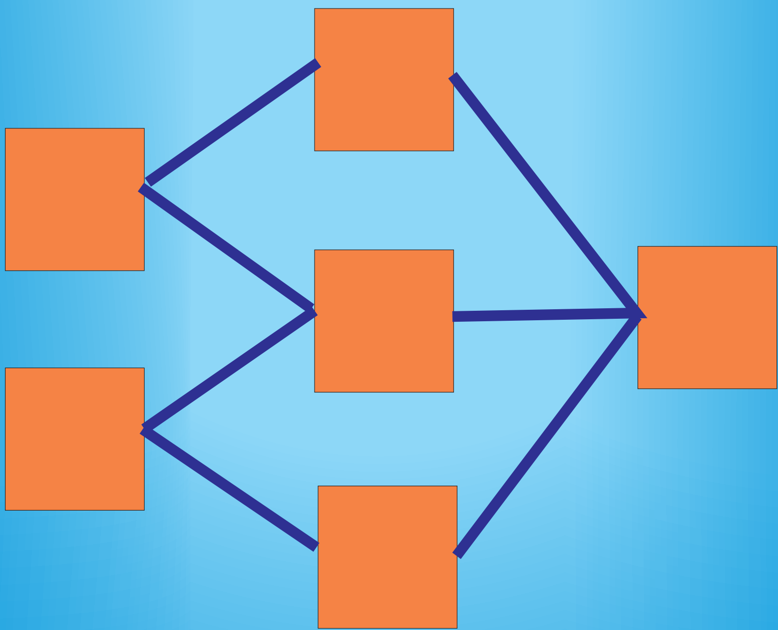
Buku Referensi

Metode Kuantitatif

edisi pertama



Sigit Nugroho, Ph.D.

UNIB Press

METODE KUANTITATIF

Metode Kuantitatif

Sigit Nugroho, Ph.D.
Universitas Bengkulu

UNIB Press
Bengkulu
2008

METODE KUANTITATIF
Sigit Nugroho, Ph.D.
ISBN : 978-979-9431-32-5 216hal.
Cetakan Pertama. Edisi 1. 2008.

Penyeleksi Naskah : Fachri Faisal

Editor : Jose Rizal
Desain Sampul : Ratna Astuti Nugrahaeni

Diterbitkan pertama kali oleh **UNIB Press**, Jalan WR Supratman, Bengkulu.

Kata Pengantar

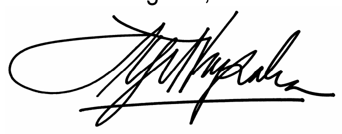
Buku yang berjudul "**Metode Kuantitatif**" ini diperlukan sebagai landasan, pedoman atau rujukan bagi para mahasiswa atau siapa saja yang ingin mempelajari dasar-dasar pengambilan keputusan dengan baik, mudah dan benar.

Materi buku ini biasanya disajikan untuk mata kuliah Metode Kuantitatif di berbagai jurusan dengan bobot 3 - 4 SKS. Materi untuk bagian pertama diawali Pengantar Model Pengambilan Keputusan Organisasional, dilanjutkan dengan permasalahan Pemrograman Linier, Pemrograman Dinamis, dan Pemrograman Integer. Materi lainnya mencakup *Branch-and-Bound Technique*, Model Inventori Deterministik, dan Masalah Pengurutan. Beberapa bagian terakhir mencakup aspek probabilistik, seperti: PERT, Teori Antrian, dan Teori Pengambilan Keputusan.

Mungkin tulisan ini jauh dari apa yang diharapkan oleh para pengguna. Oleh karena itu perlu saran dan kritik yang membangun dari para pengguna.

Ucapan terima kasih penulis sampaikan sebelum dan sesudahnya kepada mereka yang memberikan masukan baik berupa saran atau kritik yang membangun. Penulis sampaikan kepada istri, *Mucharromah Ph.D.*, dan anak-anak, *Shofa Ulfiyati Nugrahaeni* dan *Ratna Astuti Nugrahaeni*, yang telah dengan penuh pengertian dan sabar memberikan motivasi serta dorongan guna penyelesaian tulisan ini. Kepada rekan-rekan sejawat yang telah memberikan masukan-masukan untuk usaha penerbitan, penulis ucapkan ribuan terima kasih. Kepada semua pengguna, juga diucapkan terima kasih dan semoga mendapatkan manfaat dari karya tulis saya ini.

Bengkulu, 20 Juli 2008



Sigit Nugroho, Ph.D.

Oentoek:

Mucharromah Nugroho, Ph.D.,
Shofa Ulfiyati Nugrahaeni, dan
Ratna Astuti Nugrahaeni

Daftar Isi

Pengantar Model Pengambilan Keputusan Organisasional

Model merupakan representasi dalam bentuk sederhana dari aspek yang relevan akan sistem atau proses nyata.

1. **Model Fisik** merupakan representasi obyek nyata, masalah, atau sistem dalam beberapa bentuk fisik.
   a **Model Ikonik** dapat berupa replika dengan ukuran yang proporsional terhadap aslinya.
   b **Model Analog** dibangun dengan menggunakan hubungan fisik dua dimensi untuk menggambarkan hubungan yang sesuai dalam bentuk lain.
2. **Model Simbolik** merupakan model abstrak; biasanya digunakan bila tak dapat digunakan model fisik
   a **Model Matematis** biasanya menggunakan notasi matematis untuk menunjukkan hubungan antar variabel dalam suatu sistem
   b **Model Verbal** merupakan versi tulisan atau ucapan akan pikiran individual.

Tipe Model Matematis

1. Model Deskriptif dan Normatif. **Model Deskriptif** menggambarkan sesuatu seperti apa adanya; dan **Model Normatif** mengindikasikan bagaimana seharusnya.

Model Deskriptif

- i) Mendifinisikan situasi dengan lebih jelas
- ii) Mengidentifikasi kemungkinan adanya perubahan akan suatu hal
- iii) Menginvestigasi segala konsekuensi berbagai pilihan keputusan
- iv) Tidak mengidentifikasi pilihan terbaik dari semua pilihan yang mungkin

v) Menyediakan kerangka kerja untuk analisis dalam membantu pengambil keputusan dalam penentuan pilihan

Model Normatif

vi) Mengidentifikasi pilihan terbaik yang ada berdasarkan beberapa kriteria keputusan tertentu.

vii) Merupakan model keputusan yang optimum pada kondisi tertentu

2. Model Deterministik dan Probabilistik. **Model Deterministik** menggunakan kriteria tidak adanya suatu kebetulan atau sesuatu yang berhubungan dengan peluang, sehigga model deterministik digunakan pada kondisi yang sudah ditentukan. **Model Probabilistik** dipakai apabila ada ketergantungan dari sesuatu yang tidak diketahui atau adanya suatu peluang akan kejadian tersebut.

Model-model Operasional

1. **Model Teori Keputusan** (*Decision-theory Models*). Tujuan utama dari teori keputusan adalah membuat keputusan dari sudut ilmiah ketimbang dari sudut seni. Tergolong model normatif dan menggunakan variabel probabilistik.
2. **Model Alokasi** (*Allocation Models*). Tujuan utama adalah menentukan alokasi terbaik dari sumber terbatas diantara berbagai aktifitas. Tergolong model normatif dan menggunakan variabel deterministik.
3. **Model Inventori** (*Inventory Models*). Tujuan utamanya adalah menentukan kapan dan berapa banyak barang atau sesuatu harus "diadakan" bila dihubungkan dengan persediaannya. Tergolong model normatif, namun variabelnya dapat deterministik ataupun probabilistik, tergantung apakah keputusan yang diambil pada kondisi tertentu atau kondisi yang beresiko.
4. **Model Antrian** (*Queuing Models*). Tujuan utamanya adalah mempelajari banyaknya pelanggan dalam antrian atau di dalam sistem, serta waktu yang diperlukan oleh seorang pelanggan berada dalam antrian atau di dalam sistem dari berbagai

macam tipe antrian. Tergolong model deskriptif dan memiliki variabel probabilistik.

5. **Model Simulasi** (*Simulation Models*). Banyak permasalahan yang tak dapat digolongkan dalam model normatif dan harus dipecahkan dengan cara lain. Simulasi mencakup pembuatan model matematis dari suatu sistem yang dipelajari dan penampilan penghitungan tahap demi tahap dalam model untuk meniru proses yang sesungguhnya. Beberapa tindakan perlu dilakukan hingga jawaban yang tepat, bukannya jawaban yang optimal, dicapai. Tergolong model deskriptif dan memiliki variabel probabilistik.
6. **Model Markov** (*Markov Models*). Tujuan utamanya adalah untuk mempelajari tingkah laku konsumen akan suatu produk. Konsumen mungkin ada yang loyal, mungkin juga tidak. Model ini tergolong deskriptif dan memiliki variabel probabilistik.
7. **Model Jaringan** (*Network Models*). Misalnya digunakan dalam mempelajari hubungan antara rencana kerja dan pekerjaan yang sudah diselesaikannya, serta waktu pelaksanaannya. Model ini bersifat normatif dan sifat alami variabelnya biasanya probabilistik.

Model	Model (Kegunaan)		Model (Tipe Variabel)	
	Deskrip tif	Norma tif	Determi-nistik	Proba bilistik
Teori Keputusan		X		X
Alokasi		X	X	
Inventori		X	X	
Antrian	X			X
Simulasi	X			X
Markov	X			X
Jaringan		X		X

Pemrograman Linier

Pemrograman Linier saat ini merupakan salah satu alat dalam riset operasi yang paling banyak digunakan. Hampir semua permasalahan tiap tahapan industri telah dirumuskan dengan sukses dalam pemrograman linier. Penerapan dalam industri minyak dan gas termasuk ekstraksi minyak kasar dari bumi, penentuan jadwal penyulingan, penentuan campuran produk optimal pada pembuatan premium, dan fase terakhir dimana produk yang dimurnikan dikirim ke pengguna. Aplikasi lain mencakup penggunaan pemrograman linier untuk optimasi campuran makanan ternak, pengiriman barang dari sejumlah pabrik ke sejumlah warehouse, penggunaan kayu di sebuah areal hutan, penjadwalan produksi dalam kebanyakan industri, jadwal penerbangan pesawat, alokasi dana untuk periklanan, penugasan pekerjaan ke pekerja, dan lain sebagainya. Dalam tiap kasus model pemrograman linier mencakup fungsi linier dari beberapa variabel yang harus dioptimasi (maksimisasi atau minimisasi) dengan memperhatikan semua kendala linier dan batasan bahwa variabel yang digunakan tidak negatif.

Kita telah lihat bahwa dalam fungsi persamaan pemrograman dinamis (*dynamic programming*) diformulasikan berbeda-beda dari satu maslaah ke masalah yang lain. Sebagai akibatnya, program komputer yang berbeda diperlukan untuk memecahkan tiap masalah itu. Pemrograman linier tidaklah demikian. Tiap model program linier memiliki bentuk yang sama. Sekali masalah telah dibuat dalam bentuk yang khas, maka prosedsur yang sama (biasanya prosedur simplex) dapat digunakan untuk menentukan jawaban optimal.

Formulasi Model Pemrograman Linier

Tiap contoh masalah dalam seksi ini harus diformulasikan sebagai model pemrograman linier. Tujuan : menggambarkan formulasi model pemrograman linier (PL).
Bentuk umum tiap model adalah

Maksimumkan (minimumkan) $z = c_1x_1 + c_2x_2 + ... + c_kx_k$

dengan kendala $a_{11}x_1 + a_{12}x_2 + ... + a_{1k}x_{1k}$ (*) b_1

$$a_{21}x_1 + a_{22}x_2 + ... + a_{2k}x_{1k} \ (*) \ b_2$$

....

$$a_{m1}x_1 + a_{m2}x_2 + ... + a_{mk}x_{1k} \ (*) \ b_m$$

untuk semua $x_i \geq 0$ dimana c_j adalah koeffisien biaya dari x_j, dengan x_j merupakan variabel yang tak diketahui nilainya, serta a_{ij} merupakan suatu konstanta, c_j merupakan konstanta yang diketahui, serta tanda (*) dapat berarti $\geq$, $=$, atau $\leq$ untuk tiap kendala.

