.

Contoh 6.2 :

Soal seperti pada contoh 6.1.

Mencari $x_{ij} \neq 0$ dalam tiap baris dan tiap kolom dengan menggunakan metode breadth-first. Disyaratkan bahwa titik akhir merupakan titik tujuan bila letak kolom titik tersebut sama dengan letak kolom titik awal.

Titik (2,3) dianggap sebagai titik awal. Letakkan titik (2,3) ini dalam array BUKA. Ini berarti BUKA tidak kosong, kita lanjutkan ke langkah berikutnya yaitu memindahkan titik (2,3) dari array BUKA ke array TUTUP. Kemudian kita jabarkan titik (2,3) yaitu mencari $x_{ij} \neq 0$ dalam baris yang sama. Kita dapatkan titik (2,1) dan (2,2). Berarti turunan tidak kosong. Letakkan titik (2,1) dan (2,2) ini dalam array BUKA. Kita periksa apakah kolom titik turunan ini sama dengan kolom titik awal ?. Kolom titik awal adalah 3 sedangkan kolom titik turunan adalah 1 dan 2. Berarti turunan ini bukan merupakan titik akhir. Kita kembali ke langkah 2.

BUKA tidak kosong, oleh karena itu kita pindahkan titik (2,1) dan (2,2) dalam array BUKA ke array TUTUP. Kemudian kita jabarkan salah satu yaitu titik (2,1), sehingga kita dapatkan turunan titik (1,1) Kita letakkan (1,1) dalam array BUKA. Kita periksa apakah kolomnya sama dengan kolom titik awal ?. Ternyata tidak sama. Untuk itu kita ulang ke langkah 2. BUKA tidak kosong. Pindahkan (2,2) ke dalam array TUTUP. Jabarkan (2,2) dan didapat turunan yaitu titik (3,2). Letakkan (3,2) pada akhir array BUKA. Periksa apakah (3,2) merupakan titik akhir ?. Ternyata tidak. Kita kembali ke langkah 2. BUKA tidak kosong. Ada titik (1,1), pindahkan (1,1) ke array TUTUP. Jabarkan (1,1), ternyata dari (1,1) tidak ditemukan $x_{ij} \neq 0$ dalam baris yang sama. Berarti tidak ada turunan, kembali ke langkah 2. BUKA tidak kosong, ada titik (3,2) dalam BUKA. Pindahkan titik (3,2) ke dalam TUTUP. Jabarkan (3,2) didapat turunan yaitu titik (3,3). Letakkan titik (3,3) dalam array BUKA. Kita periksa apakah (3,3) merupakan titik akhir ?. Ternyata kolom (3,3) sama dengan kolom titik awal berarti (3,3) merupakan titik akhir. Jadi selesailah proses pencarian $x_{ij} \neq 0$ dalam tiap baris dan tiap kolom yang sama. Penyelesaian didapat dengan cara menelusuri pointernya yaitu membalik urutan titik paling akhir dalam array TUTUP, yaitu :

(3,3) (3,2) (1,1) (2,2) (2,1) (2,3)

dibalik menjadi

(2,3) (2,1) (2,2) (1,1) (3,2) (3,3) ∎

Untuk menguji keoptimuman penyelesaian layak basis kita menggunakan metode MODI. Langkah awal dalam uji optimum ini adalah kita tentukan $U_1 = 0$. Kemudian kita mencari $x_{ij} \neq 0$. Untuk mencari $x_{ij} \neq 0$ kita gunakan metode breadth - first. Syarat titik yang ditemukan merupakan titik akhir adalah bila dalam suatu baris atau kolom yang sesuai tidak ditemukan $x_{ij} \neq 0$ lagi.

Tabel 6.1
metode breadth-first contoh 6.2

BUKA	TUTUP
(2,3)	-
(2,1) (2,2)	(2,3)
(2,2) (1,1)	(2,1) (2,3)
(1,1) (3,2)	(2,2) (2,1) (2,3)
(3,2) (3,3)	(1,1) (2,2) (2,1) (2,3)
(3,3)	(3,2) (1,1) (2,2) (2,1) (2,3)
-	(3,3)(3,2)(1,1)(2,2)(2,1)(2,3)

Sebagai titik awal metode breadth-first adalah titik $x_{ij} \neq 0$. Letak baris titik awal kita tulis dalam variabel pakbar dan letak kolom titik awal kita tulis dalam variabel pakkol. Untuk mencari $x_{ij} \neq 0$ dalam baris atau kolom secara bergantian kita buat variabel rakol. Jika rakol = 0 maka kita mencari $x_{ij} \neq 0$ dalam kolom yang sama, yang kita tulis dalam *subrutin cakol*. Setelah $x_{ij} \neq 0$ yang kita temukan disimpan dalam array turun, kita hitung U_i-nya.

Jika riskol = 1 maka kita mencari $x_{ij} \neq 0$ dalam baris yang sama yang kita tulis dalam *subrutin cabar* dan selanjutnya kita hitung V_j-nya. Langkah ini kita ulang - ulang selama masih kita temukan $x_{ij} \neq 0$. Setiap menjalankan subrutin cabar dan subrutin cakol selalu diikuti dengan menghitung U_i dan V_j. Langkah ini ditulis dalam *subrutin pindah*. Untuk mengetahui apakah kita masih menemukan $x_{ij} \neq 0$ kita membuat *subrutin temu*. Jika buk(1,1) = 0 berarti kita tidak lagi menemukan $x_{ij} \neq 0$ maka pengulangan subrutin pindah dan subrutin temu dihentikan dan berarti juga penghitungan U_i dan V_j telah selesai. Jika buk(1,1)≠0 maka pengulangan terus dijalankan. Proses pengulangan subrutin pindah dan temu ditulis dalam *subrutin modi*. Setelah U_i dan V_j diperoleh semua, kita hitung ongkos kesempatan (CA) dengan rumus CA = $U_i + V_j - c_{ij}$ (dalam BAB III lambangnya c'_{ij}) yang kita tulis dalam *subrutin camak*. Setelah CA kita dapatkan, kita mencari CA yang maksimum. Pencarian CA maksimum ini ditulis dalam *subrutin makca*.

Dalam subrutin makca tersebut kita dapatkan CA yang maksimum. Jika CA yang kita dapatkan negatif maka penyelesaian basis sudah optimum, dan sebaliknya bila CA maksimum yang kita dapatkan tidak negatif berarti penyelesaian belum optimum. Pengecekkan CA ini kita tulis dalam *subrutin optimal*.

Langkah Ketiga

Jika CA yang kita dapatkan positif maka kita harus menggeser penyelesaian basis dengan cara membuat lintasan tertutup. Pembuatan lintasan tertutup ini ditulis dalam *subrutin lintas*. Dalam membuat sub lintas tidak sederhana bila kita mengerjakannya dengan cara manual yaitu mencari $x_{ij} \neq 0$ dalam baris atau kolom yang sama. Untuk pembuatan program kita harus memperhatikan semua kemungkinan $x_{ij} \neq 0$ yang dapat masuk dalam pembuatan lintasan tertutup. Untuk mencari semua kemungkinan $x_{ij} \neq 0$ ini kita gunakan juga **metode breadth-first**. Sebagai titik awal metode breadth-first untuk membuat lintasan tertutup adalah kotak yang mempunyai CA maksimum. Letak baris titik awal lintasan disimpan dalam variabel BPKBAR letak kolom titik awal lintasan ditulis dalam variabel BPKKOL dan BARKOL untuk mencari $x_{ij} \neq 0$ dalam baris atau kolom. Syarat titik yang ditemukan merupakan titik akhir adalah jika kolom titik tersebut sama dengan kolom titik awal. Pencarian kemungkinan $x_{ij} \neq 0$ ini kita tulis dalam *subrutin ekspan*.

Sel pertama lintasan tertutup adalah kotak yang mempunyai ongkos kesempatan paling tinggi (CA maksimum). Dari sel pertama ini kita mencari $x_{ij} \neq 0$ dalam baris atau kolom yang sama. Mencari $x_{ij} \neq 0$ dalam baris yang sama kita tulis dalam *subrutin cartembar*, sedangkan mencari x_{ij}

$\neq 0$ dalam kolom yang sama kita tulis dalam *subrutin cartemkol*. Setiap kita menemukan $x_{ij} \neq 0$, letak baris dan kolomnya kita catat dalam array turunan. Setelah kita catat dalam array turunan, letak baris dan kolom $x_{ij} \neq 0$ tersebut kita masukkan ke array buka. Setelah kita masukkan dalam array buka, letak baris dan kolom $x_{ij} \neq 0$ kita pindah ke dalam array anak dan bapak.

Setelah pencarian $x_{ij} \neq 0$ dalam baris atau kolom yang sama selesai dikerjakan, selanjutnya kita cek apakah letak kolom $x_{ij} \neq 0$ yang ditemukan sama dengan letak kolom sel pertama lintasan tertutup. Jika letak kolom $x_{ij} \neq 0$ yang didapatkan tidak sama dengan letak kolom sel awal lintasan tertutup maka pencarian $x_{ij} \neq 0$ dalam baris atau kolom masih dikerjakan terus, tetapi jika letak kolom $x_{ij} \neq 0$ yang ditemukan sudah sama dengan letak kolom sel pertama lintasan tertutup maka subrutin ekspan tidak dijalankan lagi. Untuk mengecek letak kolom $x_{ij} \neq 0$ ini kita tulis dalam *subrutin cektemu*.

Setelah subrutin cektemu menunjukkan bahwa letak kolom $x_{ij} \neq 0$ yang didapat sama dengan letak kolom sel pertama lintasan tertutup, berarti pencarian $x_{ij} \neq 0$ dalam baris atau kolom yang sama selesai dikerjakan. Setelah pencarian $x_{ij} \neq 0$ selesai kita lanjutkan dengan mengurut-kan sel-sel yang kita peroleh tersebut agar menjadi lintasan tertutup. Mencari urutan lintasan tertutup ini kita tulis dalam *subrutin urut*. Dalam mencari urutan lintasan tertutup kita mulai dari array anak yang mempunyai letak kolom sama dengan letak kolom sel pertama lintasan tertutup, kemudian dilanjutkan ke array bapak dengan urutan yang sama dengan array anak. Setelah itu kita cari array anak yang mempunyai baris dan kolom sama dengan array bapak yang kita dapatkan paling akhir. Sesudah ketemu anaknya baru kita cari bapaknya. Pencarian pasangan anak dan bapak ini kita tulis dalam *subrutin carianak* sedangkan hasilnya kita tulis dalam array jalur yang menghasilkan urutan lintasan tertutup.

Setelah terjadi lintasan tertutup kita akan menggeser penyelesaian basis dengan cara mencari $x_{ij} \neq 0$ yang minimum yang ada dalam lintasan tertutup. Penggeseran penyelesaian basis ini kita tulis dalam *subrutin carimin*. Setelah kita geser penyelesaian basis berarti kita dapatkan penyelesaian yang baru. Program kita ulang mulai langkah kedua sampai kita dapatkan CA maksimum yang tidak negatif.

Hasil dari penghitungan tersebut di atas yaitu berupa tampilan nilai-nilai x_{ij} dan fungsi sasaran optimumnya. Setelah langkah-langkah penyusunan program dapat digambarkan, langkah berikutnya adalah membuat program. Penulisan program dalam komputer dapat dilihat dalam lampiran.

Lambang-lambang yang digunakan dalam penyusunan program masalah angkutan adalah sebagai berikut :

M = banyak sumber

N = banyak tujuan

Z	=	penawaran total
S	=	permintaan total
F	=	Fungsi sasaran
H	=	M + N - 1
JUM	=	jumlah kotak isi
A( )	=	array besar penawaran
B( )	=	array besar permintaan
C( )	=	array ongkos angkut
X( )	=	array banyak barang yang harus diangkut
CA( )	=	array ongkos kesempatan
U( )	=	array bilangan baris
V( )	=	array bilangan kolom
BAR	=	letak baris pada CA( ) terbesar
KOL	=	letak kolom pada CA( ) terbesar
MAK	=	CA( ) terbesar
JALUR( )	=	array lintasan tertutup
TND( )	=	tanda selang-seling + atau - pada lintasan tertutup

D. PENGUJIAN PROGRAM

Untuk menguji program masalah angkutan kita ambil soal masalah angkutan setimbang dengan 5 sumber dan 4 tujuan berikut ini :

(4)	(6)	(4)	(3)	40
(5)	(7)	(4)	(2)	40
(6)	(8)	(4)	(7)	35
(5)	(6)	(3)	(4)	40
(5)	(4)	(2)	(3)	45
55	40	40	65	200

Soal di atas jika diselesaikan secara manual akan menjadi sebagai berikut :

Tabel 6.2

40 (4)	(6)	(4)	(3)	40	0
15 (5)	25 (7)	(4)	(2)	40	1
(6)	15 (8)	20 (4)	(7)	35	2
(5)	(6)	20 (3)	20 (4)	40	1
(5)	(4)	(2)	45 (3)	45	0
55	40	40	65		
4	6	2	3		

Tabel 6.3

40 (4)	(6)	(4)	(3)	40	0
15 (5)	5 (7)	(4)	(2)	40	1
(6)	35 (8)	ε (4)	(7)	35	2
(5)	(6)	40 (3)	(4)	40	1
(5)	(4)	(2)	45 (3)	45	2
55	40	40	65		
4	6	2	1		

Tabel 6.4

40 (4)	(6)	(4)	(3)	40	0
15 (5)	(7)	(4)	25 (2)	40	1
(6)	35 (8)	ε (4)	(7)	35	6
(5)	(6)	40 (3)	(4)	40	5
(5)	5 (4)	(2)	40 (3)	45	2
55	40	40	65		
4	2	-2	1		

Tabel 6.5

40 (4)	(6)	(4)	(3)	40	0
(5)	(7)	(4)	40 (2)	40	-3
15 (6)	20 (8)	ε (4)	(7)	35	2
(5)	(6)	40 (3)	(4)	40	1
(5)	20 (4)	(2)	25 (3)	45	-2
55	40	40	65		
4	6	2	5		