DIET. Dengan semakin menaiknya harga makanan, setiap rumah tangga menghadapi tugas yang sulit dalam menyediakan keluarganya dengan diet yang seimbang dengan tetap mempertimbangkan biaya yang realistik. Nutrisi sudah barang tentu, hanyalah suatu faktor yang perlu dipertimbangkan dalam perencanaan menu mingguan. Misalkan kita ingin hidangkan sejumlah hidangan dengan menggunakan 6 jenis sayuran berikut untuk memenuhi menu minggu ini guna meminimumkan biaya namun masih memenuhi kebutuhan gizi. Kita asumsikan bahwa sayuran hanya akan dapat sebagian dari kebutuhan harian minimum. Informasi lebih lanjut dapat dilihat pada tabel berikut

Sayuran	**Kandungan per sajian**					**Biaya per sajian**
	Zat Besi	**Fosfor**	**Vit A**	**Vit C**	**Niasin**	**Rp**
	mg	***mg***	***USP***	***mg***	***mg***	
Kacang Panjang	0.45	10	415	8	0.30	500
Wortel	0.45	28	9065	3	0.35	500
Brokoli	1.05	50	2550	53	0.60	800
Kobis	0.40	25	75	27	0.15	200
Bit	0.50	22	15	5	0.25	600
Kentang	0.50	75	235	8	0.80	300
Minimum kebutuhan mingguan	6.00	325	17500	245	5.00	

Sebagai tambahan, kubis tak dapat dihidangkan lebih dari dua kali dalam seminggu, dan sayuran lain tak dapat dihidangkan lebih dari empat kali dalam seminggu. Sebanyak 14 hidangan sayur diperlukan dalam seminggu. Berapa kali tiap sayur harus disajikan minggu mendatang sehingga biayanya minimum dan memenuhi kebutuhan minimum nutrisi dan rasa ?

Televisi. Sebuah perusahaan televisi menaruh perhatian tentang banyaknya satuan dari tiga tipe televisi portabel harus diproduksi untuk memaksimumkan keuntungan. Berdasarkan permintaan masa lalu, sebanyak berturut-turut minimum 200, 250, dan 100 satuan televisi tipe I, II, dan III dibutuhkan atau diminta. Pabrik tersebut menyediakan waktu sebanyak 1000 satuan waktu dan 1000 satuan bahan, untuk pembuatan ketiga tipe televisi tersebut. Data lain yang mungkin diperlukan adalah

	Waktu	Bahan	Harga
TV I	1	2	10
TV II	1.5	1.2	14
TV III	4	1	12

Produksi Truk. Sebuah perusahaan truk dapat memproduksi 5 tipe truk dalam satu bulan. Banyaknya tiap tipe truk yang diproduksi ditentukan dengan kapasitas dari berbagai departemen yang ada dalam pabrik tersebut

- Departemen pencetakan logam tak dapat menangani lebih dari ekuivalen 10000 truk tipe I
- Departemen pemasangan mesin tak dapat menangani lebih dari ekuivalen 15000 truk tipe I

Batasan lain akan jumlah truk dari tiap tipe yang dapat dikerjakan dapat disajikan seperti berikut:

Tipe truk	Kendala pemasangan akhir (maksimum)	Pencetakan (Rasio terhadap tipe I)	Pemasang an Mesin (rasio thd tipe I)	Keuntungan
I	7500	1.0	1.0	350
II	5000	1.4	1.6	450
III	1000	2.0	3.0	500
IV	9000	0.8	1.0	300
V	3000	2.2	2.6	400

Penjadwalan Mesin. Sebuah pabrik memiliki 4 mesin, dimana tiap mesin dapat memproduksi 3 variasi dari sebuah produk. Keuntungan per jam apabila memproduksi 3 variasi pada tiap-tiap mesin diberikan seperti pada tabel berikut

Tabel 5. 1 Tentang A

Variasi	**Mesin**			
	1	**2**	**3**	**4**
1	5	6	4	3
2	5	4	5	4
3	6	7	2	8

Laju produksi per jam dari keempat mesin apabila digunakan untuk memproduksi 3 variasi produk tersebut dapat disajikan seperti pada tabel berikut

Tabel 5. 2 Tentang B

Variasi	**Mesin**			
	1	**2**	**3**	**4**
1	8	2	4	9
2	7	6	6	3
3	4	8	5	3

Permintaan akan 3 variasi tersebut untuk bulan mendatang, berturut-turut adalah sebagai berikut: 700, 500, dan 400 untuk variasi 1, 2, dan 3. Maksimum jam yang tersedia untuk memproduksi 3 variasi selama periode produksi dengan empat mesin berturut-turut adalah 90, 75, 90, dan 85 jam untuk mesin 1, 2, 3, dan 4. Bagaimana jadwal optimal alokasi penggunaan mesin dan berapa keuntungan maksimumnya ?

Pendekatan Grafis

Meskipun jawab dari model linier dengan metode pendekatan grafis ini terbatas untuk masalah yang memiliki 2 variabel, tapi pendekatan ini memberikan cara pandang yang berharga sebelum mempelajari metode berikutnya, yaitu metode simplex yang dapat digunakan untuk memecahkan permasalahan yang lebih besar.

Kita ambil saja permasalahan yang ada diatas yang menggunakan dua variabel yaitu permaslahan produksi Televisi. Misalkan hanya ada dua tipe televisi yang dapat diproduksi dengan keuntungan 6 satuan untuk tipe I dan 4 satuan untuk tipe II. Sebagai tambahan berturut-turut 2 dan 3 satuan bahan diperlukan untuk memproduksi satu televisi tipe I dan II, dan berturut-turut 4 dan 2 satuan waktu yang diperlukan untuk memproduksi satu televisi tipe I dan II. Jika tersedia 100 satuan bahan dan 120 satuan waktu, berapa banyak masing-masing tipe televisi harus diproduksi sehingga keuntungannya maksimum?

Kita mulai dengan memisalkan x_1 adalah banyaknya televisi tipe I yang diproduksi, dan x_2 merupakan banyaknya televisi tipe II yang diproduksi. Karena tiap televisi tipe I dan II menghasilkan keuntungan berturut-turut 6 dan 4 satuan, maka fungsi yang harus dimaksimumkan adalah $z = 6x_1 + 4x_2$. Kendala dari permasalahan adalah

$$2x_1 + 3x_2 \leq 100 \quad \text{kendala bahan}$$

$$4x_1 + 2x_2 \leq 120 \quad \text{kendala waktu}$$

dan

$$x_1, \quad x_2 \geq 0$$ batasan bahwa banyaknya yang diproduksi tidak negatif.

Fungsi yang akan dioptimumkan (dalam hal ini maksimumkan) disebut sebagai ***fungsi tujuan*** (*objective function*). Karena fungsi tujuan dan semua kendala mempunyai bentuk linier dalam variabel yang dipakai, x_1 dan x_2, maka formulasi ini disebut sebagai ***model pemrograman linier***.

Secara umum, koeffisien variabel dalam fungsi tujuan disebut sebagai *koeffisien biaya*. Dalam contoh ini, lebih tepat apabila disebut dengan *koeffisien keuntungan*.

Karena hanya ada dua variabel yang digunakan dalam contoh ini, masalah ini dapat diselesaikan dengan pendekatan grafis. Langkah-langkah yang diperlukan untuk menyelesaikan permasalahan adalah

Langkah 1
Gambar kendala pertama dengan membuatnya sebagai persamaan. Yaitu buat garis $2x_1 + 3x_2 = 100$

Langkah 2
Tentukan titik-titik yang memenuhi kendala pertama ini, $2x_1 + 3x_2 \leq 100$. Hal ini dapat dilakukan dengan mudah dengan cara mencek titik pusat (0,0) apakah memenuhinya. Karena (2)(0)+(3)(0)=0 < 100 maka semua titik dibawah garis $2x_1 + 3x_2 = 100$ memenuhi kendala pertama ini. Perlu dicatat bahwa titik pusat ini memenuhi semua kendala $\leq$ jika sisi kanan dari kendala tidak negatif.

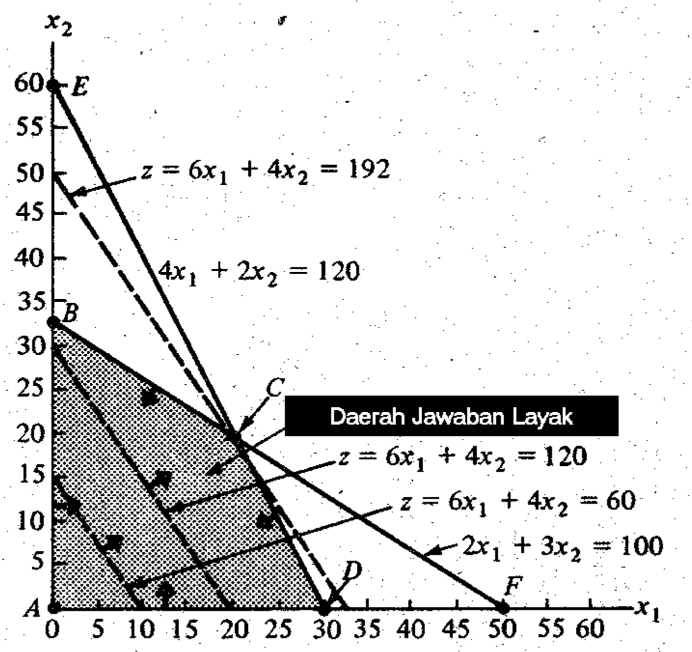
Langkah 3
Gambar kendala kedua dan terakhir dengan membuatnya sebagai persamaan. Yaitu buat garis $4x_1 + 2x_2 = 120$

Langkah 4
Tentukan gugus titik yang memenuhi kendala kedua $4x_1 + 2x_2 \leq 120$. Lagi, titik pusat memenuhi kendala ini, maka semua titik di bawah garis $4x_1 + 2x_2 = 120$ haruslah memenuhi kendala kedua ini.

Langkah 5
Tentukan daerah atau wilayah dimana $x_1 \geq 0$ dan $x_2 \geq 0$.

Langkah 6
Ambil interseksi atau irisan dari daerah-daerah yang telah ditentukan pada langkah 2, 4, dan 5. Irisan daerah ini merupakan ***jawaban layak*** (*feasible solution*) dari permasalahan. Tiap titik dalam daerah ini memenuhi semua kendala dan batasan, dan merupakan kandidat titik yang menghasilkan

keuntungan maksimum. Namun demikian, daerah ini memiliki jawaban layak yang tak hingga jumlahnya. Bagaimana jumlah ini dapat dikurangi hingga beberapa titik saja yang perlu dipertimbangkan untuk mencapai tujuan?



Sumber: Gillet, 1982

Langkah 7

Dapat dilihat bahwa fungsi tujuan $z = 6x_1 + 4x_2$ akan mencapai nilai maksimum di salah satu pojok yang ditunjukkan dengan huruf A, B, C, atau D. Dengan demikian, pembuatan garis $z = 6x_1 + 4x_2 = C_1$ dan $z = 6x_1 + 4x_2 = C_2$ dimana C_1 dan C_2 merupakan 2 konstanta yang berbeda, kemiringan (slope) dari fungsi tujuan demikian juga arah pergerakannya dengan semakin naiknya nilai z dapat diobservasi. Jawaban optimal direpresentasikan dengan titik terakhir dari jawaban layak dimana fungsi tujuan melewati apabila z bertambah besar. Jawaban ini akan memaksimumkan nilai z dalam daerah atau wilayah jawaban layak.

Dalam contoh kita $z = 6x_1 + 4x_2 = 60$ melewati daerah jawaban layak dekat dengan titik A. Semakin naiknya z misalnya ke nilai 120 kemudian ke nilai 192, slope garis yang mewakili fungsi tujuan tetap sama tetapi garis ini bergerak mendekati titik C. Tiap nilai z akan menghasilkan sebuah garis lurus paralel dengan garis $z = 6x_1 + 4x_2 = d$ untuk sembarang nilai d. Jika d kurang dari 120, garis akan paralel dan berada di bawah dan sebelah kiri $z = 6x_1 + 4x_2 = 120$. Namun, sebaliknya jika d lebih besar dari 120, garis akan paralel dan berada do atas dan sebelah kanan dari $z = 6x_1 + 4x_2 = 120$. Jika d terus bertambah besar, garis akan berada di luar wilayah jawaban layak. Dengan demikian kita dapat membuat garis $z = 6x_1 + 4x_2 = d$ untuk dua nilai d dan mengamati slope dan arah gerakan garis apabila d semakin bertambah untuk menentukan titik pojok dimana nilai maksimum (atau minimum) dari fungsi tujuan dicapai. Dalam hal ini, titik C menghasilkan nilai maksimum. Titik ini merupakan titik pertemuan antara garis $2x_1 + 3x_2 = 100$ dan $4x_1 + 2x_2 = 120$. Dengan demikian jawab dari dua persamaan linier ini adalah nilai x_1 dan x_2 yang menghasilkan nilai maksimum dari fungsi tujuan.

Langkah 8

Untuk menyelesaikan persamaan $2x_1 + 3x_2 = 100$ dan $4x_1 + 2x_2 = 120$, maka kalikan persamaan pertama dengan -2 dan tambahkan hasilnya dengan persamaan kedua.

$$\begin{array}{rcl} -4x_1 - 6x_2 & = & -200 \\ +4x_1 + 2x_2 & = & +120 \\ \hline -4x_2 & = & -80 \\ x_2 & = & +20 \end{array}$$

Jika x_2 dikurangkan dalam persamaan asal pertama, maka kita dapatkan $x_1 = 20$.

Dengan demikian, 20 unit tipe I dan 20 unit tipe II harus diproduksi untuk menghasilkan maksimum keuntungan z = (6)(20)+(4)(20) = 200 satuan keuntungan.

Maksimisasi dengan Kendala "≤"

Apabila model pemrograman linier memiliki lebih dari dua variabel, maka terlihat dengan jelas bahwa pendekatan grafis kehilangan daya tariknya, tak dapat digunakan untuk memecahkan masalah ini. Konsekuensinya, prosedur lain harus digunakan untuk memecahkan permasalahan yang menggunakan variabel lebih besar dari dua buah. Metode Simplex telah terbukti merupakan prosedur paling efisien untuk menyelesaikan model pemrograman linier dengan sembarang jumlah variabel dan kendala. Dalam prakteknya, jumlah variabel dan kendala harus dibatasi sesuai dengan kapasitas komputer yang digunakan untuk pemrosesan. Dalam bidang industri, permasalahan dengan beberapa ratus variabel dan kendala bahkan merupakan hal yang biasa.

Dalam bagian ini, model pemrograman linier kelas tertentu akan dipelajari untuk memantapkan rasionalisasi apa yang terjadi dibelakang metode simplex. Bagian ini akan dibatasi dengan pembahasan maksimisasi fungsi tujuan dan semua kendala dengan menggunakan notasi lebih kecil atau sama dengan (≤). Secara khusus misalnya perhatikan permasalahan sebelum ini

Maksimumkan $z^* = 6x_1 + 4x_2$.

dengan kendala

$$2x_1 + 3x_2 \le 100$$
$$4x_1 + 2x_2 \le 120$$

dan

$$x_1, \quad x_2 \ge 0 .$$

Untuk dapat menggunakan metode simplex, model pemrograman linier asal harus dikonversi ke model equivalen yang mengekspresikan semua kendala sebagai persamaan. Dua masalah tersebut akan menjadi ekuivalen dalam hal jawab optimal dari permasalahan baru juga kan merupakan jawab optimal dari permasalahan aslinya, jika jawaban tersebut ada.

Untuk merubah menjadi permasalahan yang ekuivalen, diperlukan variabel tak negatif x_3 dan x_4 yang disebut sebagai ***slack variable*** atau *variabel slack* yang ditambahkan berturut-turut di sebelah kiri dari kendala pertama dan kedua. Juga koeffisien biaya sebesar 0 diberikan untuk tiap variabel slack ini dalam fungsi tujuan. Dengan demikian, masalah yang ekuivalen dengan masalah aslinya adalah

Maksimumkan $z = 6x_1 + 4x_2 + 0\,x_3 + 0\,x_4$

dengan kendala

$$2x_1 + 3x_2 + x_3 \quad = 100$$

$$4x_1 + 2x_2 + \quad x_4 = 120$$

dan

$$x_1, x_2, x_3, x_4 \geq 0 .$$

Tujuan penambahan variabel slack untuk setiap kendala *lebih kecil atau sama dengan* (≤) adalah untuk merubah kendala tersebut menjadi kendala *sama dengan* (=) sehingga dapat diselesaikan. Tentu, hal ini masih saja terdapat tak terhingga jawaban yang memenuhi persamaan; namun demikian, hanya jawaban yang berkenaan dengan titik-titik A, B, C, dan D yang perlu diuji. Secara umum, titik-titik yang berada di pinggir daerah jawaban layak yang perlu diuji.

Jawaban Dasar Layak

Tiap jawaban yang berhubungan dengan satu dari titik A hingga F dalam gambar terdahulu didapatkan dengan membuat dua dari empat variabel dalam masalah yang ekuivalen sama dengan nol dan mencari jawab untuk dua variabel yang lainnya. Tiap jawaban yang didapat dengan cara ini disebut sebagai **jawaban dasar**, dan variabel yang tidak bernilai nol disebut sebagai **variabel dasar**. Semua jawaban dasar tidak memenuhi kendala awal dan batasan ketidaknegatifan; namun demikian, jawaban-jawaban dasar yang memenuhinya disebut sebagai **jawaban dasar yang layak** (*basic feasible solution*). Jawaban dasar layak dipenuhi oleh titik-titik A, B, C, dan D. dalam gambar sebelum ini.

Secara umum, jika model pemrograman linier ekuivalennya memiliki m persamaan dalam n variabel (n>m), maka definisi berikut berlaku.

1. *Jawaban dasar*. Misalkan (n-m) variabel diberikan nilai nol. Jika sistem hasil dari m persamaan dalam m variabel yang tak diketahui memiliki satu jawab yang khas (unik), maka jawaban ini disebut dengan jawaban dasar. Banyaknya jawaban dasar harus lebih kecil atau sama dengan banyaknya cara m variabel dipilih dari n variabel yang ada, yaitu

$$C_m^n = \frac{n!}{m!(n-m)!} = \frac{n(n-1)(n-2)...1}{[m(m-1)(m-2)...1][(n-m)(n-m-1)(n-m-2)...1]}$$

2. *Jawaban dasar yang layak*. Jawaban dasar yang memenuhi semua kendala asal dan batasan bahwa variabel tak negatif disebut sebagai jawaban dasar yang layak.
3. *Jawaban dasar yang layak dan optimal*. Jawaban dasar yang layak yang mengoptimalkan fungsi tujuan disebut sebagai jawaban dasar yang layak dan optimal.

Metode simplex untuk maksimisasi fungsi tujuan berawal dari nilai atau titik ekstrim atau pojok dari wilayah jawaban yang layak. Titik ini berhubungan dengan salah satu jawaban dasar yang layak dari permasalahan ekuivalennya. Metode ini kemudian bergerak ke titik ekstrim dekatnya, yang berhubungan dengan jawaban dasar yang layak yang tidak menurunkan nilai fungsi tujuan, jika ada. Proses ini terus berlanjut hingga sebuah jawaban optimal dari permasalahan ekuivalennya tercapai. Hal ini tak perlu benar apabila terdapat kendala *lebih besar atau sama dengan* ($\geq$) atau *sama dengan* (=) dalam permasalahan asalnya.

Dengan menggunakan permasalahan yang telah kita sampaikan di atas, maka kita ketahui bahwa terdapat dua persamaan dalam empat variabel yang tak dikethui, sehingga terdapat paling banyak 6 jawaban dasar. Dalam contoh ini, jika kita buat dua variabel bernilai nol, maka dua variabel yang lain dapat diketahui nilainya dan unik.

Bukan Jawaban Dasar	Jawaban Dasar	Nilai fungsi tujuan (z)	Jawaban Dasar Layak	Titik dalam gambar
$X_1 = X_2 = 0$	$X_3 = 100$ $X_4 = 120$	0	Ya	A
$X_1 = X_3 = 0$	$X_2 = 33.33$ $X_4 = 53.33$	132.32	Ya	B
$X_1 = X_4 = 0$	$X_2 = 60$ $X_3 = -80$	240	Bukan	E
$X_2 = X_3 = 0$	$X_1 = 50$ $X_4 = -80$	300	Bukan	F
$X_2 = X_4 = 0$	$X_1 = 30$ $X_3 = 40$	180	Ya	D
$X_3 = X_4 = 0$	$X_1 = 20$ $X_2 = 20$	200	Ya	C

Dengan demikian jawaban dasar yang layak dan optimal dapat dilihat dari tabel diatas, yaitu tercapai apabila $x_1 = x_2 = 20$ dengan nilai maksimum 200, yang juga merupakan titik C dalam gambar diatas. Dalam kasus ini, kedua variabel slack, x_3 dan x_4 bernilai nol. Hal ini menunjukkan bahwa dua kendala pada masalah asalnya memang memiliki hubungan "=" untuk nilai-nilai x_1 dan x_2 yang memaksimumkan z*. Seringkali, satu atau lebih variabel slack memiliki nilai yang positif. Bila demikian kendala yang sesuai dalam permasalahan asalnya memiliki hubungan "≤" untuk jawab optimalnya.

Penggunaan Metode Simplex

Metode ini merupakan prosedur yang sifatnya iteratif yang memuat pergerakan dari satu jawab dasar layak ke jawab dasar layak lainnya sedemikian rupa sehingga nilai dari fungsi tujuan tidak berkurang atau tidak menurun (dalam masalah maksimisasi). Proses ini berlanjut hingga mencapai jawab optimal, jika ada.

Langkah 1

Masalah awal atau asal dari model pemrograman linier

Maksimumkan $z^* = 6x_1 + 4x_2$

dengan kendala $2x_1 + 3x_2 \leq 100$

$$4x_1 + 2x_2 \leq 120$$

dan $x_1, \quad x_2 \geq 0$

dikonversi ke model yang ekuivalen dengannya dengan menambahkan variabel slack ke tiap kendala dan memberikan koeffisien 0 pada tiap variabel slack dalam fungsi tujuan. Dengan demikian model ekuivalennya adalah

Maksimumkan $z = 6x_1 + 4x_2 + 0x_3 + 0x_4$

dengan kendala $2x_1 + 3x_2 + x_3 \quad = 100$

$$4x_1 + 2x_2 \quad + x_4 = 120$$

dan $x_1, x_2, x_3, x_4 \geq 0$

Langkah 2

Karena setiap konstanta di sebelah kanan tanda "=" lebih besar dari nol, maka lakukan Langkah 3. Metode Simplex mensyaratkan bahwa semua konstanta yang berada di sebelah kanan tanda "=" harus tidak negatif.

Langkah 3

Tulis kembali masalah ekuivalennya sebagai berikut

Maksimumkan z

dengan kendala $z - 6x_1 - 4x_2 = 0$

$$2x_1 + 3x_2 + x_3 = 100$$

$$4x_1 + 2x_2 + x_4 = 120$$

dan $x_1, x_2, x_3, x_4 \geq 0$

atau bila dalam tabel, kendala dapat dituliskan sebagai berikut

z	**x_1**	**x_2**	**x_3**	**x_4**			Persamaan
1	-6	-4			=	0	(PL.1)
	2	3	1		=	100	(PL.2)
	4	2		1	=	120	(PL.3)

Langkah 4

Pilih jawab dasar yang layak awalnya. Yaitu pilih dua variabel atau variabel dan berikan nilainya sama dengan nol dan cari nilai variabel lainnya. Dengan memisalkan nilai $x_1 = 0$ dan $x_2 = 0$; maka kita akan dapatkan dengan mudah $x_3 = 100$ dan $x_4 = 120$ yang akan menghasilkan nilai $z = 0$. Variabel x3 dan x4 yang tidak bernilai nol ini disebut dengan variabel atau variabel dasar dan membentuk suatu basis. Variabel x_1 dan x_2 yang bernilai nol ini disebut sebagai **variabel bukan dasar**. Hasil ini ($x_1 = 0$ dan $x_2 = 0$) sebenarnya adalah titik A.

Proses berlanjut dengan merubah satu dari variabel dasar menjadi variabel bukan dasar dan sebaliknya. Dengan demikian, salah satu x_3 atau x_4 diberikan nilai nol dan salah satu x_1 atau x_2 diperbolehkan bernilai lebih besar dari nol. Variabel yang nilainya menjadi nol dikatakan "meninggalkan" basis dalam jawab dasar layak berikutnya, sedangkan yang variabel bukan dasar yang dapat bernilai lebih besar dari nol dikatakan "memasuki" basis dalam jawab dasar layak berikutnya. Namun, bagaimana seleksi harus dilakukan ?