Tabel 6.6

20 ^{4}	^{6}	^{4}	20 ^{3}	40	0
^{5}	^{7}	^{4}	40 ^{2}	40	-1
35 ^{6}	^{8}	ε ^{4}	^{7}	35	2
^{5}	^{6}	40 ^{3}	^{4}	40	1
^{5}	40 ^{4}	^{2}	5 ^{3}	45	0
55	40	40	65		
4	4	2	3		

Jadi penyelesaian optimum masalah di atas adalah :

$x_{11} = 20$ $\quad x_{12} = 0$ $\quad x_{13} = 0$ $\quad x_{14} = 20$

$x_{21} = 0$ $\quad x_{22} = 0$ $\quad x_{23} = 0$ $\quad x_{24} = 40$

$x_{31} = 35$ $\quad x_{32} = 0$ $\quad x_{33} = \varepsilon$ $\quad x_{34} = 0$

$x_{41} = 0$ $\quad x_{42} = 0$ $\quad x_{43} = 40$ $\quad x_{44} = 0$

$x_{51} = 0$ $\quad x_{52} = 40$ $\quad x_{53} = 0$ $\quad x_{54} = 5$

Fungsi sasaran masalah di atas adalah :

$$
\begin{aligned}
f &= 4.20 + 3.20 + 2.40 + 6.35 + 3.40 + 4.40 + 3.5 \\
&= 80 + 60 + 80 + 210 + 120 + 160 + 15 \\
&= 725 \quad \blacksquare
\end{aligned}
$$

Hasil pengujian dengan menggunakan komputer adalah :

```
BANYAK SUMBER = 5
BANYAK TUJUAN = 4
BESAR PENAWARAN KE  1 =? 40
BESAR PENAWARAN KE  2 =? 40
BESAR PENAWARAN KE  3 =? 35
BESAR PENAWARAN KE  4 =? 40
BESAR PENAWARAN KE  5 =? 45
PENAWARAN TOTAL =  200
BESAR PERMINTAAN KE  1 =? 55
BESAR PERMINTAAN KE  2 =? 40
BESAR PERMINTAAN KE  3 =? 40
BESAR PERMINTAAN KE  4 =? 65
PERMINTAAN TOTAL =  200
TEKAN TOMBOL ENTER?


ONGKOS ANGKUT DARI SUMBER 1 KE TUJUAN 1 =? 4
ONGKOS ANGKUT DARI SUMBER 1 KE TUJUAN 2 =? 6
ONGKOS ANGKUT DARI SUMBER 1 KE TUJUAN 3 =? 4
ONGKOS ANGKUT DARI SUMBER 1 KE TUJUAN 4 =? 3
ONGKOS ANGKUT DARI SUMBER 2 KE TUJUAN 1 =? 5
ONGKOS ANGKUT DARI SUMBER 2 KE TUJUAN 2 =? 7
ONGKOS ANGKUT DARI SUMBER 2 KE TUJUAN 3 =? 4
ONGKOS ANGKUT DARI SUMBER 2 KE TUJUAN 4 =? 2
ONGKOS ANGKUT DARI SUMBER 3 KE TUJUAN 1 =? 6
ONGKOS ANGKUT DARI SUMBER 3 KE TUJUAN 2 =? 8
ONGKOS ANGKUT DARI SUMBER 3 KE TUJUAN 3 =? 4
ONGKOS ANGKUT DARI SUMBER 3 KE TUJUAN 4 =? 7
ONGKOS ANGKUT DARI SUMBER 4 KE TUJUAN 1 =? 5
ONGKOS ANGKUT DARI SUMBER 4 KE TUJUAN 2 =? 6
ONGKOS ANGKUT DARI SUMBER 4 KE TUJUAN 3 =? 3
ONGKOS ANGKUT DARI SUMBER 4 KE TUJUAN 4 =? 4
ONGKOS ANGKUT DARI SUMBER 5 KE TUJUAN 1 =? 5
ONGKOS ANGKUT DARI SUMBER 5 KE TUJUAN 2 =? 4
ONGKOS ANGKUT DARI SUMBER 5 KE TUJUAN 3 =? 2
ONGKOS ANGKUT DARI SUMBER 5 KE TUJUAN 4 =? 3
MEMAKSIMUMKAN (MAKS) / MEMINIMUMKAN (MIN) FUNGSI SASARAN? min


A( 1 )= 40     A( 2 )= 40     A( 3 )= 35     A( 4 )= 40     A( 5 )= 45

B( 1 )= 55     B( 2 )= 40     B( 3 )= 40     B( 4 )= 65
PENYELESAIAN AWALNYA ADALAH
X( 1 , 1 )= 40            X( 1 , 2 )= 0             X( 1 , 3 )= 0
X( 1 , 4 )= 0             X( 2 , 1 )= 15            X( 2 , 2 )= 25
X( 2 , 3 )= 0             X( 2 , 4 )= 0             X( 3 , 1 )= 0
X( 3 , 2 )= 15            X( 3 , 3 )= 20            X( 3 , 4 )= 0
X( 4 , 1 )= 0             X( 4 , 2 )= 0             X( 4 , 3 )= 20
X( 4 , 4 )= 20            X( 5 , 1 )= 0             X( 5 , 2 )= 0
X( 5 , 3 )= 0             X( 5 , 4 )= 45
JUMLAH KOTAK ISI =  8
PENYELESAIAN TIDAK MEROSOT
FUNGSI SASARAN  =  885
ENTER?
```

```
BILANGAN MODINYA ADALAH
U( 1 )= 0       U( 2 )= 1       U( 3 )= 2       U( 4 )= 1       U( 5 )= 0

V( 1 )= 4       V( 2 )= 6       V( 3 )= 2       V( 4 )= 3
CA( 1   1 )= 0                  CA( 1   2 )= 0                  CA( 1   3 )=-2
CA( 1   4 )= 0                  CA( 2   1 )= 0                  CA( 2   2 )= 0
CA( 2   3 )=-1                  CA( 2   4 )= 2                  CA( 3   1 )= 0
CA( 3   2 )= 0                  CA( 3   3 )= 0                  CA( 3   4 )=-2
CA( 4   1 )= 0                  CA( 4   2 )= 1                  CA( 4   3 )= 0
CA( 4   4 )= 0                  CA( 5   1 )=-1                  CA( 5   2 )= 2
CA( 5   3 )= 0                  CA( 5   4 )= 0                  ENTER?


URUTAN LINTASAN TERTUTUP
JALUR( 1 ,1) = 4                JALUR( 1 ,2) = 4
JALUR( 2 ,1) = 4                JALUR( 2 ,2) = 3
JALUR( 3 ,1) = 3                JALUR( 3 ,2) = 3
JALUR( 4 ,1) = 3                JALUR( 4 ,2) = 2
JALUR( 5 ,1) = 2                JALUR( 5 ,2) = 2
JALUR( 6 ,1) = 2                JALUR( 6 ,2) = 4
ENTER?


X( 1 , 1 )= 40                  X( 1 , 2 )= 0                   X( 1 , 3 )= 0
X( 1 , 4 )= 0                   X( 2 , 1 )= 15                  X( 2 , 2 )= 5
X( 2 , 3 )= 0                   X( 2 , 4 )= 20                  X( 3 , 1 )= 0
X( 3 , 2 )= 35                  X( 3 , 3 )=-1                   X( 3 , 4 )= 0
X( 4 , 1 )= 0                   X( 4 , 2 )= 0                   X( 4 , 3 )= 40
X( 4 , 4 )= 0                   X( 5 , 1 )= 0                   X( 5 , 2 )= 0
X( 5 , 3 )= 0                   X( 5 , 4 )= 45
JUMLAH KOTAK ISI = 8
FUNGSI SASARAN = 845
ENTER?


BILANGAN MODINYA ADALAH
U( 1 )= 0       U( 2 )= 1       U( 3 )= 2       U( 4 )= 1       U( 5 )= 2

V( 1 )= 4       V( 2 )= 6       V( 3 )= 2       V( 4 )= 1
CA( 1   1 )= 0                  CA( 1   2 )= 0                  CA( 1   3 )=-2
CA( 1   4 )=-2                  CA( 2   1 )= 0                  CA( 2   2 )= 0
CA( 2   3 )=-1                  CA( 2   4 )= 0                  CA( 3   1 )= 0
CA( 3   2 )= 0                  CA( 3   3 )= 0                  CA( 3   4 )=-4
CA( 4   1 )= 0                  CA( 4   2 )= 1                  CA( 4   3 )= 0
CA( 4   4 )=-2                  CA( 5   1 )= 1                  CA( 5   2 )= 4
CA( 5   3 )= 2                  CA( 5   4 )= 0                  ENTER?
```

```
URUTAN LINTASAN TERTUTUP
JALUR( 1 ,1) = 2            JALUR( 1 ,2) = 2
JALUR( 2 ,1) = 2            JALUR( 2 ,2) = 4
JALUR( 3 ,1) = 5            JALUR( 3 ,2) = 4
JALUR( 4 ,1) = 5            JALUR( 4 ,2) = 2
ENTER?


X( 1 , 1 )= 40              X( 1 , 2 )= 0               X( 1 , 3 )= 0
X( 1 , 4 )= 0               X( 2 , 1 )= 15              X( 2 , 2 )= 0
X( 2 , 3 )= 0               X( 2 , 4 )= 25              X( 3 , 1 )= 0
X( 3 , 2 )= 35              X( 3 , 3 )=-1               X( 3 , 4 )= 0
X( 4 , 1 )= 0               X( 4 , 2 )= 0               X( 4 , 3 )= 40
X( 4 , 4 )= 0               X( 5 , 1 )= 0               X( 5 , 2 )= 5
X( 5 , 3 )= 0               X( 5 , 4 )= 40
JUMLAH KOTAK ISI = 8
FUNGSI SASARAN = 825
ENTER?


BILANGAN MODINYA ADALAH
U( 1 )= 0      U( 2 )= 1      U( 3 )= 6      U( 4 )= 5      U( 5 )= 2

V( 1 )= 4      V( 2 )= 2      V( 3 )=-2      V( 4 )= 1
CA( 1  1 )= 0               CA( 1  2 )=-4               CA( 1  3 )=-6
CA( 1  4 )=-2               CA( 2  1 )= 0               CA( 2  2 )=-4
CA( 2  3 )=-5               CA( 2  4 )= 0               CA( 3  1 )= 4
CA( 3  2 )= 0               CA( 3  3 )= 0               CA( 3  4 )= 0
CA( 4  1 )= 4               CA( 4  2 )= 1               CA( 4  3 )= 0
CA( 4  4 )= 2               CA( 5  1 )= 1               CA( 5  2 )= 0
CA( 5  3 )=-2               CA( 5  4 )= 0               ENTER?


URUTAN LINTASAN TERTUTUP
JALUR( 1 ,1) = 2            JALUR( 1 ,2) = 1
JALUR( 2 ,1) = 2            JALUR( 2 ,2) = 4
JALUR( 3 ,1) = 5            JALUR( 3 ,2) = 4
JALUR( 4 ,1) = 5            JALUR( 4 ,2) = 2
JALUR( 5 ,1) = 3            JALUR( 5 ,2) = 2
JALUR( 6 ,1) = 3            JALUR( 6 ,2) = 1
ENTER?


X( 1 , 1 )= 40              X( 1 , 2 )= 0               X( 1 , 3 )= 0
X( 1 , 4 )= 0               X( 2 , 1 )= 0               X( 2 , 2 )= 0
X( 2 , 3 )= 0               X( 2 , 4 )= 40              X( 3 , 1 )= 15
X( 3 , 2 )= 20              X( 3 , 3 )=-1               X( 3 , 4 )= 0
X( 4 , 1 )= 0               X( 4 , 2 )= 0               X( 4 , 3 )= 40
X( 4 , 4 )= 0               X( 5 , 1 )= 0               X( 5 , 2 )= 20
X( 5 , 3 )= 0               X( 5 , 4 )= 25
JUMLAH KOTAK ISI = 8
FUNGSI SASARAN = 765
ENTER?
```

```
BILANGAN MODINYA ADALAH
U( 1 )= 0      U( 2 )=-3      U( 3 )= 2      U( 4 )= 1      U( 5 )=-2

V( 1 )= 4      V( 2 )= 6      V( 3 )= 2      V( 4 )= 5
CA( 1   1 )= 0                CA( 1   2 )= 0                CA( 1   3 )=-2
CA( 1   4 )= 2                CA( 2   1 )=-4                CA( 2   2 )=-4
CA( 2   3 )=-5                CA( 2   4 )= 0                CA( 3   1 )= 0
CA( 3   2 )= 0                CA( 3   3 )= 0                CA( 3   4 )= 0
CA( 4   1 )= 0                CA( 4   2 )= 1                CA( 4   3 )= 0
CA( 4   4 )= 2                CA( 5   1 )=-3                CA( 5   2 )= 0
CA( 5   3 )=-2                CA( 5   4 )= 0                ENTER?


URUTAN LINTASAN TERTUTUP
JALUR( 1 ,1) = 5              JALUR( 1 ,2) = 4
JALUR( 2 ,1) = 5              JALUR( 2 ,2) = 2
JALUR( 3 ,1) = 3              JALUR( 3 ,2) = 2
JALUR( 4 ,1) = 3              JALUR( 4 ,2) = 1
JALUR( 5 ,1) = 1              JALUR( 5 ,2) = 1
JALUR( 6 ,1) = 1              JALUR( 6 ,2) = 4
ENTER?


X( 1 , 1 )= 20                X( 1 , 2 )= 0                 X( 1 , 3 )= 0
X( 1 , 4 )= 20                X( 2 , 1 )= 0                 X( 2 , 2 )= 0
X( 2 , 3 )= 0                 X( 2 , 4 )= 40                X( 3 , 1 )= 35
X( 3 , 2 )= 0                 X( 3 , 3 )=-1                 X( 3 , 4 )= 0
X( 4 , 1 )= 0                 X( 4 , 2 )= 0                 X( 4 , 3 )= 40
X( 4 , 4 )= 0                 X( 5 , 1 )= 0                 X( 5 , 2 )= 40
X( 5 , 3 )= 0                 X( 5 , 4 )= 5
JUMLAH KOTAK ISI = 8
FUNGSI SASARAN = 725
ENTER?


BILANGAN MODINYA ADALAH
U( 1 )= 0      U( 2 )=-1      U( 3 )= 2      U( 4 )= 1      U( 5 )= 0

V( 1 )= 4      V( 2 )= 4      V( 3 )= 2      V( 4 )= 3
CA( 1   1 )= 0                CA( 1   2 )=-2                CA( 1   3 )=-2
CA( 1   4 )= 0                CA( 2   1 )=-2                CA( 2   2 )=-4
CA( 2   3 )=-3                CA( 2   4 )= 0                CA( 3   1 )= 0
CA( 3   2 )=-2                CA( 3   3 )= 0                CA( 3   4 )=-2
CA( 4   1 )= 0                CA( 4   2 )=-1                CA( 4   3 )= 0
CA( 4   4 )= 0                CA( 5   1 )=-1                CA( 5   2 )= 0
CA( 5   3 )= 0                CA( 5   4 )= 0                ENTER?
```

```
PENYELESAIAN OPTIMUMNYA ADALAH
X( 1 , 1 )= 20          X( 1 , 2 )= 0          X( 1 , 3 )= 0
X( 1 , 4 )= 20          X( 2 , 1 )= 0          X( 2 , 2 )= 0
X( 2 , 3 )= 0           X( 2 , 4 )= 40         X( 3 , 1 )= 35
X( 3 , 2 )= 0           X( 3 , 3 )=-1          X( 3 , 4 )= 0
X( 4 , 1 )= 0           X( 4 , 2 )= 0          X( 4 , 3 )= 40
X( 4 , 4 )= 0           X( 5 , 1 )= 0          X( 5 , 2 )= 40
X( 5 , 3 )= 0           X( 5 , 4 )= 5
JUMLAH KOTAK ISI = 8
FUNGSI SASARAN OPTIMUM = 725
ENTER?
```