Langkah 5

Untuk menentukan variabel yang memasuki basis dalam jawab dasar layak berikutnya perlu dilihat koeffisien dari variabel bukan dasarnya dalam persamaan (PL.1) yang paling negatif. Jika semua koeffisien lebih besar atau sama dengan nol, maka jawab optimal telah tercapai. Tiap koeffisien negatif ini menggambarkan seberapa banyak z akan bertambah untuk setiap satuan penambahan variabel bukan dasar yang sesuai. Dalam kasus ini, koeffisien x_1 adalah yang paling negatif, yaitu -6, dengan demikian variabel x_1 akan memasuki jawab dasar layak berikutnya.

Langkah 6

Periksa rasio positif dari konstanta sebelah kanan dari tiap persamaan dengan koeffisien x_1 nya.

$100 / 2 = 50$ untuk persamaan (PL.2)

$120 / 4 = 30$ untuk persamaan (PL.3)

Nilai minimum didapat pada persamaan (PL.3). Sebagai konsekuensinya, variabel dasar saat ini dalam persamaan (PL.3), variabel 4, akan meninggalkan basis dalam jawab dasar layak berikutnya. Jika rasio positif tidak didapatkan, maka fungsi tujuan tidak berbatas atas oleh kendala yang ada. Dengan demikian, jawab optimalnya tak ada.

Langkah 7

Lakukan transformasi elemen untuk tiap persamaan dalam tahap 3 hingga koeffisien x_1 bernilai 1 dalam persamaan terakhir dan 0 dalam dua persamaan pertama. Proses ini dikenal sebagai *perubahan basis*. Yang dimaksud transformasi elemen disini adalah

Mengalikan atau membagi persamaan dengan suatu konstanta

Menambah kelipatan dari suatu persamaan ke persamaan lainnya

Operasi ini tidak akan merubah persamaan yang berhubungan dengan jawaban-jawaban yang mungkin.

Untuk mendapatkan koeffisien x_1 bernilai 1 dalam persamaan ketiga, hanya dengan membagi persamaan (PL.3) dengan 4, maka koeffisien-koeffisien tersebut menjadi

z	x_1	x_2	x_3	x_4			Persamaan
1	-6	-4			=	0	(PL.4)
	2	3	1		=	100	(PL.5)
	1	2/4		1/4	=	30	(PL.6)

Kemudian, kalikan persamaan (PL.6) dengan 6 dan tambahkan hasilnya ke persamaan (PL.4) sehingga diperoleh

z	x_1	x_2	x_3	x_4			Persamaan
1	0	-1		6/4	=	180	(PL.7)
	2	3	1		=	100	(PL.8)
	1	2/4		1/4	=	30	(PL.9)

Akhirnya, kalikan persamaan (PL.6) dengan -2 dan tambahkan hasilnya ke persamaan (PL.5) sehingga diperoleh

z	x_1	x_2	x_3	x_4			Persamaan
1	0	-1		6/4	=	180	(PL.10)
	0	2	1	-2/4	=	40	(PL.11)
	1	2/4		1/4	=	30	(PL.12)

Kita tentukan dalam langkah 5 dan 6 bahwa x4 harus meninggalkan basis dalam jawab dasar layak berikutnya. Karenanya, bila x_2 dan x_4 diberikan nilai 0, persamaan (PL.10) hingga (PL.12) menghasilkan $x_1 = 30$, $x_3 = 40$ dan z = 180. Titik dengan $x_1 = 30$ dan $x_2 = 0$ adalah titik D dalam gambar terdahulu. Pergerakan dari titik A ke titik D menaikkan nilai z dari 0 menjadi 180.

Langkah 8

Koeffisien dari variabel bukan dasar dalam persamaan (PL.10) yang paling negatif adalah koeffisien x_2 yaitu sebesar -1. Sehingga x_2 merupakan variabel yang masuk ke basis berikutnya.

Langkah 9

Periksa rasio positif dari konstanta sebelah kanan tanda "=" dari persamaan (PL.11) dan (PL.12) terhadap koeffisien x2 nya, dan dapatkan nilai minimumnya.

40 / 2 = 20 untuk persamaan (PL.11)

30 / (2/4) = 60 untuk persamaan (PL.12)

Nilai minimum diperoleh pada persamaan (PL.11), karenanya variabel dasar sekarang dalam persamaan (PL.11), variabel 3, akan meninggalkan basis dalam jawab dasar layak berikutnya.

Langkah 10

Bagi persamaan (PL.11) dengan 2 sehingga menghasilkan

z	x_1	x_2	x_3	x_4			Persamaan
1	0	-1		6/4	=	180	(PL.13)
	0	1	1/2	-1/4	=	20	(PL.14)
	1	2/4		1/4	=	30	(PL.15)

Langkah 11

Tambahkan persamaan (PL.14) ke persamaan (PL.13) sehingga diperoleh

z	x_1	x_2	x_3	x_4			Persamaan
1	0	0	1/2	5/4	=	200	(PL.16)
	0	1	1/2	-1/4	=	20	(PL.17)
	1	2/4		1/4	=	30	(PL.18)

Langkah 12

Kalikan persamaan (PL.17) dengan -2/4 dan tambahkan hasilnya ke persamaan (PL.18) sehingga hasilnya

z	x_1	x_2	x_3	x_4			Persamaan
1	0	0	1/2	5/4	=	200	(PL.19)
	0	1	1/2	-1/4	=	20	(PL.20)
	1	0	-1/4	6/16	=	20	(PL.21)

Dengan demikian, karena x_3 dan x_4 akan menjadi variabel bukan dasar dalam jawab dasar layak berikutnya (berarti $x_3 = 0$ dan $x_4 = 0$), dan x_1 dan x_2 merupakan variabel dasar, maka dari persamaan (PL.19) hingga (PL.21) diperoleh $x_1 = 20$, $x_2 = 20$ dan nilai z = 200.

Karena koeffisien x_3 dan x_4 semuanya positif dalam persamaan (PL.19), maka jawab optimal telah tercapai. Jawaban tersebut adalah $x_1 = 20$, $x_2 =$

20, x_3 = 0, dan x_4 = 0 untuk masalah ekuivalennya, serta x_1 = 20 dan x_2 = 20 untuk masalah asalnya. Nilai maksimum fungsi tujuan asalnya adalah z* = (6)(20)+(4)(20) = 200.

Maksimisasi dengan Kendala "≥" dan "="

Jika permasalahan terdahulu kemudian dibatasi oleh keadaan dimana sebanyak 14 televisi tipe I dan sedikitnya 22 televisi tipe II harus diproduksi, maka model pemrograman liniernya menjadi

Maksimumkan $z^* = 6x_1 + 4x_2$

dengan kendala $2x_1 + 3x_2 \leq 100$ (PL.22)

$4x_1 + 2x_2 \leq 120$ (PL.23)

$x_1 = 14$ (PL.24)

$x_2 \geq 22$ (PL.25)

dan $x_1, \quad x_2 \geq 0$

Dalam permasalahan ini telah ditambahkan kendala *sama dengan* dan *lebih besar atau sama dengan*. Tujuan pertama dari sembarang model pemrograman linier adalah untuk mengkonversi ke model ekuivalen dimana semua kendala mempunyai hubungan *sama dengan*. Variabel slack x3 dan x4 ditambahkan di sisi kiri pertidaksmaan (PL.22) dan (PL.23) berturut-turut untuk membuatnya menjadi persamaan. Namun demikian, sebuah bilangan positif harus dikurangkan di sebelah kiri pertidaksamaan (PL.25) sehingga menjadi persamaan. Variabel positif x5 ini disebut sebagai **variabel surplus**. Koeffisien sebesar 0 diberikan untuk tiap variabel slack dan variabel surplus dalam fungsi tujuan. Masalah ekuivalennya, demgan demikian, dapat dituliskan sebagai berikut

Maksimumkan $z = 6x_1 + 4x_2 + 0x_3 + 0x_4 + 0x_5$

dengan kedala $2x_1 + 3x_2 + x_3 = 100$ (PL.26)

$4x_1 + 2x_2 + x_4 = 120$ (PL.27)

$x_1 = 14$ (PL.28)

$x_2 - x_5 = 22$ (PL.29)

untuk semua $x_i \geq 0$, $i = 1,2,3,4,5$

Metode simplex mensyaratkan bahwa

1. Semua konstanta di sebelah kanan tanda sama dengan lebih besar atau sama dengan nol
2. Tiap persamaan memiliki variabel yang koeffisiennya 1 dalam persamaan itu dan 0 dalam semua persamaan lainnya.

Kondisi ini memberikan satu gugus variabel yang dapat digunakan sebagai variabel dasar dalam jawab awalnya, dan juga meyakinkan bahwa jawab yang menggunakan variabel-variabel sebagai variabel dasar akan merupakan jawab dasar layak untuk model pemrograman linier ekuivalennya.

Kondisi pertama dipenuhi dalam masalah contoh dan variabel 3 dan 4 berturut-turut memenuhi kondisi 2 dalam persamaan (PL.26) dan (PL.27). Namun demikian, tak ada variabel dalam persamaan (PL.28) dan (PL.29) yang memenuhi kondisi 2. Karenanya, variabel lain harus ditambahkan.

Meskipun (PL.28) dan (PL.29) telah merupakan persamaan, **variabel artifisial** atau variabel buatan, x_6 dan x_7 ditambahkan ke bagian kiri persamaan-persamaan tersebut sehingga mereka memiliki variabel-variabel yang memenuhi kondisi 2. Tiap variabel artifisial ini diberikan sembarang koeffisien yang kecil dan negatif (secara aljabar sebut saja -T) dalam fungsi tujuan.

Variabel artifisial ini hanya digunakan agar prosedur simplex segera dapat dilakukan dan memberikan jawaban artifisial ke jawaban asalnya sepanjang variabel artifisial ini berada dalam basis pada taraf positif (tak kosong). Segera setelah variabel artifisial meninggalkan basis, variabel tersebut kemudian dapat diabaikan karena tak akan lagi memasuki basis. Namun, jika varaiebl artifisial tak dapat "diarahkan" atau "dipaksa" menjadi nol dan dengan demikian berada dalam basis dan bernilai positif dimana jawab optimal dari permasalahan ekuivalennya tercapai, maka masalah asalnya tidak memiliki jawaban dasar yang layak.

Dengan penambahan variabel artifisial tersebut, x_6 dan x_7 ke persamaan (PL.28) dan (PL.29), maka permasalahan menjadi

Maksimumkan $z = 6x_1 + 4x_2 + 0x_3 + 0x_4 + 0x_5 - Tx_6 - Tx_7$

dengan kedala

$$2x_1 + 3x_2 + x_3 = 100 \quad \text{(PL.30)}$$

$$4x_1 + 2x_2 + x_4 = 120 \quad \text{(PL.31)}$$

$$x_1 + x_6 = 14 \quad \text{(PL.32)}$$

$$x_2 - x_5 + x_7 = 22 \quad \text{(PL.33)}$$

untuk semua xi ≥ 0, i = 1,2,3,4,5,6,7

Peubah x_3, x_4, x_6, dan x_7 sekarang dapat digunakan sebagai variabel dasar bagi jawab awal permasalahan selagi variabel x_1, x_2, dan x_5 sebagai variabel bukan dasar. Jawab dari permasalahan ekuivalennya dengan demikian adalah

$x_1 = 0$, $x_2 = 0$, $x_3 = 100$, $x_4 = 120$, $x_5 = 0$, $x_6 = 14$, dan $x_7 = 22$

Ini merupakan jawab dasar layak dari persamaan (PL.31) hingga (PL.34); tetapi, karena variabel artifisialnya bernilai positif, jawab ini bukan jawab dasar yang layak untuk model pemrograman linier asalnya. Sebagai contoh, $x_1 = 14$ dan $x_2 \geq 22$ dalam model asal atau aslinya, tidak dapat dipenuhi oleh jawaban diatas.

Tujuan dari pemberian sembarang koefisien biaya yang kecil dan negatif (secara aljabar) pada variabel artifisial adalah untuk memaksa mereka keluar dari basis, atau setidaknya membuatnya menjadi nol, karena sembarang nilai positif dari variabel artifisial akan membuat nilai dari fungsi tujuan akan kecil.

Selangkah lagi diperlukan untuk menyiapkan persamaan (PL.30) hingga (PL.34) untuk metode simplex. Dengan menuliskan kembali persamaan (PL.30) menjadi

$$z - 6x_1 - 4x_2 + Tx_6 + Tx_7 = 0 \quad \text{(PL.35)}$$

dan menghilangkan variabel x_6 dan x_7 dari persamaan (PL.35). Dengan demikian, persamaan (PL.33) dan (PL.34) masing-masing dikalikan dengan -T dan ditambahkan ke persamaan (PL.35) untuk mendpatkan model pemrograman ekuivalen yang baru sebagai berikut

Maksimumkan z

dengan kendala

$$2x_1 + 3x_2 + x_3 = 100 \quad \text{(PL.36)}$$

$$4x_1 + 2x_2 \qquad + x_4 \qquad\qquad\qquad = 120 \qquad \text{(PL.37)}$$

$$x_1 \qquad\qquad\qquad\qquad + x_6 \quad = 14 \qquad \text{(PL.38)}$$

$$x_2 \qquad\qquad - x_5 \qquad + x_7 = 22 \qquad \text{(PL.39)}$$

$$z - (T+6)x_1 - (T+4)x_2 \qquad + Tx_5 \qquad\qquad = -36T \qquad \text{(PL.40)}$$

Persamaan (PL.36) hingga (PL.40) sudah dalam bentuk sebagaimana disyaratkan apabila digunakan penyelesaian metode Simplex.

Minimisasi Fungsi Tujuan

Jika fungsi tujuan dalam model pemrograman linier adalah Minimumkan, bukannya Maksimumkan, adalah hal yang mudah dengan menkonversi model dengan fungsi tujuan Maksimumkan. Model konversi ini dapat diselesaikan dengan metode simplex maksimisasi seperti apa yang baru saja dipelejari.

Misalkan, permasalah model pemrograman liniernya

Minimumkan $\quad z^* = \sum_{j=1}^{b} c_j x_j$

dengan kendala $\quad \sum_{j=1} a_{ij} x_j \quad (\le, =, \ge) \quad b_i \qquad$ i = 1,2,…,m

dan batasan bahwa semua $x_j \ge 0$ untuk semua j = 1,2,…,k

Dengan mengalikan fungsi tujuannya dengan -1, maka permasalahan diatas sama dengan

Maksimumkan $\quad \tilde{z} = -z^* = -\sum_{j=1}^{b} c_j x_j$

dengan kendala $\quad \sum_{j=1} a_{ij} x_j \quad (\le, =, \ge) \quad b_i \qquad$ i = 1,2,…,m

dan batasan bahwa semua $x_j \ge 0$ untuk semua j = 1,2,…,k

Maksimisasi dengan metode simplex kemudian dapat dilakukan untuk menjawab permasalahan baru tersebut, dan kemudian nilai fungsi tujuan untuk masalah aslinya adalah $z^* = -\tilde{z}$. Nilai variabel yang memaksimumkan $\tilde{z}$ juga meminimumkan z^*.

Sebagai teladan

Minimumkan $z^* = x_1 + 2x_2$

dengan kendala

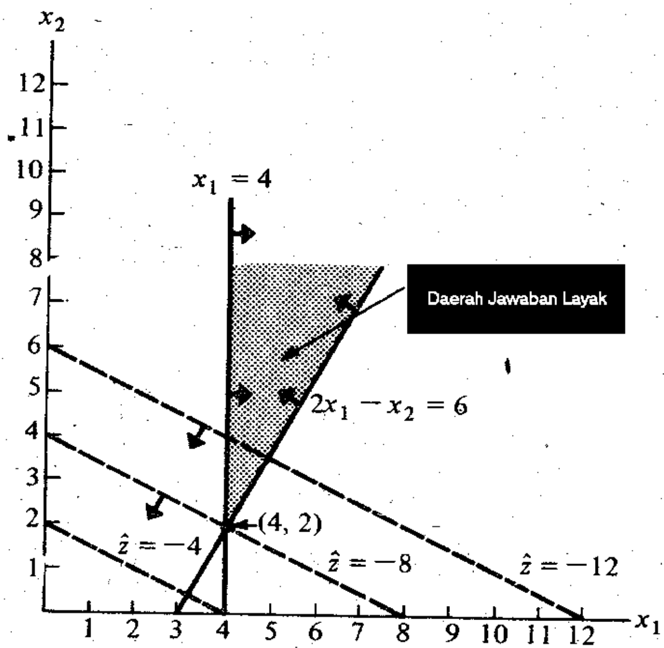
$$x_1 \geq 4$$
$$2x_1 - x_2 \leq 6$$
$$x_2 \geq 0$$

maka permasalahan tersebut setara dengan

Maksimumkan $\tilde{z} = -z^* = -x_1 - 2x_2$

dengan kendala

$$x_1 \geq 4$$
$$2x_1 - x_2 \leq 6$$
$$x_2 \geq 0$$



Sumber: Gillet, 1982

Gambar diatas mengilustrasikan jawaban secara grafis dari permasalahan ini. Maksimum nilai $\tilde{z}$ didapat apabila $x_1 = 4$ dan $x_2 = 2$ dan nilainya adalah -8. Dengan demikian, $x_1 = 4$ dan $x_2 = 2$ juga meminimumkan z^* pada permasalahan asalnya, dan minimum nilai dari z^* adalah min(z^*) = -max($\tilde{z}$)= -(-8) = 8.

Program Komputer

```
REM *****************************************************************
REM *          Program Optimasi Dengan Metode Simplex               *
REM *              Ir. SIGIT NUGROHO, M.Sc., Ph.D.                  *
REM *****************************************************************
REM
CLS
INPUT "Jumlah Kendala      ="; M
INPUT "Jumlah Variabel     ="; K
INPUT "Jumlah Kendala (<=) ="; NLET
INPUT "Jumlah Kendala (>=) ="; NGET
```

```
      INPUT "Jumlah Kendala (=)  ="; NET
      PRINT "Tipe Optimisasi  0 = Minimisasi"
      PRINT "                 1 = Maksimisasi"
      INPUT "Pilih 0 atau 1      ="; NTYPE
      DIM B(M + 1), C(K + 1 + 2 * NGET + NLET + NET), CODE(M + 1),
KODE$(M + 1)
      DIM A(M + 1, K + 2 * NGET + NLET + NET + 1), XB(K + 1 + 2 * NGET +
NLET + NET)
      DIM AA(M + 1, K + 2 * NGET + NLET + NET + 1), XBB(K + 1 + 2 * NGET +
NLET + NET)
      KODE$(0) = "<=": KODE$(1) = ">=": KODE$(2) = "="
      FOR I = 1 TO M
        PRINT "KODE KENDALA KE  "; I; : INPUT CODE(I)
        PRINT "BATAS KENDALA KE "; I; : INPUT B(I)
        FOR J = 1 TO K
          PRINT "KOEFFISIEN KENDALA KE "; I; " VARIABEL KE"; J; : INPUT
AA(I, J)
          A(I, J) = AA(I, J)
        NEXT J
        PRINT
      NEXT I
      FOR J = 1 TO K
        PRINT "KOEFFISIEN OPTIMISASI VARIABEL KE "; J; " ADALAH ="; :
INPUT C(J)
      NEXT J
      CLS
      PRINT
      FOR I = 1 TO M
        PRINT I;
        FOR J = 1 TO K
          PRINT A(I, J);
        NEXT J: PRINT KODE$(CODE(I)), B(I)
      NEXT I
      PRINT
      IF NTYPE <> 0 GOTO 35
      PRINT "KOEFFISIEN FUNGSI TUJUAN YANG DIMINIMISASI ADALAH"
      GOTO 37
35     PRINT "KOEFFISIEN FUNGSI TUJUAN YANG DIMAKSIMISASI ADALAH"
37     FOR J = 1 TO K
        PRINT "KOEFFISIEN KE "; J; " ADALAH "; C(J)
      NEXT J
      PRINT
      INPUT "BILA INGIN TAHAP AKHIR SAJA ... KETIKKAN ANGKA 0 ="; NOPT
```

```
      GOSUB 2000
      IF IFLAG = 1 GOTO 50
      BASICS = 0
      OPTSOL = 0
      FOR I = 1 TO M
        FOR J = 1 TO K
          A(I, K) = AA(I, J)
        NEXT J
      NEXT I
      GOSUB 3000
      IF (NFLAG = 1 OR NFLAG = 2) GOTO 50
      IF NTYPE = 1 GOTO 220
      SUM = -SUM
220     PRINT
      PRINT "NILAI OPTIMUM FUNGSI TUJUAN ADALAH ="; SUM
      GOTO 50
50     END

2000    REM SUBROUTINE SSARTV(A, XB)
      DIM ARTV(2 * (M + 1))
      IFLAG = 0
      IA = 1
      KP1 = K + 1
      MP1 = M + 1
      N = K + 2 * NGET + NLET + NET
      NP1 = N + 1
      NC = K + NGET + 1
      NC1 = NC + NLET
      INDEXG = K + 1
      INDEXL = K + NGET + 1
      INDEXE = K + NGET + NLET + 1
      FOR I = 1 TO MP1
        FOR J = KP1 TO NP1
          A(I, J) = 0
        NEXT J
      NEXT I
      FOR I = 1 TO M
        A(I, NP1) = B(I)
      NEXT I
      FOR I = 1 TO M
        IF CODE(I) = 0 GOTO 6
        IF CODE(I) = 1 GOTO 8
        ARTV(IA) = I
```

```
        IA = IA + 1
        XB(I) = INDEXE
        A(I, INDEXE) = 1
        INDEXE = INDEXE + 1
        GOTO 4
8        XB(I) = INDEXE
        ARTV(IA) = I
        IA = IA + 1
        INDEXE = INDEXE + 1
        A(I, INDEXG) = -1
        INDEXG = INDEXG + 1
        GOTO 4
6        XB(I) = INDEXL
        A(I, INDEXL) = 1
        INDEXL = INDEXL + 1
4      NEXT I
      IF INDEXG <> NC GOTO 100
      IF INDEXL <> NC1 GOTO 110
      IF INDEXE <> NP1 GOTO 120
      GOTO 151
100     PRINT : PRINT "JUMLAH KENDALA (>=) TAK SESUAI MASUKAN"
      IFLAG = 1
      RETURN
110     PRINT : PRINT "JUMLAH KENDALA (<=) TAK SESUAI MASUKAN"
      IFLAG = 1
      RETURN
120     PRINT : PRINT "JUMLAH KENDALA (=) TAK SESUAI MASUKAN"
      IFLAG = 1
      RETURN
151     IF NTYPE = 0 GOTO 12
        FOR J = 1 TO K
        A(MP1, J) = -C(J)
      NEXT J
      GOTO 53
12      FOR J = 1 TO K
        A(MP1, J) = C(J)
      NEXT J
53      FOR J = KP1 TO NP1
        A(MP1, J) = 0
        C(J) = 0
      NEXT J
      FOR J = 1 TO K
        C(J) = -A(MP1, J)
```

```
      NEXT J
      FOR J = NC1 TO N
        C(J) = -10 * 10 ^ 2
      NEXT J
      IF NGET + NET = 0 THEN RETURN
      IA = IA - 1
      KPGTE = K + NGET
      FOR J = 1 TO KPGTE
        SUM = 0
        FOR I = 1 TO IA
          SUM = SUM + A(ARTV(I), J)
        NEXT I
        A(MP1, J) = A(MP1, J) - SUM * 10 * 10 ^ 2
      NEXT J
      SUM = 0
      FOR I = 1 TO IA
        SUM = SUM + A(ARTV(I), NP1)
      NEXT I
      A(MP1, NP1) = A(MP1, NP1) - SUM * 10 * 10 ^ 2
      RETURN
      END

3000  REM SUBROUTINE SIMPLX(A, XB)
      NFLAG = 0
101     BASICS = BASICS + 1
      IF NOPT = 0 GOTO 200
105     PRINT : PRINT "JAWABAN DASAR ="; BASICS
      FOR I = 1 TO M
        PRINT I, XB(I), A(I, NP1)
      NEXT I
      SUM = 0
      FOR I = 1 TO M
        SUM = SUM + C(XB(I)) * A(I, NP1)
      NEXT I
      PRINT : PRINT "NILAI FUNGSI TUJUAN SAAT INI ADALAH ="; SUM
      IF OPTSOL = 1 GOTO 920
200     NEG = 0
      GNEG = 0
      FOR J = 1 TO N
        IF A(MP1, J) >= GNEG GOTO 21
        GNEG = A(MP1, J)
        NEG = J
21     NEXT J
```

```
      IF NEG = 0 GOTO 900
400     SPR = 10 * 10 ^ 10
      FOR I = 1 TO M
        IF A(I, NEG) <= .00001 GOTO 410
        IF A(I, NP1) / A(I, NEG) >= SPR GOTO 410
        SPR = A(I, NP1) / A(I, NEG)
        NSPR = I
410     NEXT I
      IF SPR <= 10 * 10 ^ 8 GOTO 510
      PRINT : PRINT "FUNGSI TUJUAN TIDAK DIBATASI OLEH KENDALA"
      NFLAG = 1
      RETURN
510     PELE = A(NSPR, NEG)
      FOR J = 1 TO NP1
        A(NSPR, J) = A(NSPR, J) / PELE
      NEXT J
      XB(NSPR) = NEG
      FOR I = 1 TO MP1
        IF I = NSPR GOTO 610
        HOLD = A(I, NEG)
        FOR J = 1 TO NP1
          A(I, J) = A(I, J) - HOLD * A(NSPR, J)
        NEXT J
610     NEXT I
      GOTO 101
900     OPTSOL = 1
      IF NOPT = 1 GOTO 920
      GOTO 105
920     FOR I = 1 TO M
        IF XB(I) < NC1 GOTO 930
        IF A(I, NP1) <= 0 GOTO 930
        PRINT "JAWABAN LAYAK TAK DIDAPATKAN"
        NFLAG = 2
        RETURN
930     NEXT I
      PRINT : PRINT "JAWABAN TERAKHIR LAYAK DAN OPTIMAL"
      RETURN
      END

SUB SSARTV
END SUB
```