BAB VII
PENUTUP

A. KESIMPULAN

Masalah angkutan merupakan kejadian khusus dari masalah program linear. Oleh karena itu masalah angkutan dapat dikerjakan dengan menggunakan metode simpleks maupun matriks. Akan tetapi perhitungan masalah angkutan dengan menggunakan metode simpleks maupun matriks sangat panjang dan sulit, sehingga diciptakan metode tersendiri yaitu metode angkutan. Penyelesaian dengan menggunakan metode angkutan ini akan lebih singkat dibandingkan bila masalah dikerjakan dengan metode simpleks maupun matriks.

Pengerjaan dengan metode angkutan terdiri atas 3 langkah, yaitu :

1. Menyusun penyelesaian layak basis awal yang tidak merosot
2. Uji optimum
3. Memperbaiki penyelesaian bila penyelesaian yang didapat belum optimum

Ada enam metode yang dibahas untuk menentukan penyelesaian awal, yaitu metode sudut barat laut, metode baris minimum, metode kolom minimum, metode matriks minimum, metode Vogel dan metode Russel. Dari keenam metode tersebut di atas hanya metode sudut barat laut yang tidak memperhatikan ongkos, sehingga fungsi sasaran yang diperoleh masih jauh dari optimum dibandingkan bila soal dikerjakan dengan menggunakan metode yang lain yang sudah memperhatikan ongkos. Untuk uji optimum ada dua metode yang dibahas yaitu metode Stepping Stone dan metode MODI.

Suatu penyelesaian layak basis dikatakan optimum bila ongkos kesempatan (c'_{ij}) tak positif. Setelah uji optimum dan ternyata masih ada $c'_{ij} > 0$ maka diperlukan perbaikan penyelesaian dengan cara mengganti satu penyelesaian basisnya. Apabila $c'_{ij} = 0$ berarti ada pilihan penyelesaian optimum dengan nilai fungsi sasaran sama dengan nilai fungsi sasaran yang sudah diperoleh sebelumnya.

Suatu penyelesaian layak basis masalah angkutan dengan m sumber dan n tujuan terdiri atas m+n-1 buah $x_{ij}>0$. Jika banyaknya $x_{ij} > 0$ kurang dari m+n-1 maka penyelesaian layak basis disebut merosot. Kemerosotan dapat terjadi sewaktu kita menyusun tabel awal mupun dalam tabel bukan awal yaitu pada waktu menentukan basis berikutnya.

Kejadian masalah angkutan ada 2 macam yaitu masalah angkutan setimbang dan masalah angkutan tidak setimbang. Masalah angkutan yang tidak setimbang terdiri dari dua kejadian yaitu masalah angkutan tidak setimbang layak dan masalah angkutan tidak setimbang tidak layak. Masalah angkutan setimbang yaitu masalah dengan penawaran total sama dengan permintaan total. Masalah angkutan tidak setimbang layak yaitu masalah dengan penawaran total lebih besar dari permintaan total, sedangkan masalah angkutan tidak setimbang tidak layak yaitu masalah dengan penawaran total lebih kecil dari permintaan total. Untuk menyelesaikan masalah angkutan yang tidak setimbang harus disetimbangkan terlebih dahulu dengan cara menambah sumber semu dan tujuan semu. Dengan ditambahnya sumber semu dan tujuan semu seolah-olah ada sumber baru atau tujuan baru yang memberikan kekurangan besar penawaran atau permintaan. Ongkos angkut dari sumber ke tujuan semu maupun dari sumber semu ke tujuan adalah nol.

Teori masalah angkutan yang dibahas di sini berkaitan dengan masalah angkutan setimbang yang berpola minimum. Ada masalah angkutan yang berpola maksimum, ada juga masalah yang merupakan kejadian khusus dari masalah angkutan yaitu masalah penugasan dan ada juga masalah yang merupakan bentuk umum dari masalah angkutan yaitu masalah pemindahan muatan dan masalah penjaluran tangki. Masalah tersebut di atas dapat diselesaikan dengan menggunakan metode angkutan dengan cara mengembalikan masalah tersebut ke masalah angkutan berpola minimum. Meskipun masalah tersebut di atas dapat dikembalikan ke masalah angkutan pola minimum, untuk mempermudah penyelesaiannya telah disediakan metode tersendiri sehingga jalan penyelesaiannya lebih ringkas.

Masalah angkutan yang berformat besar bila dikerjakan dengan cara manual akan memakan tempat dan waktu yang tidak sedikit. Oleh karena itu diciptakan program komputer yang dapat membantu kita dalam melakukan komputasi. Bahasa komputer yang digunakan untuk membuat program masalah angkutan ini adalah TURBO BASIC. Dalam membuat program diperlukan empat langkah penyusunan program, yaitu :

1. Pemahaman masalah yang akan diprogramkan
2. Merencanakan langkah proses (flowcharting)
3. Membuat program
4. Pengujian program

Untuk mencari penyelesaian layak basis awal dalam program digunakan metode sudut barat laut, untuk uji optimum digunakan metode MODI dan untuk membuat lintasan tertutup berdasarkan pada metode breadth first. Masalah yang dapat diselesaikan dengan program dalam BAB VI adalah masalah angkutan setimbang dan masalah angkutan tidak setimbang baik yang meminimumkan fungsi sasaran maupun yang memaksimumkan fungsi sasaran.

B. SARAN

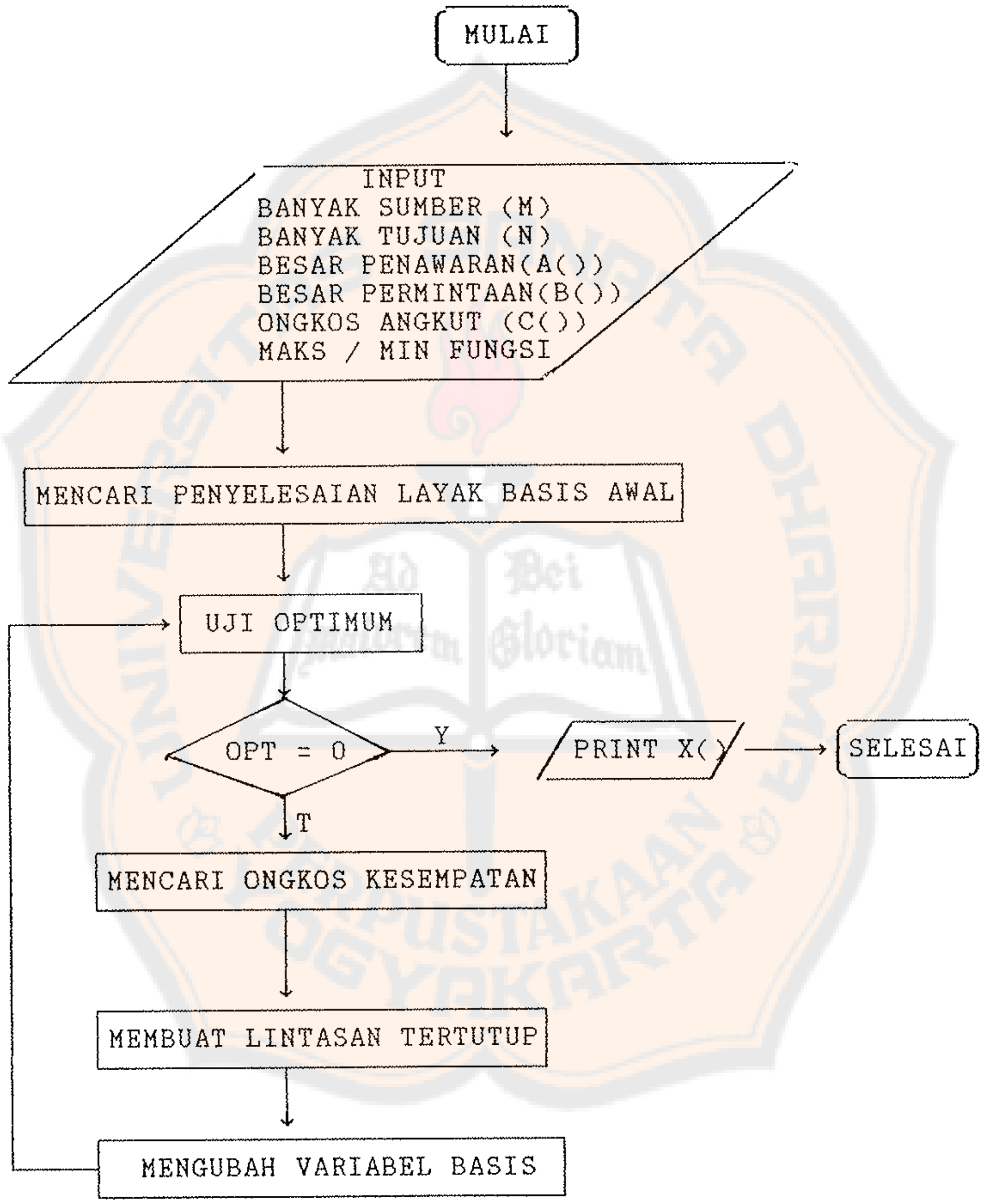
Pembaca dapat meneruskan komputasi masalah angkutan dengan bantuan komputer ini untuk masalah angkutan yang diperluas lainnya.

DAFTAR PUSTAKA

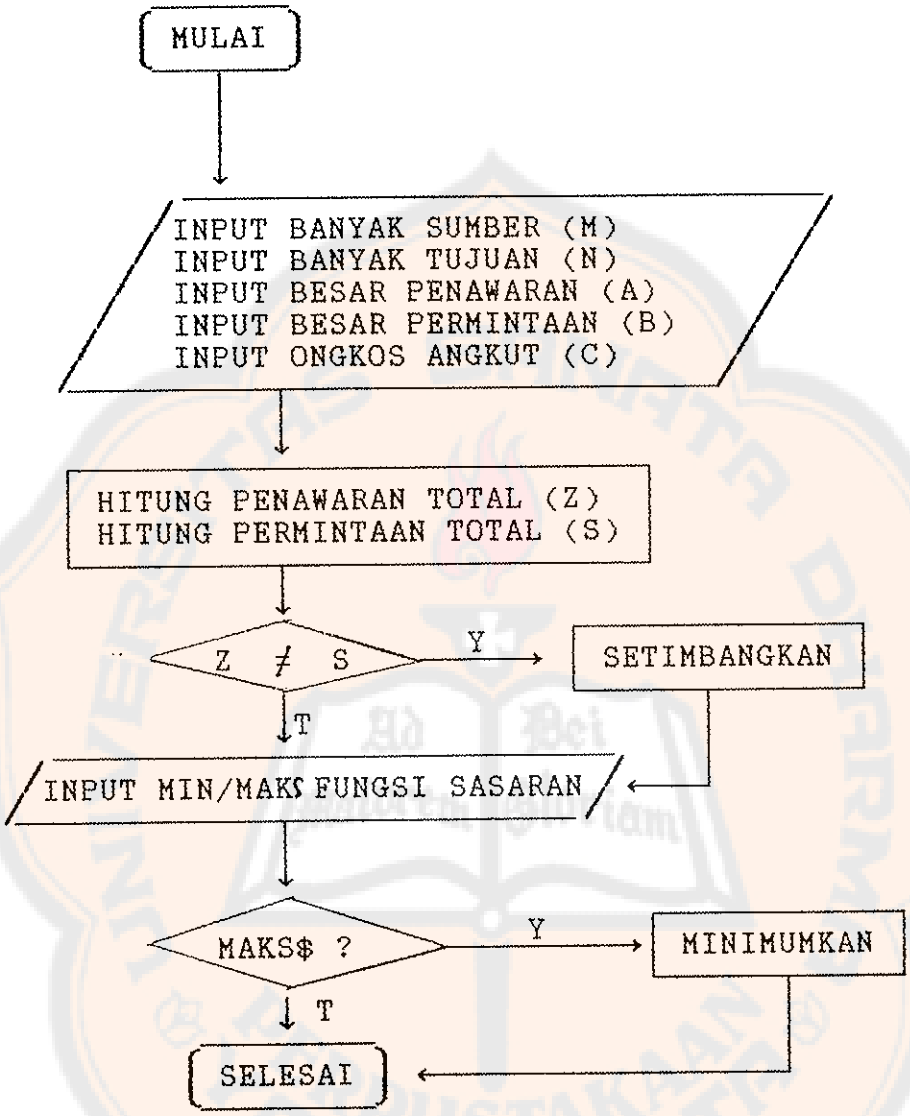
1. Eiselt,Horst A dan Frajer,Helmut, Operation Research Handbook, London and Basingstoke, The Macmillan press LTD, 1977.
2. Garvin,Walter W, Introduction to Linear Programming, New York, Mc Graww Hill Inc, 1960.
3. Gass,saul,I, Linear Programming, Tokyo, Koga Kusha Company LTD, 1969.
4. Kreko,Bela, Linear Programming, London, Sir Isaac Pitman & Sons LTD, 1968.
5. Loomba,Paul,N, Linear Programming, New York, San Fransisco, Torondo, London, 1964.
6. Mosher,Frederick,E dan Schneider,david,I, Using Turbo Basic, Philippines, Borland Osborne, Mc Graw Hill, 1991.
7. Susanta, B, Program Linear, Yogyakarta, Proyek Bahan Ajar MIPA, LPTK, Departemen Pendidikan dan Kebudayaan, 1996.