Latihan

A. Minimumkan $z^* = -6x_1 - 4x_2$

dengan kendala
$$2x_1 + 3x_2 \leq 30$$
$$3x_1 + 2x_2 \leq 24$$
$$x_1 + x_2 \geq 3$$
$$x_1, x_2 \geq 0$$

B. Sebuah perusahaan ingin membeli paling banyak 1800 unit produk. Terdapat dua macam tipe produk, M1 dan M2. M1 memerlukan ruang 2 m^3 dengan harga Rp 12 juta per unit dan perusahaan akan memperoleh keuntungan Rp 3 juta per unit; sedangkan M2 memerlukan ruang 3 m^3 dengan harga Rp 15 juta per unit dengan keuntungan Rp 4 juta per unit. Jika dana yang tersedia untuk itu Rp 15 milliar dan gudang perusahaan yang tersedia untuk itu memiliki kapasitas 3000 m^3 untuk produk tersebut.

a. Tuliskan bentuk permasalahan programliniernya.
b. Selesaikan dengan cara grafik.

Masalah Investasi

Perhatikan permasalahan umum dalam pengalokasian sejumlah resource, uang misalnya, ke sejumlah kegiatan, misalnya program investasi, sedemikian rupa sehingga total return atau penghasilan akibat kegiatan tersebut maksimum.

Permasalahan model ini adalah : " Bagaimana mengalokasian sejumlah uang untuk sejumlah kegiatan sedemikian rupa sehingga total penghasilannya maksimum ?". Misal: kita memiliki sebanyak 8 satuan investasi. Satu satuan investasi dapat berarti Rp. 50.000,-, Rp. 100.000,- ataupun Rp. 1.000.000,- Misalkan terdapat 3 lembaga keuangan dimana kita dapat melakukan investasi. Tabel, katakanlah, keuntungan yang kita dapatkan sebagai hasil investasi sebanyak x satuan investasi ke lembaga investasi j dinyatakan sebagai $g_j(x)$. Demikian juga, satuan keuntungan atau pendapatan tambahan akibat investasi tersebut tidak harus sama dengan satuan investasinya.

X	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
$g_1(x)$	0	5	15	40	80	90	95	98	100
$g_2(x)$	0	5	15	40	60	70	73	74	75
$g_3(x)$	0	4	26	40	45	50	51	52	53

Apabila kita miliki 8 satuan investasi, berapa satuan harus diinvestasikan ke tiap lembaga investasi agar keuntungan yang didapatkan maksimum ? Bagaimana bila hanya terdapat 6 satuan investasi ?

Penghasilan dari tiap program saling bebas dari pengalokasian ke program lainnya. Sebagai contoh, $g_2(5) = 70$ merupakan penghasilan dari investasi sebesar 5 satuan investasi dalam program investasi 2 tanpa perlu tahu bagaimana alokasi 3 unit lainnya harus dialokasikan kemana. Asumsi lain yang diperlukan dalam permasalahan ini adalah

- Penghasilan atau return dari semua program dapat diukur dalam satuan yang sama. Catatan : besarnya satu satuan investasi tidak perlu sama dengan satu satuan penghasilannya.
- Total penghasilan merupakan jumlah dari penghasilan semua program investasi.

- Fungsi return merupakan fungsi menaik

Dengan asimsi-asumsi tersebut, permasalahan dapat digabungkan dalam suatu permasalahan yang lebih umum yang dapat diselesaikan secara sekuensial dalam beberapa tahapan. Setiap tahap merupakan submasalah yang apabila terselesaikan menghasilkan informasi untuk tahapan berikutnya. Jawaban atas submasalah dalam tahap terakhir juga merupakan jawaban permasalahan aslinya.

Tahap 1

Asumsikan untuk sementara hanya ada Program Investasi ke 3. Karena $g_3(x)$, fungsi penghasilan (return) dari program ke-3 merupakan fungsi yang menaik bila sejumlah x satuan diinvestasikan, kita harus menginvestasikan sebanyak-banyak modal investasi yang kita punyai. Tentunya $f_3(8) = g_3(8) = 53$ dan jumlah satuan investasi untuk menghasilkan nilai tersebut adalah $d_3(8) = 8$ dimana $f_3(8)$ merupakan penghasilan optimal dari program 3 saja bila sebanyak 8 satuan investasi digunakan.

Tahap 2

Misalkan $f_3(x) = g_3(x)$ untuk $x = 0, 1, 2, \ldots, 7$ merupakan penghasilan optimal dari program 3 saja apabila terdapat x satuan investasi yang dapat digunakan.

Tahap 3

Sekarang misalkan terdapat 2 program investasi, yaitu program ke 2 dan ke 3, dan seluruh satuan investasi yang ada dapat dialokasikan kedalam dua program ini. Karena kedua fungsi merupakan fungsi menaik, maka keseluruhan investasi yang ada dapat diinvestasikan. Pertanyaannya adalah : berapa satuan harus diinvestasikan ke tiap program ? Kita telah mengetahui besarnya penghasilan bila hanya digunakan program 3 saja, untuk itu kita perlu menguji tiap jumlah berikut:

$g_2(0)+f_3(8) = 53$
$g_2(1)+f_3(7) = 57$
$g_2(2)+f_3(6) = 66$
$g_2(3)+f_3(5) = 90$
$g_2(4)+f_3(4) = 105$
$g_2(5)+f_3(3) = 110$

$g_2(6)+f_3(2) = 99$
$g_2(7)+f_3(1) = 78$
$g_2(8)+f_3(0) = 75$

Maksimum dari penggunaan program 2 dan program 3 ini, apabila sebanyak 8 satuan investasi tersedia dituliskan sebagai

$$f_2(8) = \max_{z=0,1,2,\ldots,8} \left[g_2(z) + f_3(8-z) \right]$$

Banyaknya satuan investasi optimal dengan digunakannya 2 program investasi ini dinyatakan dengan $d_2(8)$, yaitu nilai z yang menghasilan $f_2(8)$. Dalam hal ini $f_2(8) = g_2(5) + f_3(3) = 110$ dan $d_2(8) = 5$.

Tahap 4

Masih dengan menggunakan asumsi bahwa hanya tersedia program ke 2 dan ke 3, dan asumsikan pula sekarang hanya terdapat x satuan tersedia untuk investasi dalam program ke 2 dan ke 3 (x= 0, 1, 2, …, 7). Untuk tiap nilai x kita hitung return optimal dari program gabungan ini dengan $f_2(x) = \max_{z=0,1,2,\ldots,x} \left[g_2(z) + f_3(x-z) \right]$ dan banyaknya satuan investasi yang harus diinvestasikan dalam program 2 ini adalah $d_2(x)$ yaitu nilai z yang menghasilkan $f_2(x)$. Kita hitung $f_2(x)$ dan $d_2(x)$ untuk x = 0, 1, 2, …, 7. Nilai-nilai ini adalah

x = 0 $\quad f_2(0) = \max_{z=0} \left[g_2(z) + f_3(0-z) \right]$

$= g_2(0) + f_3(0)$

$= 0$

$d_2(0) = 0$

x =1 $\quad f_2(1) = \max_{z=0,1} \left[g_2(z) + f_3(1-z) \right]$

$$= \max \begin{pmatrix} g_2(0) + f_3(1) \\ g_2(1) + f_3(0) \end{pmatrix}$$

$$= \max \begin{pmatrix} 0+4 \\ 5+0 \end{pmatrix}$$

$= 5$

$d_2(1) = 1$

Hal ini berarti bahwa apabila satu satuan investasi tersedia untuk investasi dalam program 2 dan program 3, kebijakan optimal adalah menginvestasikannya ke dalam program 2 yang akan menghasilkan 5 satuan.

$$x = 2 \qquad f_2(2) = \max_{z=0,1,2}\left[g_2(z) + f_3(2-z)\right]$$

$$= \max\begin{pmatrix} g_2(0) + f_3(2) \\ g_2(1) + f_3(1) \\ g_2(2) + f_3(0) \end{pmatrix}$$

$$= \max\begin{pmatrix} 0 + 26 \\ 5 + 4 \\ 15 + 0 \end{pmatrix}$$

$$= 26$$

$$d_2(2) = 0$$

Jika 2 satuan investasi tersedia untuk program 2 dan 3, kebijakan optimal diperoleh dengan mengalokasikan 0 satuan ke program 2 dan 2 satuan ke program 3 yang akan menghasilkan 26 satuan.

Perlu diingat bahwa kita akan menjawab seluruh kelas subpermasalahan yang akhirnya sampai dapat menjawab permasalahan awalnya. Dan juga kita harus menyelesaikan permasalahan serupa untuk program 2 dan 3 untuk sejumlah satuan investasi yang ada hingga 8 satuan investasi.

$$x = 3 \qquad f_2(3) = \max_{z=0,1,2,3}\left[g_2(z) + f_3(3-z)\right]$$

$$= \max\begin{pmatrix} g_2(0) + f_3(3) \\ g_2(1) + f_3(2) \\ g_2(2) + f_3(1) \\ g_2(3) + f_3(0) \end{pmatrix}$$

$$= 40$$

$$d_2(3) = 0 \text{ atau } 3$$

Jika kita ingin satu jawaban optimal daripada semua jawaban optimal, maka kita perlu memilih $d_2(3) = 0$ atau $d_2(3) = 3$ dan bukan keduanya. Nilai-nilai $f_2(x)$ dan $d_2(x)$ untuk x = 4, 5, 6, 7 ataupun 8 dapat dicari dengan cara yang sama.

Tahap 5

Tahap akhir ini sama dengan permasalahan awalnya. Bagaimana 8 satuan investasi ini harus dialokasikan dalam program investasi 1, 2, dan 3 sekaligus. Caranya adalah dengan memeriksa hasil dari investasi sebanyak z satuan ke program investasi 1 dan sebanyak 8-z ke program investasi 2 dan 3 untuk z = 0, 1, 2, …, 8. Dengan demikian $f_1(8) = \max\limits_{z=0,1,2,\ldots,8} g_1(z) + f_2(8-z)$ atau

$$f_1(8) = \max \begin{pmatrix} g_1(0) + f_2(8) \\ g_1(1) + f_2(7) \\ g_1(2) + f_2(6) \\ g_1(3) + f_2(5) \\ g_1(4) + f_2(4) \\ g_1(5) + f_2(3) \\ g_1(6) + f_2(2) \\ g_1(7) + f_2(1) \\ g_1(8) + f_2(0) \end{pmatrix} = \max \begin{pmatrix} 110 \\ 105 \\ 101 \\ 110 \\ 140 \\ 130 \\ 121 \\ 103 \\ 100 \end{pmatrix} = 140$$

$d_1(8) = 4$

Perlu diingat bahwa $f_1(8)$ merupakan penghasilan atau return optimal apabila 8 satuan diinvestasikan ke dalam program 1, 2, dan 3; dan $d_1(8)$ merupakan jumlah optimal yang yang harus diinvestasikan dalam program 1 pada kondisi ini. Karena $d_1(8) = 4$, hal ini berarti masih ada 4 satuan lain yang diinvestasikan ke dalam program 2 dan 3. Tetapi hal ini telah kita kerjakan pada tahap sebelumnya, $d_2(8\text{-}4) = d_2(4) = 4$ yang menunjukkan jumlah yang harus diinvestasikan ke program 2 apabila hanya program 2 dan 3 saja yang tersedia. Hal ini berarti menyisakan $d_3(4\text{-}4) = d_3(0) = 0$

satuan investasi yang harus diinvestaikan untuk program 3 saja. Dengan demikian alokasi optimal dari permasalahan yang kita selesaikan adalah

$d_1(8)$ = 4 satuan untuk program 1
$d_2(4)$ = 4 satuan untuk program 2
$d_3(0)$ = 0 satuan untuk program 3

yang akan menghasilkan return sebesar $f_1(8)$ = 140.