LAMPIRAN

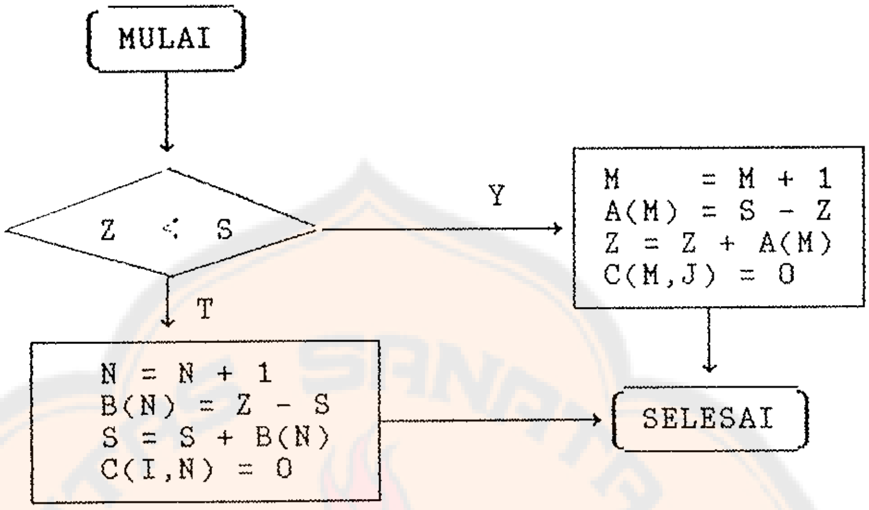
Flowchart menu utama masalah angkutan :



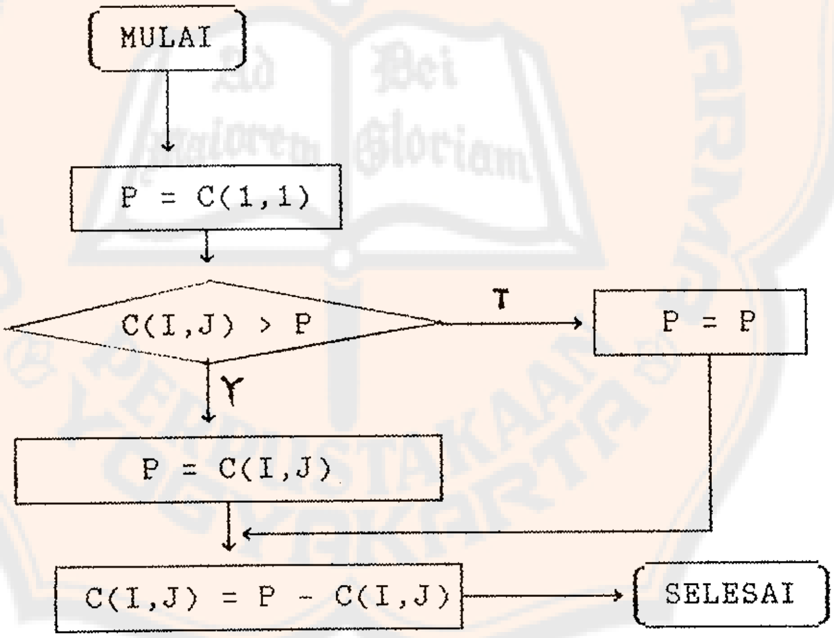
Flowchart sub masukan :



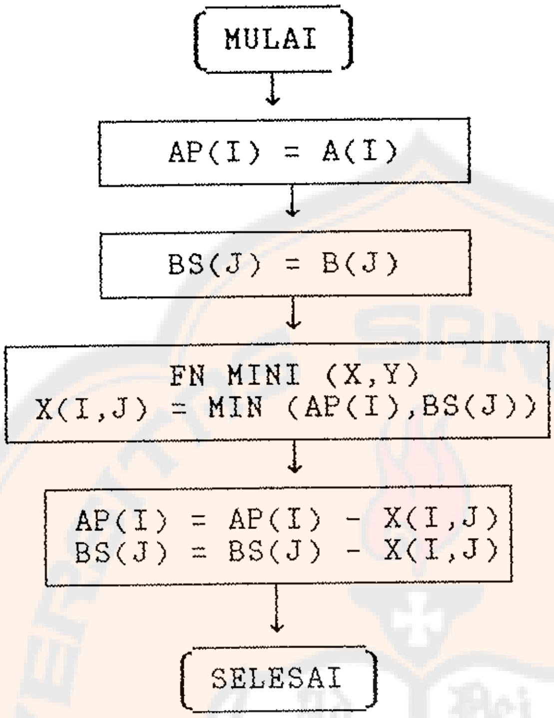
Flowchart sub setimbangkan :



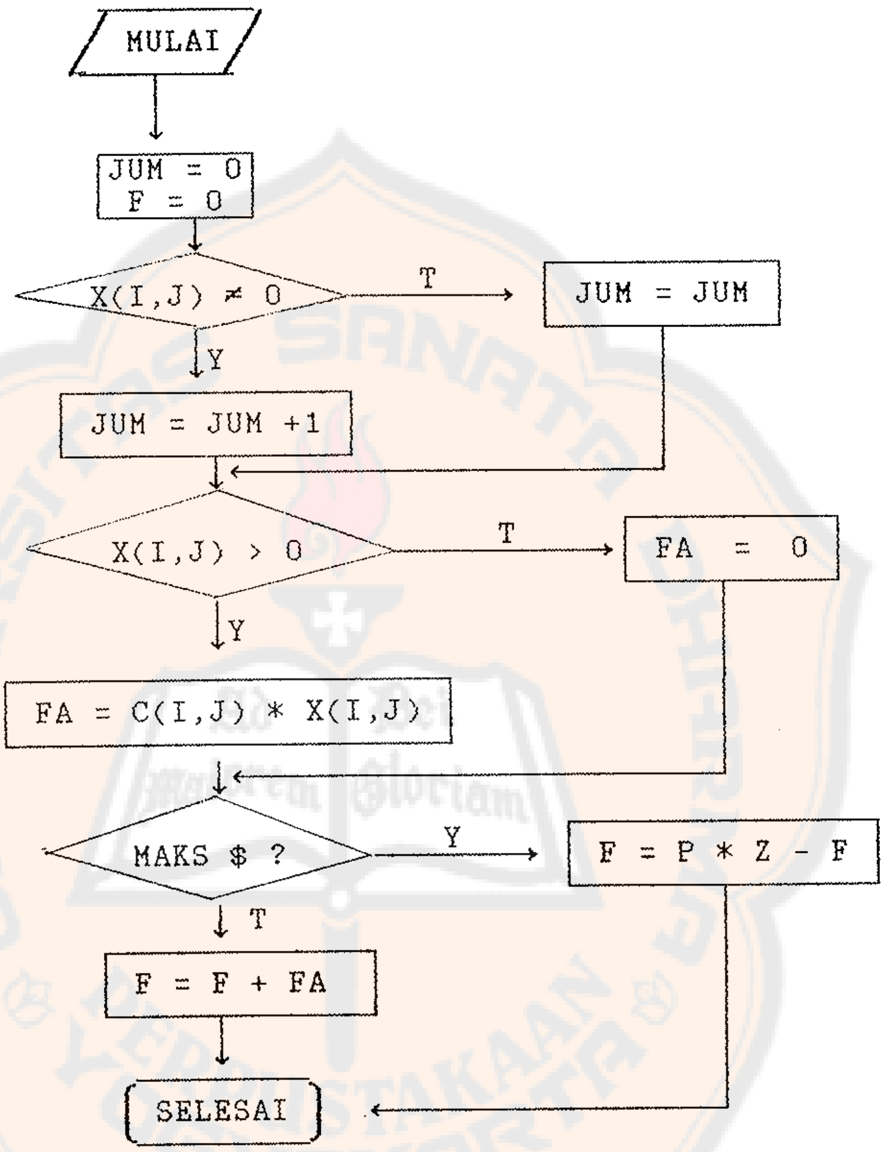
Flowchart sub maksimum :



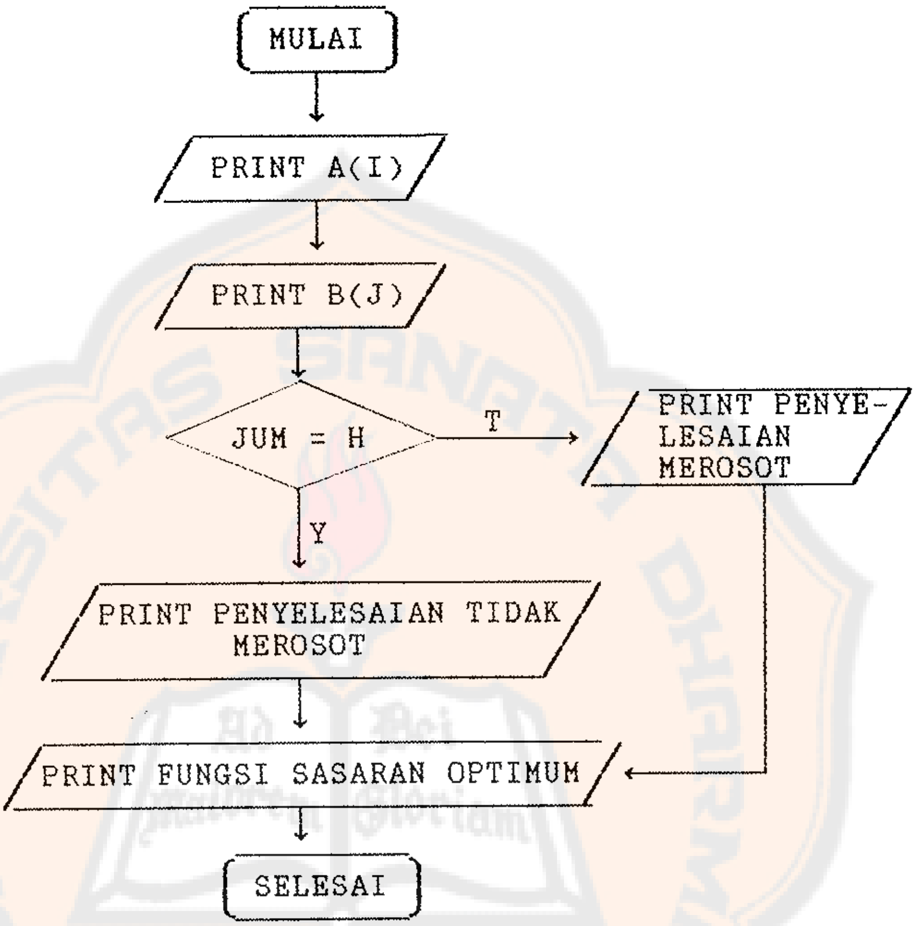
Flowchart Sub awal :



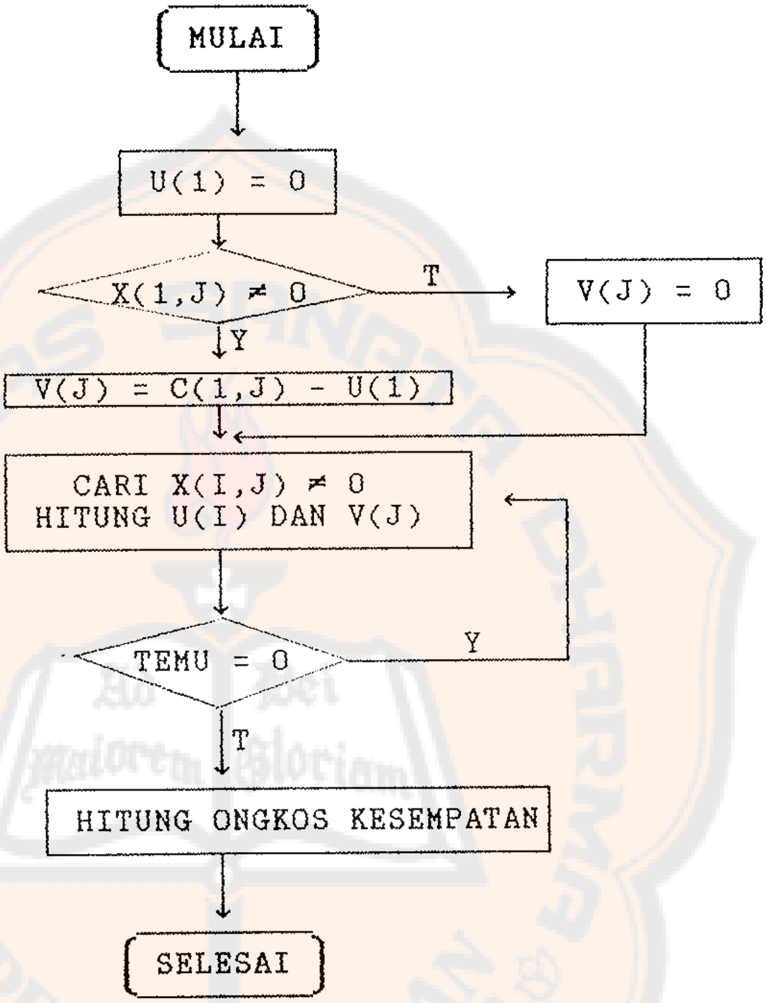
Flowchart Sub Sasaran :



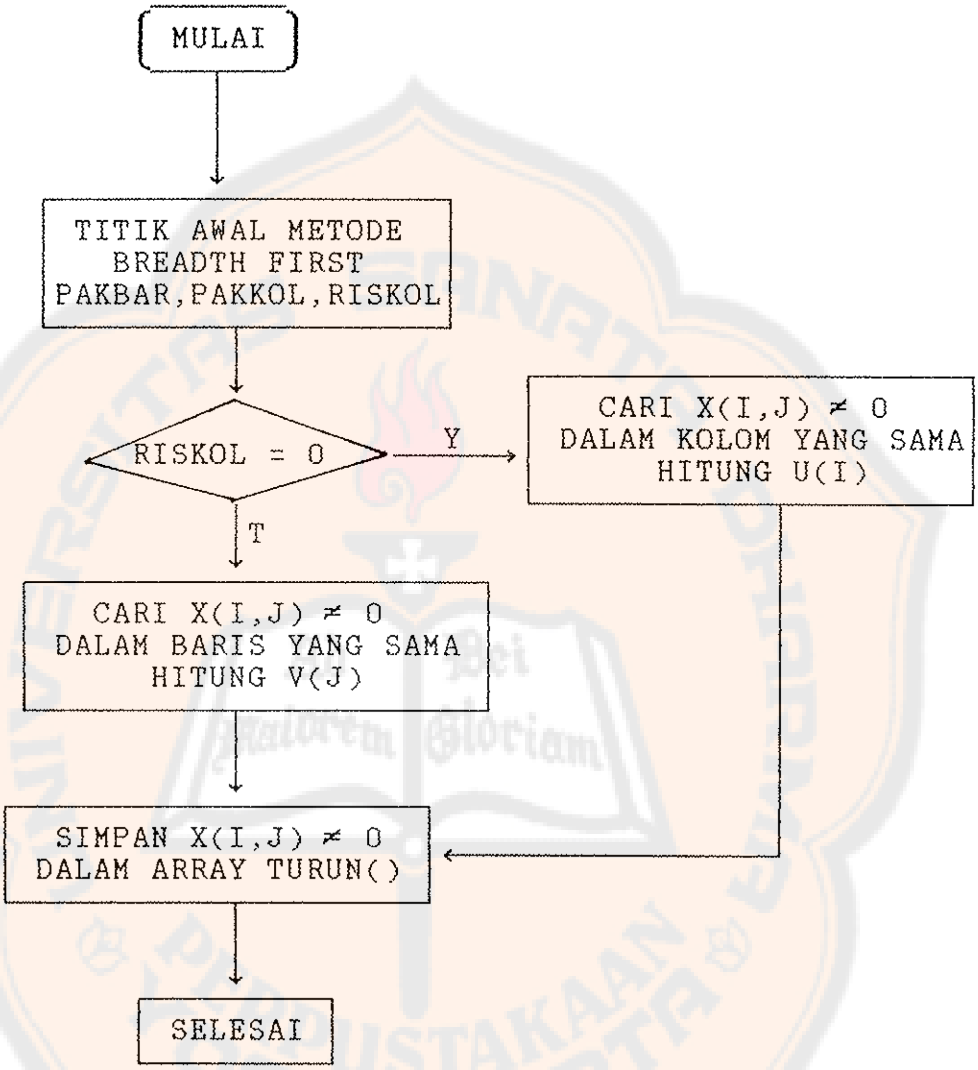
Flowchart Sub Cek 1 :



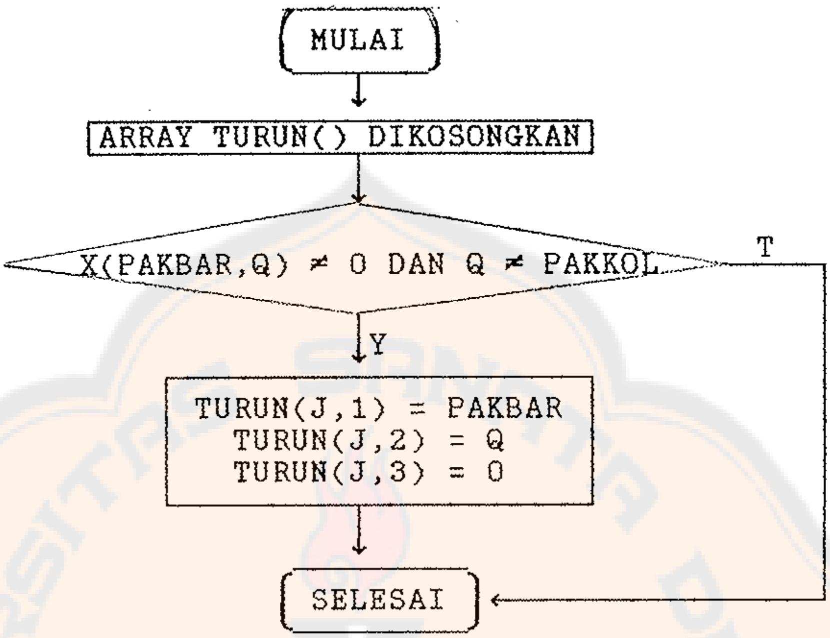
Flowchart Sub MODI :



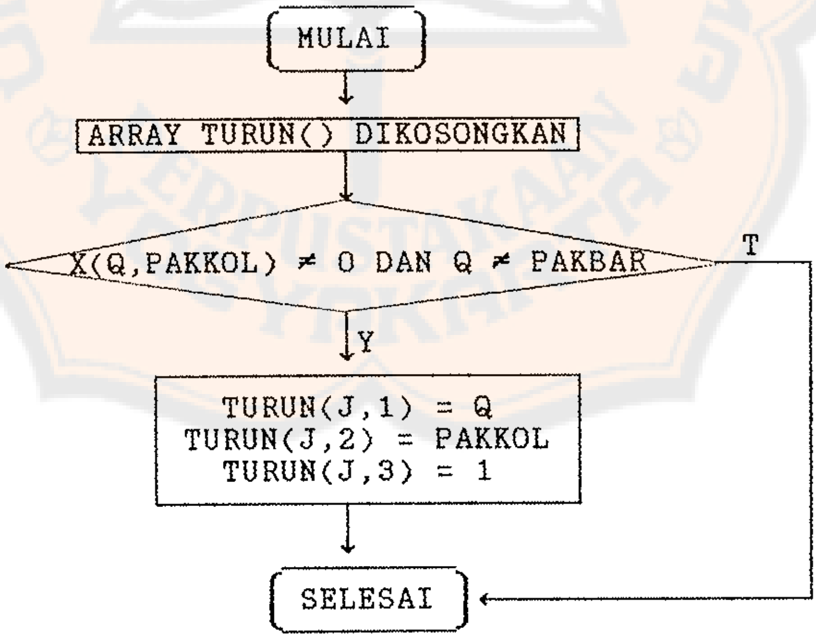
Flowchart Sub Pindah :



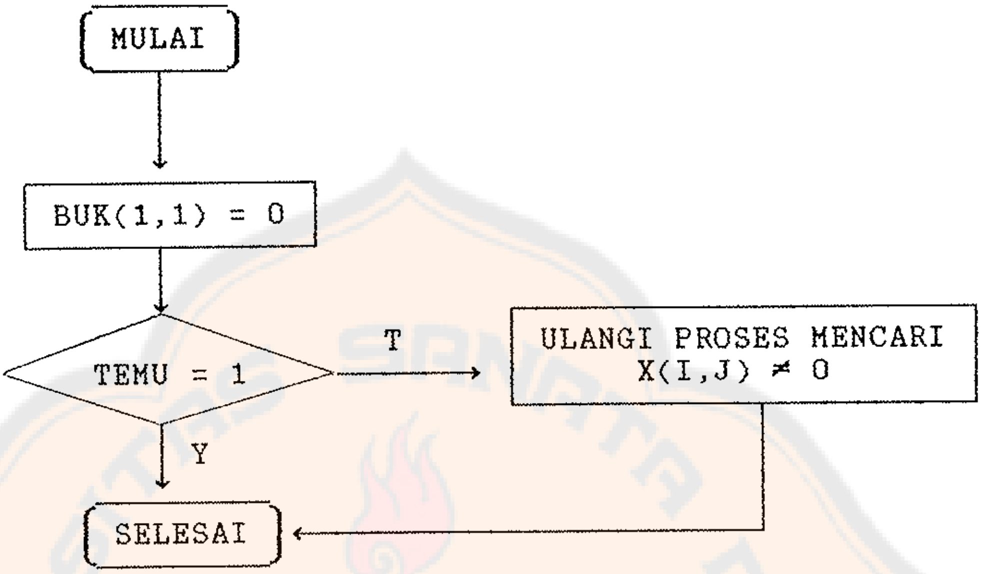
Flowchart Sub Cabar :



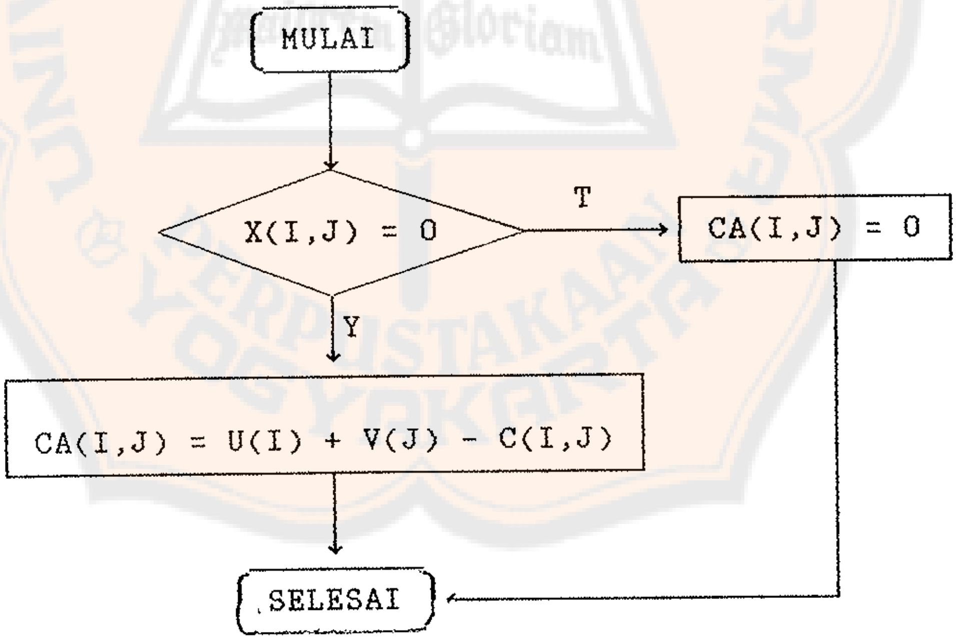
Flowchart Sub Cakol :



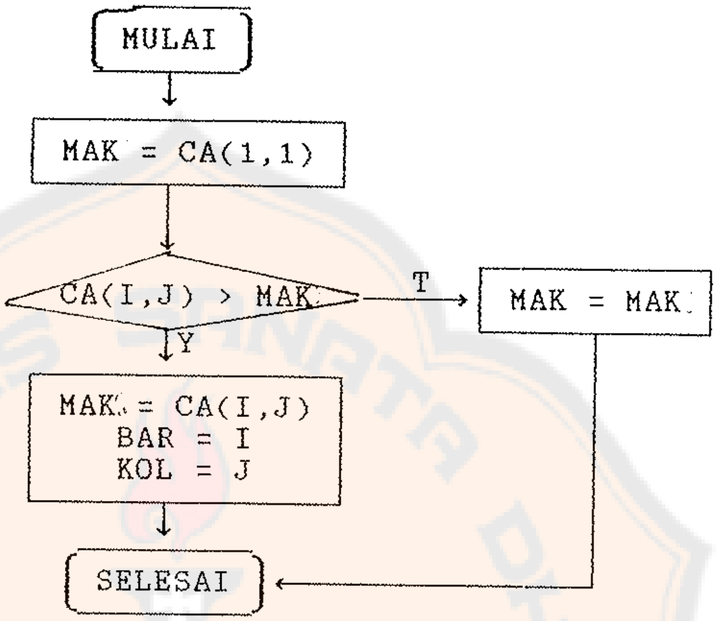
Flowchart Sub temu :



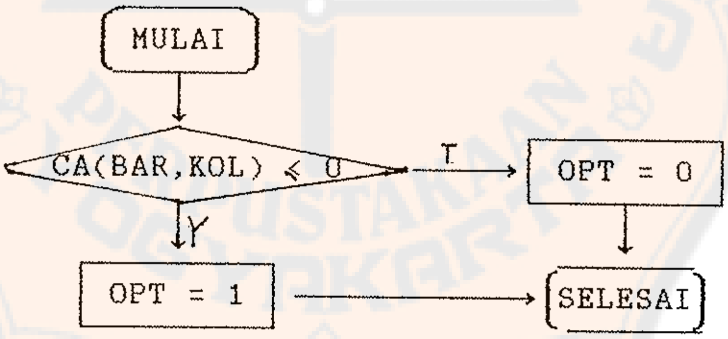
Flowchart Sub Camak :