Kita dapat melakukan verifikasi ulang dengan menghitung hal berikut
$g_1(4) + g_2(4) + g_3(0) = 80 + 60 + 0 = 140$

Secara ringkas, fungsi persamaan pemrograman dinamis yang digunakan untuk menyelesaikan permasalahan investasi sederhana ini adalah

$f_3(x) = g_3(x)$

$d_3(x) = x$ untuk x = 0, 1, 2, …, 8

serta untuk i = 2 dan 1

$$f_i(x) = \max_{z=0,1,\ldots,x} \left(g_i(z) + f_{i+1}(x - z) \right)$$

$d_i(x) =$ nilai z yang menghasilkan $f_i(x)$ untuk x = 0, 1, 2, …, 8

dimana $f_i(x)$ merupakan penghasilan atau return optimal bila x satuan diinvestasikan ke program ke i, i+1, …,3. Dan $d_i(x)$ merupakan jumlah optimal yang harus diinvestasikan ke program i apabila x satuan investasi dialokasikan untuk program i, i+1, …,3.

x	$f_3(x)$	$d_3(x)$	$f_2(x)$	$d_2(x)$	$f_1(x)$	$d_1(x)$
0	0	0	0	0	0	0
1	4	1	5	1	5	0
2	26	2	26	0	26	0
3	40	3	40	0	40	0
4	45	4	60	4	80	4
5	50	5	70	5	90	5
6	51	6	86	4	106	4
7	52	7	100	4	120	4
8	53	8	110	5	140	4

Dalam pencapaian jawaban yang optimal dari permasalahan awalnya, kita telah jawab semua kelas permasalahan. Perlu diketahui bahwa contoh ini merupakan contoh yang sangat sederhana untuk mengilustrasikan teknik pemrograman dinamis.

Program Komputer

Berikut ini dituliskan program untuk menyelesaikan masalah investasi yang baru saja dipelajari, yang dituliskan dengan bahasa Basic.

```
REM ****************************************************************
REM *                PROGRAM MODEL INVESTASI                       *
REM *       K Satuan Investasi dan N Program Investasi             *
REM *               untuk Maksimisasi Pendapatan                   *
REM *              Ir. SIGIT NUGROHO, M.Sc., Ph.D.                 *
REM ****************************************************************
PRINT "MODEL MASALAH INVESTASI"
PRINT "Sigit Nugroho"
PRINT
PRINT
DATA 0, 4, 26, 40, 45, 50, 51, 52, 53
DATA 0, 5, 15, 40, 60, 70, 73, 74, 75
DATA 0, 5, 15, 40, 80, 90, 95, 98, 100
DATA 0, 4, 30, 38, 65, 70, 71, 72, 73
CLS
INPUT "BANYAKNYA SATUAN INVESTASI ="; K
INPUT "BANYAKNYA PROGRAM INVESTASI ="; N
DIM G(N, K + 1), F(N, K + 1), D(N, K + 1), S(N)
P = K + 1
PRINT
FOR I = 1 TO N
  FOR J = 1 TO P
          REM PRINT "PROG KE "; I; " SATUAN KE "; J; : INPUT G(I, J)
  READ G(I, J)
  NEXT J
  PRINT
NEXT I
PRINT
FOR X = 1 TO P
  F(N, X) = G(N, X)
  D(N, X) = X - 1
NEXT X
I = N - 1
```

```
130   X = 1
        F(I, X) = G(I, 1) + F(I + 1, 1)
        D(I, X) = 0
        FOR X = 2 TO P
          F(I, X) = G(I, 1) + F(I + 1, X)
          D(I, X) = 0
          FOR Z = 2 TO X
            IF G(I, Z) + F(I + 1, X - Z + 1) <= F(I, X) GOTO 230
            F(I, X) = G(I, Z) + F(I + 1, X - Z + 1)
            D(I, X) = Z - 1
230     NEXT Z
        NEXT X
        IF I = 1 GOTO 280
        I = I - 1
        GOTO 130
280   S(1) = D(1, P)
        FOR I = 2 TO N
          M = 0
          L = I - 1
          FOR J = 1 TO L
            M = M + S(J)
          NEXT J
          S(I) = D(I, P - M)
        NEXT I
        PRINT
        PRINT
370   PRINT "PENDAPATAN OPTIMUM ="; F(1, P)
        PRINT
        PRINT
380   FOR I = 1 TO N
          PRINT "INVESTASIKAN KE PROGRAM "; I, "SEBESAR "; S(I), "SATUAN"
        NEXT I
        PRINT
        PRINT
        END
```

Latihan

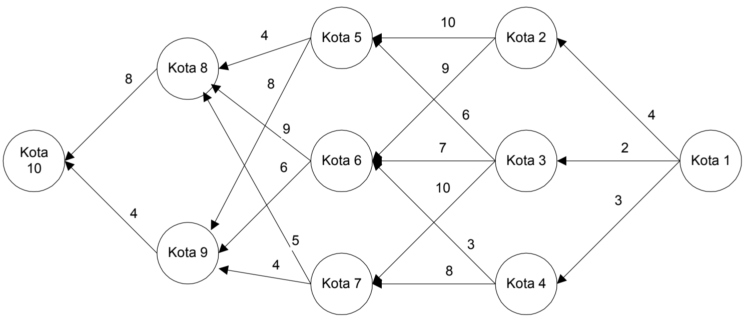
1. Dengan masih menggunakan informasi yang ada seperti tersebut di atas, bila kita hanya memiliki sebanyak 6 satuan investasi, bagaimana kita harus alokasikan agar return yang kita dapat maksimum ?
2. Bila $g_1(x) = 5\sqrt{x}$, $g_2(x) = x$, dan $g_3(x) = 0.07x^2$ dan x = 0, 1, 2, …, 19, 20.merupakan fungsi return dari tiga program investasi.

Tentukan bagaimana alokasi ke tiap program investasi untuk mendapatkan return yang maksimum, dan berapa nilai retrun tersebut ?

Masalah Stagecoach

Masalah yang agak berbeda yang dapat diselesaikan dengan pemrograman dinamis adalah apa yang dikenal dengan *Stagecoach Problem*. Hal ini mirip masalah alokasi, tetapi cukup berbeda untuk dipresentasikan. Dalam kenyataan sehari hari, ini dikenal dengan mencari route atau jalur optimal melalui suatu jaringan kerja (*network*).

Pada jaman dahulu seorang salesman yang berada di pantai timur Amerika Serikat memutuskan untuk melakukan perjalanan ke pantai barat dengan menggunakan kereta andong (*Stagecoach*) yang merupakan alat angkut utama. Ia tahu bahwa setiap jalur yang dapat ditempuh akan sangat menentukan besarnya biaya perjalanan, termasuk biaya keselamatan dalam bentuk asuransi. Tentu dapat secara umum dikatakan bahwa semakin murah harga asuransi (premi) dapat diartikan sebagai semakin aman jalurnya.



Gambar berikut memberikan ilustrasi berbagai jalur dan biaya (*policy cost*) dari satu kota ke kota lainnya. Tujuan dari permasalahan ini adalah mencari rute paling aman bila ia akan bepergian dari kota 1 di *stage* atau tahapan 1 ke kota 10 di *stage* 5.

Angka yang berada di atas garis panah, menunjukkan *policy cost* dari kota asal ke kota tujuannya. Misalnya *policy cost* dari kota 3 ke kota 6 atau disingkat sebagai C_{36} = 7 satuan biaya.

Permasalahan menentukan jalur yang paling aman berarti juga menentukan *policy cost* untuk asuransi yang paling minimum. Sebelum menjawab permasalahan tersebut, dapat dibagi kota-kota tersebut dalam *stage*, dimana kota pemberangkatan atau kota **1** berada di stage 1; kota **2**, **3**, dan **4** berada di stage 2; kota **5**, **6**, dan **7** berada di stage 3; kota **8** dan **9** berada di stage 4; dan kota tujuan akhir yaitu kota **10** berada di stage 5.

Tahap 1

Ia mulai dengan mengasumsikan bahwa posisi terakhir ia saat ini ada di stage 4. Misalkan pula ia berada di kota 8. Maka *policy cost* dari kota ini yaitu biaya minimum dari kota 8 ke kota 10 adalah 8 satuan. Jadi misalkan $f_4(8) = 8$ menunjukkan biaya minimum tersebut, dan misalkan $d_4(8) = 10$ menunjukkan kota berikutnya yang harus ditempuh. Pada stage 4 dari kota 8, maka kota berikutnya yang harus ditempuh adalah kota 10. Dengan cara yang sama ia dapat mencari $f_4(9) = 4$ dan $d_4(9) = 10$. Dengan demikian, bila ia berada di kota 8 atau 9 dalam stage 4, maka *policy* terbaik adalah pergi ke kota 10 karena itu adalah jalur satu-satunya. Ia dapat simpulkan dalam tabel kecil berikut

X (Kota)	**8**	**9**
$f_4(X)$	8	4
$d_4(X)$	10	10

Tahap 2

Ia mundur selangkah dan berada di stage 3, misalkan berada di kota 5, 6, atau 7 serta ingin meminimumkan *policy cost* ke kota 10 dengan melewati salah satu kota yang ada di stage 4. Seandainya ia berada di kota 5, maka perlu dilihat nilai minimum dari

- Jumlah biaya dari kota 5 ke kota 8 dan biaya optimal dari kota 8 ke kota 10
- Jumlah biaya dari kota 5 ke kota 9 dan biaya optimal dari kota 9 ke kota 10

Nilainya adalah

- 4 + $f_4(8)$ = 4 + 8 = 12 dari kota 5 ke 8 ke 10
- 8 + $f_4(9)$ = 8 + 4 = 12 dari kota 5 ke 9 ke 10

Ia dapat lihat bahwa biaya keduanya sama, artinya $f_3(5) = 12$ yaitu biaya optimal dari kota 5 di stage 3 ke kota tujuan akhir yaitu kota 10; sedangkan $d_3(5) = 8\ atau\ 9$ yaitu kota berikutnya yang harus ditempuh dari kota 5 untuk menuju kota tujuan terakhir (kota 10) harus dituju kota 8 atau kota 9.
Dengan cara yang sama, bila ia tiba di kota 6 dalam stage 3, maka nilai

$f_3(6) = \min\begin{pmatrix} 9 + f_4(8) \\ 6 + f_4(9) \end{pmatrix} = \min\begin{pmatrix} 9+8 \\ 6+4 \end{pmatrix} = 10$ dan $d_3(6) = 9$.

Akhirnya $f_3(7) = \min\begin{pmatrix} 5 + f_4(8) \\ 4 + f_4(9) \end{pmatrix} = \min\begin{pmatrix} 5+8 \\ 4+4 \end{pmatrix} = 8$ dan $d_3(7) = 9$

Sekali lagi, dalam setiap kasus $f_3(X)$ menunjukkan *policy cost* yang minimum untuk pergi dari kota X di stage 3 ke kota 10; $d_3(X)$ menunjukkan kota yang harus ditempuh guna mendapatkan *policy cost* minimum.

Perhatikan tabel berikut

X	**5**	**6**	**7**	**8**	**9**
$f_4(X)$				8	4
$d_4(X)$				10	10
$f_3(X)$	12	10	8		
$d_3(X)$	8 , 9	9	9		

Tahap 3

Sekarang apabila ia berada di stage 2 dimana secara teori ia dapat berada di kota 2, 3, atau 4. Bila ia berada di kota 2, misalnya, maka ia akan dapat menuju kota 5 atau 6 di stage 3 sebelum akhirnya sampai kota tujuan terakhir. Nilai *policy cost* minimum bila ia harus menuju kota 5 dahulu adalah $10 + f_3(5)$ = 10 + 12 = 22 atau bila ia harus menuju kota 6 dahulu adalah $9 + f_3(6)$ = 9 + 10 = 19. Dengan demikian

$$f_2(2) = \min\begin{pmatrix} 10 + f_3(5) \\ 9 + f_3(6) \end{pmatrix} = \min\begin{pmatrix} 10+12 \\ 9+10 \end{pmatrix} = 19$$ dan $d_2(2) = 6$.