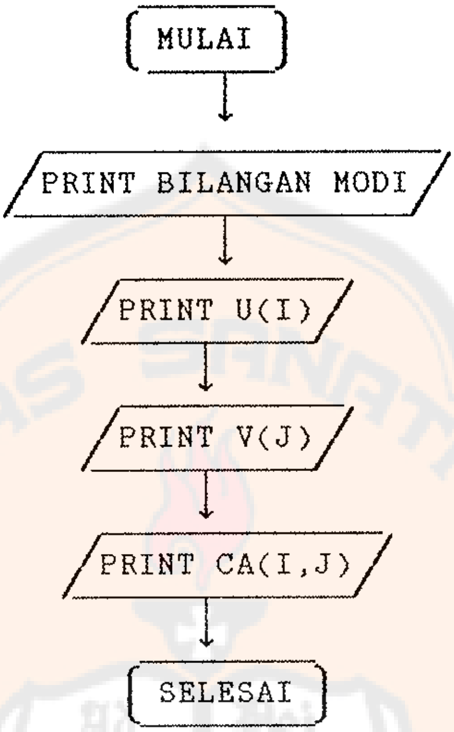
Flowchart Sub Makca



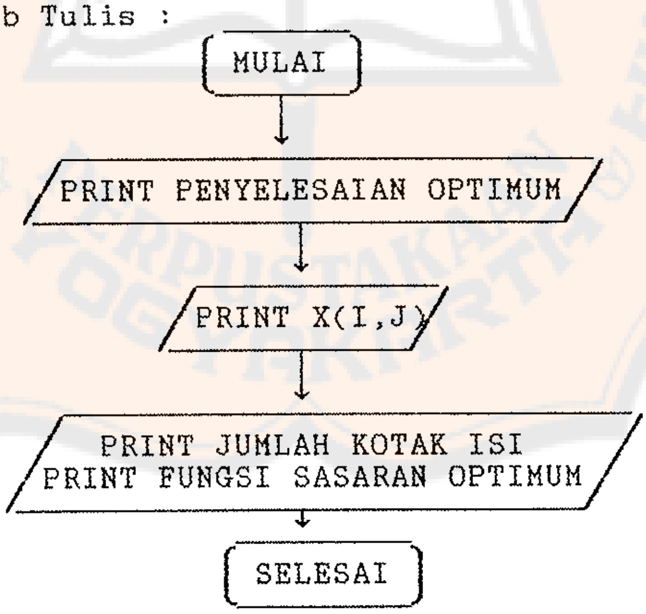
Flowchart Sub Optimal



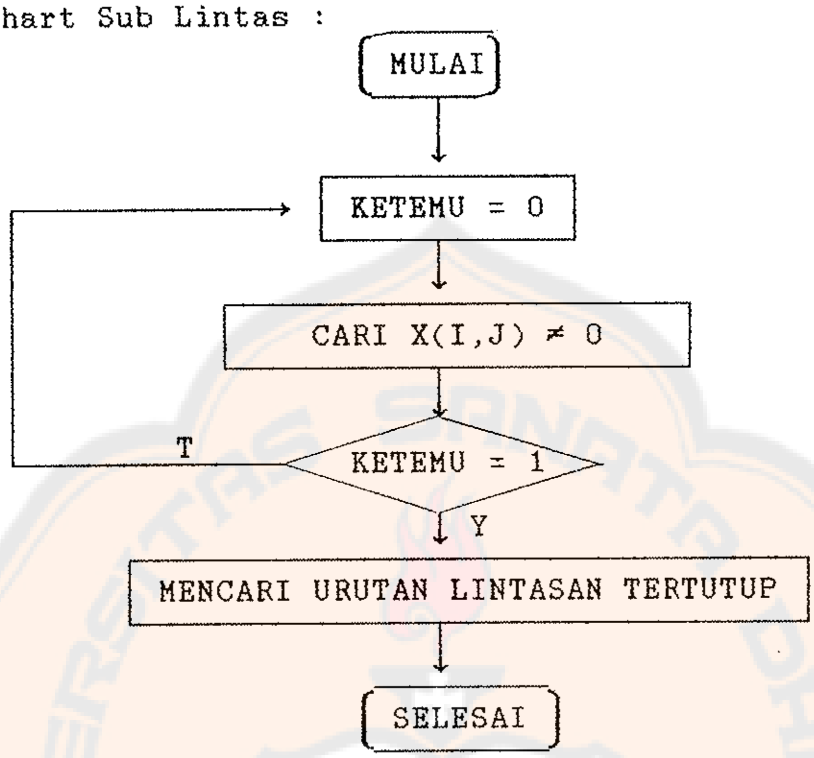
Flowchart Sub Cek 2 :



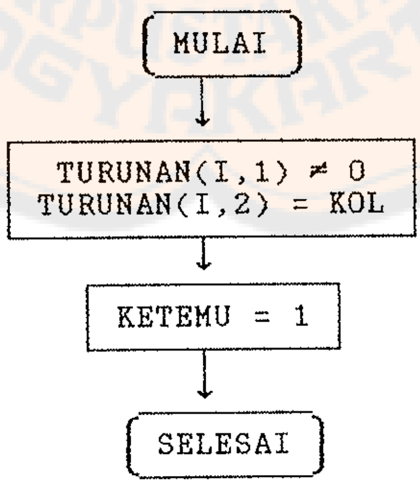
Flowchart Sub Tulis :



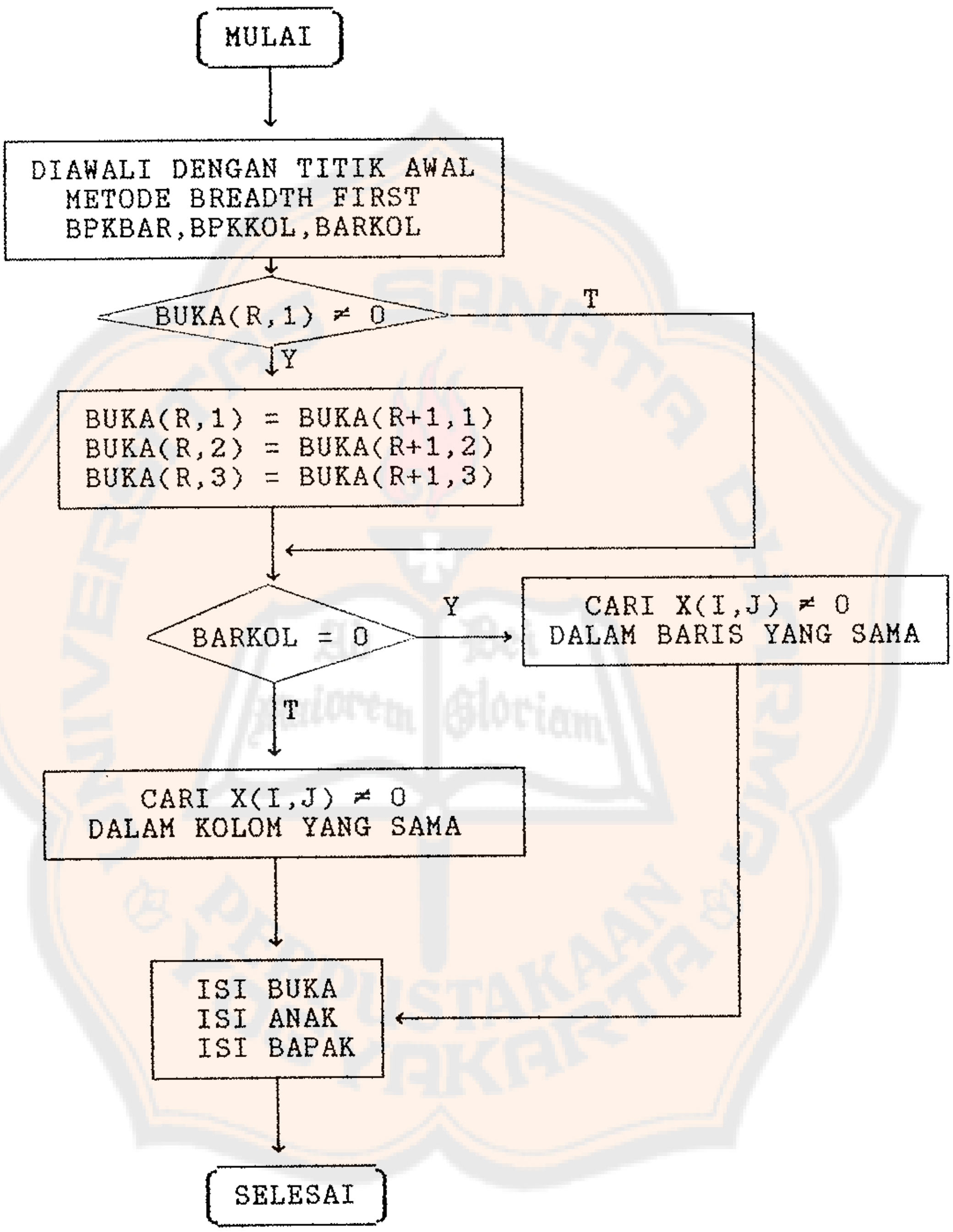
Flowchart Sub Lintas :



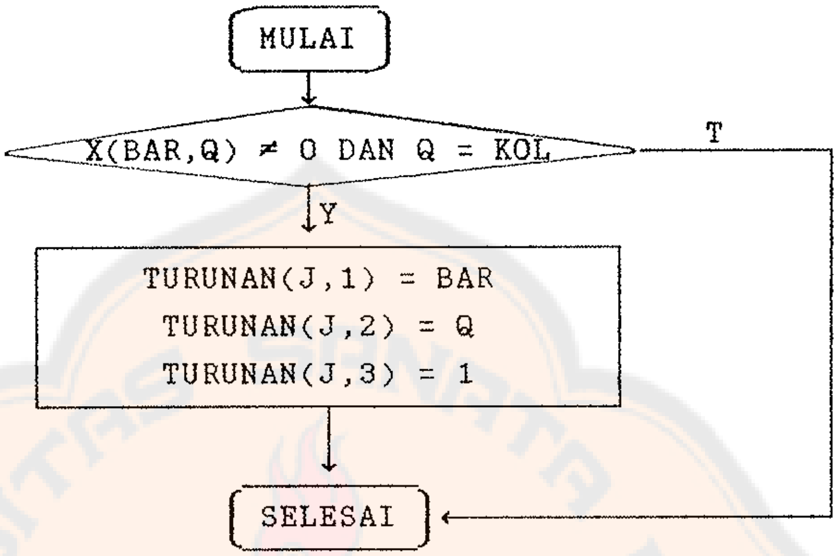
Flowchart Sub Cektemu :



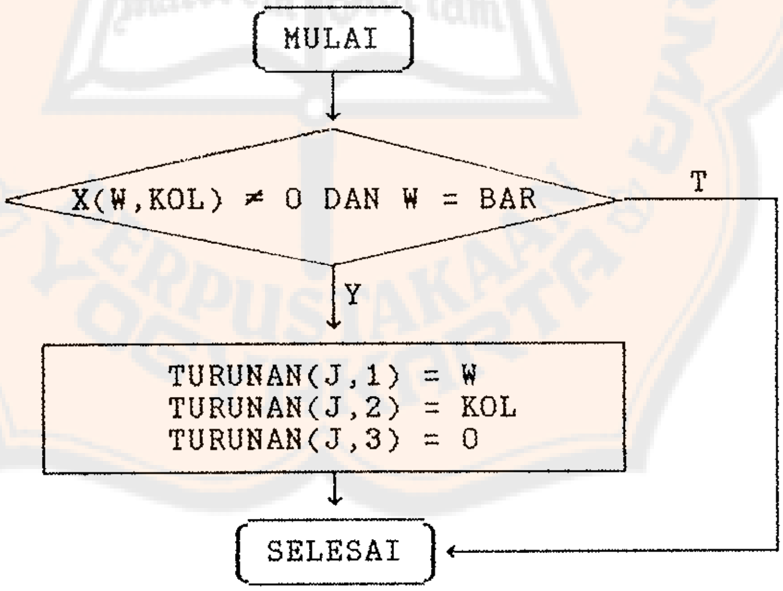
Flowchart Sub Ekspan :



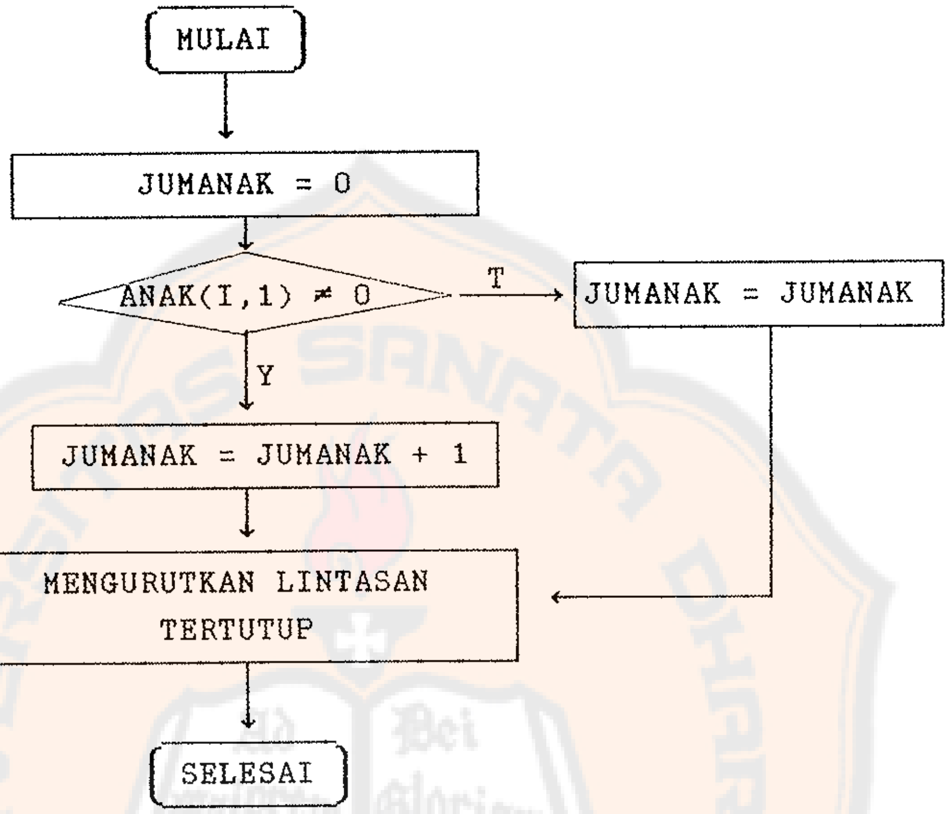
Flowchart Sub Cartembar :



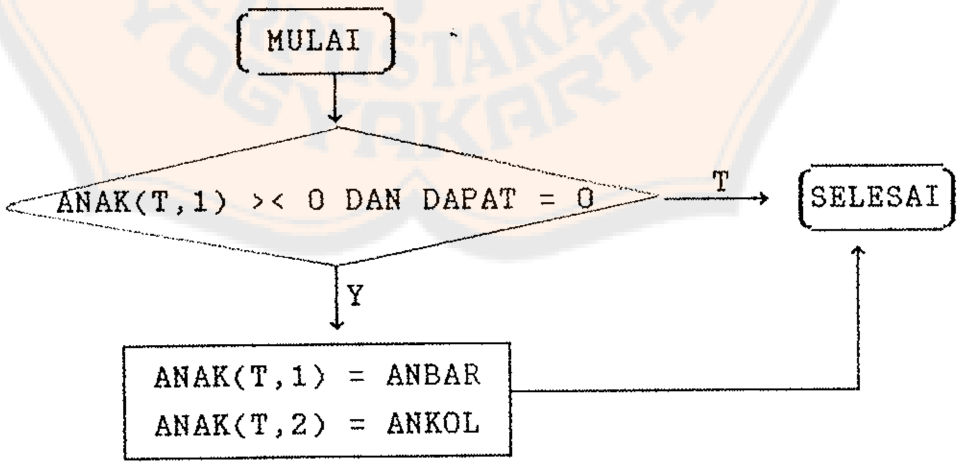
Flowchart Sub Cartemkol :



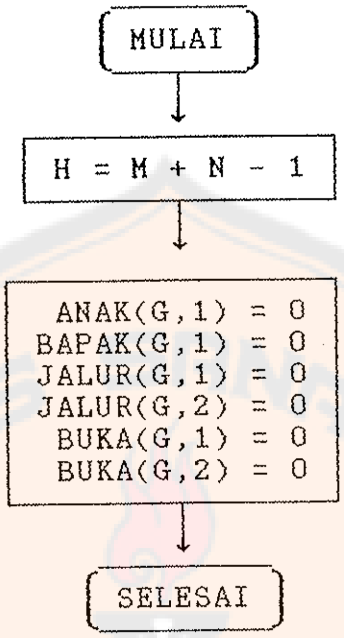
Flowchart Sub urut :



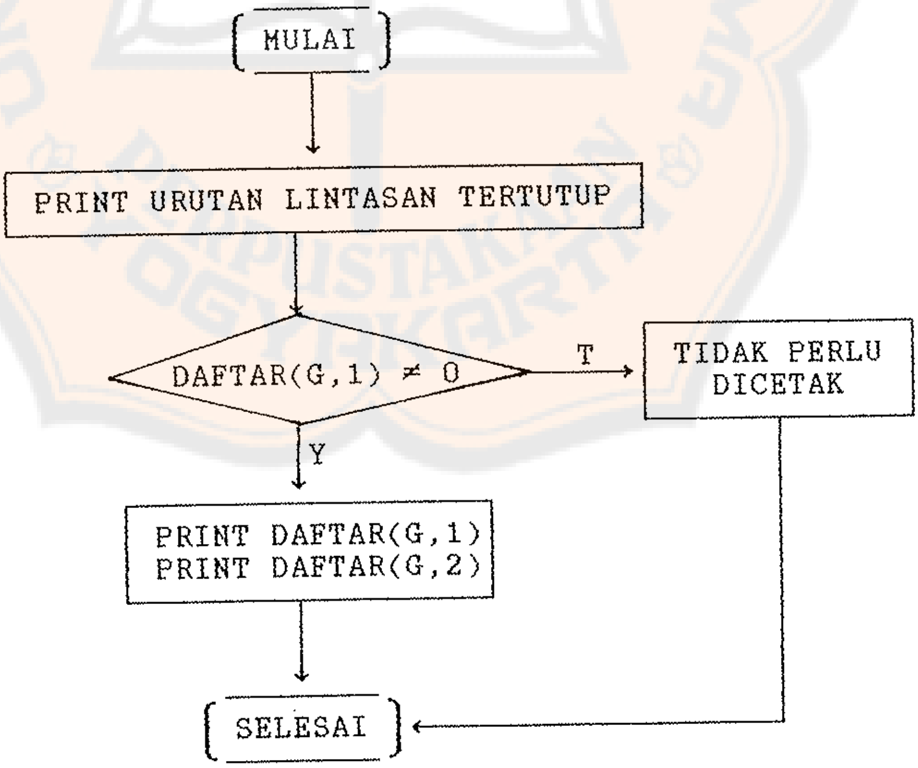
Flowchart Sub Carianak :



Flowchart Sub Kosong :



Flowchart Sub Tampil :



LAMPIRAN

```
REM ==============================
REM    "MASALAH ANGKUTAN"
REM ==============================

CLS
INPUT "BANYAK SUMBER = ",M
INPUT "BANYAK TUJUAN = ",N
H=M+N-1
DIM A(M+1),B(N+1),C(M+1,N+1),X(M+1,N+1)
DIM U(M+1),V(N+1),CA(M+1,N+1),BYS(W,2)
DIM TURUNAN(H,3),BUKA(H,3),ANAK(H,2),BAPAK(H,2),JALUR(H,2)
DIM TURUN(H,3),BUK(H,3)
CALL MASUKAN (M,N,S,Z,A(),B(),C(),P,L$)
CALL SETIMBANGKAN (M,N,S,Z,A(),B(),C())
CALL AWAL(M,N,A(),B(),X(),C())
CALL SASARAN(M,N,X(),C(),JUM,F,P,Z,L$)
CALL CEK1(A(),B(),X(),M,N,JUM,F)
CALL MODI(M,N,U(),V(),BUK(),TURUN(),C(),X(),PAKBAR,PAKKOL,CA())
CALL MAKCA(M,N,CA(),BAR,KOL)
CALL CEK2(M,N,U(),V(),CA())
CALL OPTIMAL(CA(),BAR,KOL,OPT)
WHILE OPT = 0
   BUKA(1,1) = BAR
   BUKA(1,2) = KOL
   BUKA(1,3) = 0
   CALL LINTAS(M,N,X(),BAR,KOL,JALUR(),BUKA(),TURUNAN(),
               ANAK(),BAPAK(),KETEMU)
   CALL URUT(M,N,BAPAK(),ANAK(),JALUR(),W,K,T,TND(),KOL)
   CALL TAMPIL(W,JALUR(),"JALUR",TND())
   CALL CARIMIN (W,JALUR(),TND(),X())
   CALL SASARAN(M,N,X(),C(),JUM,F,P,Z,L$)
   CALL TULIS(M,N,X(),JUM,F,OPT)
   CALL MODI(M,N,U(),V(),BUK(),TURUN(),C(),X(),PAKBAR,PAKKOL,CA())
   CALL MAKCA(M,N,CA(),BAR,KOL)
   CALL OPTIMAL(CA(),BAR,KOL,OPT)
   CALL CEK2(M,N,U(),V(),CA())
   CALL KOSONG(M,N,ANAK(),BAPAK(),JALUR(),BUKA())
WEND
CALL SASARAN(M,N,X(),C(),JUM,F,P,Z,L$)
CALL TULIS(M,N,X(),JUM,F,OPT)
1000 PRINT "GANTI SOAL YA"
END


SUB MASUKAN (M,N,S,Z,A(1),B(1),C(2),P,L$)
rem sub program ini untuk memasukkan semua data
rem yang harus diketahui oleh komputer
rem M = banyak baris, N = banyak kolom
rem S = total permintaan, Z = total penawaran
rem A() array untuk penawaran
rem B() array untuk permintaan
rem C() array untuk ongkos
Z=0
FOR I=1 TO M
    PRINT "BESAR PENAWARAN KE ";I;"=";
    INPUT A(I)
    Z = A(I) + Z
NEXT I
PRINT "PENAWARAN TOTAL = ";Z
```

```
S=0
FOR J=1 TO N
    PRINT "BESAR PERMINTAAN KE ";J;"=";
    INPUT B(J)
    S = B(J) + S
NEXT J
PRINT "PERMINTAAN TOTAL = ";S
IF S >< Z THEN
     PRINT "MASALAH ANGKUTAN TIDAK SETIMBANG"
END IF
INPUT "TEKAN TOMBOL ENTER";E

CLS
FOR I=1 TO M
  FOR J=1 TO N
    PRINT "ONGKOS ANGKUT DARI SUMBER";I;"KE TUJUAN";J;"=";
    INPUT C(I,J)
  NEXT J
NEXT I
INPUT "MEMAKSIMUMKAN (MAKS) / MEMINIMUMKAN (MIN) FUNGSI SASARAN";L$
IF L$ = "MAKS" OR L$ = "maks" THEN
  CALL MAKSIMUM(M,N,C(),P)
END IF
END SUB


SUB SETIMBANGKAN (M,N,S,Z,A(1),B(1),C(2))
rem sub program ini dijalankan jika masalah belum setimbang
rem M = banyak baris, N = banyak kolom
rem S = total permintaan, Z = total penawaran
rem A() array penawaran
rem B() array permintaan
rem C() array ongkos angkut
IF S><Z THEN
  IF S>Z THEN
    M=M+1
    A(M)=S-Z
    Z = Z + A(M)
    FOR J=1 TO N
      C(M,J)=0
    NEXT J
    ELSEIF S<Z THEN
    N=N+1
    B(N) = Z - S
    S = S + B(N)
    FOR I=1 TO M
      C(I,N)=0
    NEXT I
  END IF
END IF
END SUB


SUB MAKSIMUM(M,N,C(2),P)
P = C(1,1)
FOR I = 1 TO M
  FOR J = 1 TO N
    IF C(I,J) > P THEN
      P = C(I,J)
```

```
    END IF
  NEXT J
NEXT I
FOR I = 1 TO M
  FOR J = 1 TO N
    C(I,J) = P - C(I,J)
  NEXT J
NEXT I
END SUB


SUB CEK1 (A(1),B(1),X(2),M,N,JUM,F)
CLS
FOR I = 1 TO M
  PRINT "A(";I;")=";A(I),
NEXT I
PRINT
FOR J = 1 TO N
  PRINT "B(";J;")=";B(J),
NEXT  J
PRINT
PRINT "PENYELESAIAN AWALNYA ADALAH"
FOR I = 1 TO M
  FOR J = 1 TO N
    PRINT "X(";I;",";J;")=";X(I,J),
  NEXT J
NEXT I
PRINT
PRINT "JUMLAH KOTAK ISI = ";JUM
H=M+N-1
IF JUM<>H THEN
  PRINT "PENYELESAIAN MEROSOT"
  GOTO 1000
  ELSE
    PRINT "PENYELESAIAN TIDAK MEROSOT"
END IF
PRINT "FUNGSI SASARAN  = ";F
INPUT "ENTER";E
END SUB


SUB AWAL(M,N,A(1),B(1),X(2),C(2))
rem sub program ini untuk menentukan penyelesaian
rem layak basis awal dengan menggunakan metode Sudut Barat Laut
rem M = banyak baris, N = banyak kolom
rem A() array penawaran, B() array permintaan
rem X() array banyak barang yang harus diangkut
rem C() array ongkos angkut barang
FOR I = 1 TO M
  AP(I) = A(I)
NEXT I
FOR J = 1 TO N
  BS(J) = B(J)
NEXT J
I=1
WHILE I <= M
  J=1
  WHILE J <= N
    X(I,J) = FNMINI(AP(I),BS(J))
```

```
     AP(I) = AP(I) - X(I,J)
     BS(J) = BS(J) - X(I,J)
   J=J+1
   WEND
 I=I+1
 WEND
 END SUB

 DEF FNMINI(X,Y)
   IF X > Y THEN
     FNMINI = Y
   ELSE
     FNMINI = X
   END IF
 END DEF


 SUB SASARAN (M,N,X(2),C(2),JUM,F,P,Z,L$)
 JUM=0
 F=0
 FOR I=1 TO M
   FOR J=1 TO N
     IF X(I,J) >< 0 THEN
       JUM=JUM+1
     END IF
     IF X(I,J) = -1 THEN
       FA = 0
     ELSE
       FA = C(I,J) * X(I,J)
     END IF
     F = F + FA
   NEXT J
 NEXT I
 IF L$ = "MAKS" THEN
   F = (P * Z) - F
 END IF
 END SUB

 SUB MODI(M,N,U(1),V(1),BUK(2),TURUN(2),C(2),X(2),PAKBAR,PAKKOL,CA(2))
 U(1) = 0
 P = 1
 FOR J = 1 TO N
   IF X(1,J) <> 0 THEN
     BUK(P,1) = 1
     BUK(P,2) = J
     BUK(P,3) = 0
     V(J) = C(1,J) - U(1)
     P = P + 1
   END IF
 NEXT J
 TEMU = 0
 WHILE TEMU = 0
   CALL PINDAH(U(),V(),BUK(),PAKBAR,PAKKOL,TURUN(),X(),M,N,C())
   CALL TEMU(BUK(),TEMU)
 WEND
 CALL CAMAK(M,N,X(),CA(),U(),V(),C())
 END SUB
```

```
SUB PINDAH(U(1),V(1),BUK(2),PAKBAR,PAKKOL,TURUN(2),X(2),M,N,C(2))
PAKBAR = BUK(1,1)
PAKKOL = BUK(1,2)
RISKOL = BUK(1,3)
R = 1
WHILE BUK(R,1) <> 0
   BUK(R,1) = BUK(R+1,1)
   BUK(R,2) = BUK(R+1,2)
   BUK(R,3) = BUK(R+1,3)
   R = R + 1
WEND
IF RISKOL <> 0 THEN
   O = TURUN(1,1)
   CALL CABAR(N,TURUN(),X(),PAKBAR,PAKKOL)
   T = 1
   WHILE TURUN(T,1) <> 0
      V(TURUN(T,2)) = C(TURUN(T,1),TURUN(T,2)) - U(TURUN(T,1))
      T = T + 1
   WEND
ELSE
   O = TURUN(1,1)
   CALL CAKOL(M,TURUN(),X(),PAKBAR,PAKKOL)
   D = 1
   WHILE TURUN(D,1) <> 0
      U(TURUN(D,1)) = C(TURUN(D,1),TURUN(D,2)) - V(TURUN(D,2))
      D = D + 1
   WEND
END IF

R = 1
WHILE BUK(R,1) <> 0
   R = R + 1
WEND
Y = 1
WHILE TURUN(Y,1) <> 0
   BUK(R,1) = TURUN(Y,1)
   BUK(R,2) = TURUN(Y,2)
   BUK(R,3) = TURUN(Y,3)
   R = R + 1
   Y = Y + 1
WEND
END SUB

SUB CABAR(N,TURUN(2),X(2),PAKBAR,PAKKOL)
FOR J = 1 TO N
   TURUN(J,1) = 0
   TURUN(J,2) = 0
NEXT J
J = 1
FOR Q = 1 TO N
   IF X(PAKBAR,Q) <> 0 AND Q <> PAKKOL THEN
      TURUN(J,1) = PAKBAR
      TURUN(J,2) = Q
      TURUN(J,3) = 0
      J = J + 1
   END IF
NEXT Q
END SUB
```

```
SUB CAKOL(M,TURUN(2),X(2),PAKBAR,PAKKOL)
FOR I = 1 TO M
   TURUN(I,1) = 0
   TURUN(I,2) = 0
NEXT I
I = 1
FOR Q = 1 TO M
   IF X(Q,PAKKOL) <> 0 AND Q <> PAKBAR THEN
      TURUN(I,1) = Q
      TURUN(I,2) = PAKKOL
      TURUN(I,3) = 1
      I = I + 1
   END IF
NEXT Q
END SUB

SUB TEMU(BUK(2),TEMU)
IF BUK(1,1) = 0 THEN
      TEMU = 1
END IF
END SUB

SUB CAMAK(M,N,X(2),CA(2),U(1),V(1),C(2))
REM HITUNG Cij'
FOR I=1 TO M
    FOR J = 1 TO N
       IF X(I,J) = 0 THEN
            CA(I,J) = U(I) + V(J) - C(I,J)
       ELSE
            CA(I,J) = 0
       END IF
     NEXT J
NEXT I
END SUB

SUB CEK2 (M,N,U(1),V(1),CA(2))
CLS
PRINT "BILANGAN MODINYA ADALAH"
FOR I=1 TO M
   PRINT "U(";I;")=";U(I),
NEXT I
PRINT
FOR J=1 TO N
   PRINT "V(";J;")=";V(J),
NEXT J
PRINT
FOR I=1 TO M
   FOR J=1 TO N
      PRINT "CA(";I;J;")=";CA(I,J),
   NEXT J
NEXT I
INPUT "ENTER";E
END SUB
```

```
SUB MAKCA(M,N,CA(2),BAR,KOL)
rem sub program mencari sel lintasan awal
rem M = banyak baris, N = banyak kolom
rem CA() array ongkos kesempatan
rem (bar,kol) sel lintasan tertutup awal
MAK = CA(1,1)
FOR I = 1 TO M
    FOR J = 1 TO N
       IF CA(I,J) > MAK THEN
          MAK = CA(I,J)
          BAR = I
          KOL = J
        END IF
    NEXT J
NEXT I
END SUB


SUB OPTIMAL(CA(2),BAR,KOL,OPT)
rem opt=1 berarti masalah sudah optimum
rem CA() array ongkos kesempatan
rem (bar,kol) sel awal lintasan tertutup
IF CA(BAR,KOL) <= 0 THEN
   OPT = 1
ELSE
   OPT = 0
END IF
END SUB


SUB CARIMIN (W,JALUR(2),TND(1),X(2))

rem sub program untuk mencari X(i,j) minimum dalam lintasan tertutup
rem jalur() array urutan lintasan tertutup
rem tnd() array tanda + , - lintasan tertutup
rem X() array jumlah barang yang harus diangkut

B = X(JALUR(1,1),JALUR(1,2))
IF B = -1 THEN
   BS(1,1) = JALUR(1,1)
   BS(1,2) = JALUR(1,2)
END IF
T = 1
FOR D = 3  TO  W STEP 2
    IF X(JALUR(D,1),JALUR(D,2)) < B THEN
      B = X(JALUR(D,1),JALUR(D,2))
      YS(1,1) = JALUR(D,1)
      YS(1,2) = JALUR(D,2)
    ELSEIF X(JALUR(D,1),JALUR(D,2)) = B THEN
      BYS(T,1) = JALUR(D,1)
      BYS(T,2) = JALUR(D,2)
      K = T
      T = T + 1
    END IF
NEXT D
IF B = -1 THEN
    X(JALUR(W,1),JALUR(W,2)) = -1
    X(BYS(T,1),BYS(T,2)) = 0
ELSE
```

```
   FOR D = 1 TO W
    IF X(JALUR(D,1),JALUR(D,2)) = -1 THEN
     X(JALUR(D,1),JALUR(D,2)) = B
    ELSE
     X(JALUR(D,1),JALUR(D,2)) = X(JALUR(D,1),JALUR(D,2)) + (TND(D) * B)
    END IF
   NEXT D
END IF
FOR D = 1 TO K
  X(BYS(D,1),BYS(D,2)) = -1
NEXT D
IF X(JALUR(1,1),JALUR(1,2)) = -1 THEN
   X(BS(1,1),BS(1,2)) = 0
END IF
X(YS(1,1),YS(1,2)) = 0
END SUB


SUB TULIS (M,N,X(2),JUM,F,OPT)
CLS
IF OPT=1 THEN
  PRINT "PENYELESAIAN OPTIMUMNYA ADALAH"
END IF
FOR I = 1 TO M
   FOR J = 1 TO N
      PRINT "X(";I;",";J;")=";X(I,J),
   NEXT J
NEXT I
PRINT
PRINT "JUMLAH KOTAK ISI =";JUM
IF OPT = 1 THEN
  PRINT "FUNGSI SASARAN OPTIMUM =";F
ELSE
  PRINT "FUNGSI SASARAN =";F
END IF
INPUT "ENTER";E
END SUB


SUB LINTAS(M,N,X(2),BAR,KOL,JALUR(2),BUKA(2),TURUNAN(2),
           ANAK(2),BAPAK(2),KETEMU)
rem sub program untuk mencari lintasan tertutup
rem (bar,kol) sel awal lintasan
rem jalur() array untuk menampung sel-sel lintasan tertutup
rem M = jumlah baris;  N = jumlah kolom
rem X() array banyak barang yang harus diangkut
KETEMU = 0
WHILE KETEMU = 0
   CALL EKSPAN(BUKA(),BAPAK(),ANAK(),TURUNAN(),M,N,X())
   CALL CEKTEMU(TURUNAN(),KETEMU,KOL)
WEND
END SUB


REM =================================
SUB CARTEMBAR(M,N,X(2),BPKBAR,BPKKOL,TURUNAN(2))
rem sub program untuk mencari teman x(i,j) > 0
rem M = banyak baris, N = banyak kolom
rem X() array untuk banyak barang yang harus diangkut
```

```
rem turunan() array untuk menyimpan x(i,j) > 0
    dalam baris atau kolom
FOR J=1 TO N
  TURUNAN(J,1) = 0
  TURUNAN(J,2) = 0
  TURUNAN(J,3) = 0
NEXT J
J=1
FOR Q=1 TO N
    IF X(BPKBAR,Q) <> 0 AND Q <> BPKKOL THEN
        TURUNAN(J,1) = BPKBAR
        TURUNAN(J,2) = Q
        TURUNAN(J,3) = 1
        J=J+1
    END IF
NEXT Q
END SUB


REM ==================================================
SUB CARTEMKOL(M,N,X(2),BPKBAR,BPKKOL,TURUNAN(2))
rem sub program mencari teman x(i,j) > 0 dalam kolom yang sama
rem M = banyak sumber, N = banyak tujuan
rem X() array untuk menyimpan banyak barang yang harus diangkut
rem (bar,kol) sel awal lintasan tertutup
rem turunan() array untuk menyimpan x(i,j) > 0
    dalam baris / kolom
FOR J=1 TO M
  TURUNAN(J,1) = 0
  TURUNAN(J,2) = 0
  TURUNAN(J,3) = 0
NEXT J
J=1
FOR W=1 TO M
    IF (X(W,BPKKOL) <> 0 AND W <> BPKBAR) THEN
        TURUNAN(J,1) = W
        TURUNAN(J,2) = BPKKOL
        TURUNAN(J,3) = 0
        J=J+1
    END IF
NEXT W
END SUB

REM ==================================================
SUB  EKSPAN(BUKA(2),BAPAK(2),ANAK(2),TURUNAN(2),M,N,X(2))
rem mengekspan x(i,j) > 0 yamg diperoleh dari array open
rem ke array bapak dan anak
rem buka() array yang diekspan dari turunan()
rem bapak() array yang diekspan dari buka()
rem anak() array yang diekspan dari buka()
rem barkol untuk memanggil cartembar atau cartemkol
rem turunan() array untuk menyimpan x(i,j) > 0 dalam baris/kolom
BPKBAR = BUKA(1,1)
BPKKOL = BUKA(1,2)
BARKOL = BUKA(1,3)
```

```
R = 1
WHILE BUKA(R,1) <> 0
    BUKA(R,1) = BUKA(R+1,1)
    BUKA(R,2) = BUKA(R+1,2)
    BUKA(R,3) = BUKA(R+1,3)
    R=R+1
WEND
IF BARKOL = 0 THEN
   O = TURUNAN(1,1)
   CALL CARTEMBAR(M,N,X(),BPKBAR,BPKKOL,TURUNAN())
ELSE
   O = TURUNAN(1,1)
   CALL CARTEMKOL(M,N,X(),BPKBAR,BPKKOL,TURUNAN())
END IF

REM ISI BUKA --------------------------------
R = 1
WHILE BUKA(R,1) <> 0
    R = R + 1
WEND
J = 1
WHILE TURUNAN(J,1) <> 0
    BUKA(R,1) = TURUNAN(J,1)
    BUKA(R,2) = TURUNAN(J,2)
    BUKA(R,3) = TURUNAN(J,3)
    R = R + 1
    J = J + 1
WEND

REM ISI-ANAK ------------------------------
R =1
WHILE ANAK(R,1) <> 0
   R=R+1
WEND
J=1
WHILE TURUNAN(J,1) <> 0
    ANAK(R,1) = TURUNAN(J,1)
    ANAK(R,2) = TURUNAN(J,2)
    R=R+1
    J=J+1
WEND

REM ISI BAPAK ----------------------------
R=1
WHILE BAPAK(R,1) <> 0
    R=R+1
WEND
K = 1
WHILE K <= J-1
   BAPAK(R,1) = BPKBAR
   BAPAK(R,2) = BPKKOL
   R = R + 1
   K = K + 1
WEND
END SUB
```

```
SUB CEKTEMU(TURUNAN(2),KETEMU,KOL)
rem sub program ini untuk mengetahui apakah turunan yang diperoleh
rem sudah sama dengan kolom sel awal lintasan
rem turunan() array untuk menyimpan x(i,j)>0 dalam baris / kolom
rem ketemu = 1 berarti pencarian x(i,j)>0 selesai
rem (bar,kol) sel awal lintasan tertutup
I=1
WHILE TURUNAN(I,1) <> 0
   IF TURUNAN(I,2) = KOL THEN
       KETEMU = 1
   END IF
   I= I +1
WEND
END SUB


SUB URUT(M,N,BAPAK(2),ANAK(2),JALUR(2),W,K,T,TND(1),KOL)
rem sub program ini untuk mencari urutan lintasan tertutup
rem bapak() array untuk mengekspan anak()
rem anak() array untuk mengekspan buka()
rem jalur() array untuk menyimpan urutan lintasan tertutup
H = M + N - 1
JUMANAK = 0
FOR I=1 TO H
  IF ANAK(I,1) <> 0 THEN
   JUMANAK = JUMANAK + 1
  END IF
NEXT I
J = JUMANAK
ROOTBAR = BAPAK(1,1)
ROOTKOL = BAPAK(1,2)
FOR W = 1 TO J
  IF ANAK(W,2) = KOL THEN
    JALUR(1,1) = ANAK(W,1)
    JALUR(1,2) = ANAK(W,2)
    JALUR(2,1) = BAPAK(W,1)
    JALUR(2,2) = BAPAK(W,2)
  END IF
NEXT W
T = J - 1
I=3
WHILE JALUR(I-1,2) <> ROOTKOL
    CALL CARIANAK(W,T,JALUR(I-1,1),JALUR(I-1,2),K,JALUR(),ANAK())
    JALUR(I,1) = BAPAK(K,1)
    JALUR(I,2) = BAPAK(K,2)
    W = I
    I = I + 1
WEND
FOR S = 1 TO W
  IF (S MOD 2 = 0) THEN
    TND(S) = 1
  ELSE
    TND(S) = -1
  END IF
NEXT S
END SUB
```

```
REM ==================================

SUB CARIANAK(W,T,ANBAR,ANKOL,K,JALUR(2),ANAK(2))

rem sub program ini untuk apakah pencarian urutan
rem lintasan tertutup sudah selesai
rem anakbar = jalur(I,1), anakkol = jalur(I,2)
rem K untuk meneruskan nomer array bapak()

S = T
DAPAT = 0
WHILE ANAK(S,1) <> 0 AND DAPAT = 0 AND S > 0
  IF ANAK(S,1) = ANBAR AND ANAK(S,2) = ANKOL THEN
    DAPAT = 1
    K = S
  END IF
S = S - 1
WEND

END SUB


SUB TAMPIL(W,DAFTAR(2),TEKS$,TND(2))
CLS
PRINT "URUTAN LINTASAN TERTUTUP"
G=1
WHILE DAFTAR(G,1) <> 0
  PRINT TEKS$;"(";G;",1) =";DAFTAR(G,1),
  PRINT TEKS$;"(";G;",2) =";DAFTAR(G,2)
G = G + 1
WEND
INPUT "ENTER";E
END SUB

SUB KOSONG(M,N,ANAK(2),BAPAK(2),JALUR(2),BUKA(2))
H = M + N - 1
FOR G = 1 TO H
  ANAK(G,1) = 0
  ANAK(G,2) = 0
  BAPAK(G,1) = 0
  BAPAK(G,2) = 0
  JALUR(G,1) = 0
  JALUR(G,2) = 0
  BUKA(G,1) = 0
  BUKA(G,2) = 0
NEXT G
END SUB
```