Dengan cara yang sama, dapat dicari

$$f_2(3) = \min\begin{pmatrix} 6 + f_3(5) \\ 7 + f_3(6) \\ 10 + f_3(7) \end{pmatrix} = \min\begin{pmatrix} 6+12 \\ 7+10 \\ 10+8 \end{pmatrix} = 17$$ dengan

$d_2(3) = 6$

serta

$$f_2(4) = \min\begin{pmatrix} 3 + f_3(6) \\ 8 + f_3(7) \end{pmatrix} = \min\begin{pmatrix} 3+10 \\ 8+8 \end{pmatrix} = 13$$ dengan $d_2(4) = 6$

X	2	3	4	5	6	7	8	9
$f_4(X)$							8	4
$d_4(X)$							10	10
$f_3(X)$				12	10	8		
$d_3(X)$				8 , 9	9	9		
$f_2(X)$	19	17	13					
$d_2(X)$	6	6	6					

Tahap 4

Akhirnya, dari stage 1 dimana kota 1 berada, digunakan sebagai titik awal. Untuk menentukan *policy cost* yang minimum dari kota 1 ke kota 10, perlu melihat tiga jumlah berikut

- Biaya dari kota 1 ke kota 2 ditambah dengan biaya minimum dari kota 2 ke kota 10.
- Biaya dari kota 1 ke kota 3 ditambah dengan biaya minimum dari kota 3 ke kota 10.
- Biaya dari kota 1 ke kota 4 ditambah dengan biaya minimum dari kota 4 ke kota 10.

Sudah dihitung pada tahap terdahulu biaya minimum dari sembarang kota di stage 2 ke kota 10. Dengan demikian, kita dapatkan biaya minimum dari kota 1 ke kota 10 sebagai berikut

$$f_1(1) = \min\begin{pmatrix} C_{12} + f_2(2) \\ C_{13} + f_2(3) \\ C_{14} + f_2(4) \end{pmatrix} = \min\begin{pmatrix} 4+19 \\ 2+17 \\ 3+13 \end{pmatrix} = 16$$ dengan

$$d_1(1) = 4$$

Ringkasan lengkap *optimal policy cost* dari kota 1 ke kota 10 dapat disajikan pada tabel berikut

X	1	2	3	4	5	6	7	8	9
$f_4(X)$								8	4
$d_4(X)$								10	10
$f_3(X)$					12	10	8		
$d_3(X)$					8 , 9	9	9		
$f_2(X)$		19	17	13					
$d_2(X)$		6	6	6					
$f_1(X)$	16								
$d_1(X)$	4								

Dengan demikian *optimal policy cost* dari kota 1 ke kota 10 adalah 16 dengan *optimal route* adalah 1$\rightarrow$ 4$\rightarrow$ 6$\rightarrow$ 9$\rightarrow$ 10.

Program Komputer

Berikut ini dituliskan program untuk menyelesaikan masalah Stage Coach yang baru saja dipelajari, yang dituliskan dengan bahasa Basic.

```
REM ************************************************************************************
REM *   PROGRAM UNTUK MENYELESAIKAN STAGECOACH PROBLEM       *
REM *                                                        *
REM *         SESUAIKAN DIMENSI VARIABEL BILA DIPERLUKAN     *
REM *                                                        *
REM *              Ir. SIGIT NUGROHO, M.Sc., Ph.D.           *
REM ************************************************************************************
CLS
DIM C(7, 8, 8), F(7, 8), M(8)
DIM S(8, 8), D(7, 8), XSTAR(8)
REM INPUT "BANYAKNYA STAGE / TAHAP ="; N
```

```
PRINT "BANYAKNYA STAGE / TAHAP ="; : READ N
DATA 5
FOR I = 1 TO N
    REM PRINT "BANYAKNYA KOTA DI STAGE "; I; " ="; : INPUT M(I)
    PRINT "BANYAKNYA KOTA DI STAGE "; I; " ="; : READ M(I)
NEXT I
DATA 1, 3, 3, 2, 1
NM1 = N - 1
FOR I = 1 TO NM1
    MI = M(I)
    MIP1 = M(I + 1)
    FOR K = 1 TO MIP1
        FOR J = 1 TO MI
            REM PRINT " DARI KOTA "; J; " STAGE "; I; "KE KOTA"; K; "STAGE
"; I + 1; : INPUT C(I, J, K)
            PRINT " DARI KOTA "; J; " STAGE "; I; "KE KOTA"; K; "STAGE "; I +
1; : READ C(I, J, K)
        NEXT J
    NEXT K
NEXT I
DATA 4, 2, 3
DATA 10, 6, 1000
DATA 9, 7, 3
DATA 1000, 10, 8
DATA 4, 9, 5
DATA 8, 1000, 4
DATA 8, 4
FOR I = 1 TO N
    MI = M(I)
    FOR J = 1 TO MI
        REM PRINT "STAGE "; I; "KOTA "; J; : INPUT S(I, J)
        PRINT "STAGE "; I; "KOTA "; J; : READ S(I, J)
    NEXT J
NEXT I
DATA 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
I = N - 1
MI = M(I)
FOR J = 1 TO MI
    F(I, J) = C(I, J, 1)
    D(I, J) = S(N, 1)
NEXT J
NM2 = N - 2
FOR II = 1 TO NM2
    I = N - II - 1
    MI = M(I)
    MIP1 = M(I + 1)
    FOR J = 1 TO MI
```

```
                  F(I, J) = C(I, J, 1) + F(I + 1, 1)
                  D(I, J) = S(I + 1, 1)
                  IF MIP1 = 1 GOTO 2
                  FOR K = 2 TO MIP1
                        IF F(I, J) <= C(I, J, K) + F(I + 1, K) GOTO 1
                        F(I, J) = C(I, J, K) + F(I + 1, K)
                        D(I, J) = S(I + 1, K)
1                  NEXT K
2            NEXT J
      NEXT II
      XSTAR(1) = S(1, 1)
      XSTAR(2) = D(1, 1)
      KK = 1
      FOR I = 2 TO NM1
            MI = M(I)
            FOR J = 1 TO MI
                  IF S(I, J) <> D(I - 1, KK) GOTO 3
                  XSTAR(I + 1) = D(I, J)
                  KK = J
                  GOTO 4
3            NEXT J
4      NEXT I
      CLS
      PRINT "BANYAK STAGE / TAHAPAN ="; N
REM       FOR I = 1 TO N
REM             MI = M(I)
REM             FOR J = 1 TO MI
REM                   PRINT I, S(I, J); :
REM             NEXT J
REM       NEXT I
REM     FOR I = 1 TO NM1
REM             MI = M(I)
REM             MIP1 = M(I + 1)
REM             FOR J = 1 TO MI
REM                   FOR K = 1 TO MIP1
REM                         PRINT I, J
REM                         PRINT C(I, J, K); :
REM                   NEXT K
REM             NEXT J
REM       NEXT I
      PRINT
      PRINT "URUTAN KOTA YANG DILALUI"
      FOR I = 1 TO N
            PRINT XSTAR(I); :
      NEXT I
      PRINT
      PRINT "NILAI OPTIMAL"; F(1, 1)
```

```
PRINT
END
```

Latihan

1. Dengan menggunakan data yang sama, carilah *policy cost* minimum bila kita ingin pergi dari kota 1 ke kota 10 dan balik lagi ke kota 1, apabila diketahui pada saat berangkat jalur dari kota 6 ke kota 9 terputus akibat banjir, dan pada saat pulang jalur dari kota 6 ke kota 4 terputus karena tanah longsor, sedangkan dari kota 9 ke kota 6 sudah pulih kembali.

2. Berikut ini adalah tabel keuntungan yang dapat diperoleh akibat pengoperasian kendaraan umum dari satu kota ke kota lainnya.

Dari Kota	**Ke Kota**										
	2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	5	4	2								
2				8	10	5	7				
3				6	3	8	10				
4				8	9	6	4				
5								8	4	3	
6								5	2	7	
7								4	10	6	
8								12	5	2	
9											7
10											3
11											6

Cari jalur yang paling menguntungkan bila berangkat dari kota 1 ke kota 12.

Penjadwalan Produksi

Dalam tulisan ini akan disajikan masalah inventori dalam penjadwalan produksi untuk barang tertentu selama N periode ke depan, dimana terdapat permintaan terhadap barang tersebut selama tiap periode. Untuk ilustrasi, dapat diasumsikan bahwa waktu pembuatan barang dapat diabaikan; namun demikian, biaya produksi semakin naik sejalan dengan naiknya barang yang diproduksi pada periode tersebut. Kelebihan barang yang diproduksi pada periode tersebut harus disimpan sebagai inventori, yang sudah tentu dikenai biaya inventori.

Tujuan dari permasalahan ini adalah untuk menentukan penjadwalan produksi yang akan meminimumkan biaya produksi total dan biaya inventori dengan tetap memperhatikan bahwa semua permintaan harus dipenuhi tepat waktu dan inventori di akhir periode ke N adalah nol. Kita asumsikan bahwa jumlah barang dalam inventori di akhir periode ke I-1 ditambah dengan barang yang diproduksi dalam periode ke I dapat digunakan untuk memenuhi permintaan periode ke I tersebut. Juga diasumsikan bahwa biaya inventori untuk periode ke I tergantung dari banyaknya barang yang harus diinventori di akhir periode ke I.

Permasalahan

Perhatikan permasalahan berikut. Misalkan

n = banyaknya periode = 6

I0 = inventori di awal periode ke-1 = 0

IN = inventori di akhir periode ke-n = 0

$BP_i(j) = \begin{cases} 0 \quad untuk\, j = 0\ dan\ i = 1,2,...,n \\ 20 + 5j \quad untuk\ j = 1,2,...\ dan\ i = 1,2,...,n \end{cases}$ = Biaya untuk memproduksi j satuan barang dalam periode ke i.

$BI_i(j) = j\ untuk\ j = 1,2,... \quad dan \quad i = 1,2,...,n$ = Biaya inventori sebanyak j satuan barang di akhir periode ke i.

Sedangkan tabel permintaan barang untuk 6 periode ke depan adalah sebagai berikut

Periode (i)	1	2	3	4	5	6
Permintaan (d_i)	8	4	6	2	10	4

Tentukan jumlah barang yang harus diproduksi di awal setiap periode yang meminimumkan biaya total (*policy cost*)untuk 6 periode.

Pemecahan

Pendekatan pemrograman dinamis mengasumsikan bahwa proses telah mencapai awal periode ke n atau periode ke 6 dengan sejumlah inventori, dan kemudian, berdasarkan permintaan pada periode terakhir ini, digunakan untuk menentukan banyaknya barang yang harus diproduksi sekaligus dapat memenuhi permintaan barang pada periode terakhir ini dan tidak ada lagi barang yang harus masuk inventori.

Periode terakhir memiliki permintaan sebanyak 4 barang. Berapa banyak kemungkinan barang sebagai inventori di akhir periode ke n-1 ? Tentunya tidak mungkin kurang dari nol dan paling banyak adalah sebanyak permintaan pada periode ke-n tersebut, yaitu 4. Dengan demikian, misalkan

$$f_n(K) = \begin{cases} 0 \quad untuk \quad K = d_n \\ 20 + 5(d_n - K) \quad untuk \quad K = 0,1,2,\ldots,d_n - 1 \end{cases}$$

merupakan biaya kebijakan minimum periode ke n saja, apabila inventori dari periode sebelumnya sebanyak K satuan barang. Misalkan juga

$$x_n(K) = \begin{cases} 0 \quad untuk \quad K = d_n \\ d_n - K \quad untuk \quad K = 0,1,2,\ldots,d_n - 1 \end{cases}$$

merupakan jumlah optimal barang yang harus diproduksi di awal periode ke n atau terakhir apabila inventori dari periode sebelumnya adalah K satuan barang.

Karena permintaan pada periode terakhir adalah sebanyak 4 satuan barang, maka kemungkinan nilai K adalah sedikitnya 0 dan sebanyak-banyaknya sama dengan permintaan barang pada periode tersebut atau 4. Dengan demikian kita dapat membuat tabel seperti berikut:

Inventori periode 5 K	$f_6(K)$	$x_6(K)$
0	40	4
1	35	3
2	30	2
3	25	1
4	0	0

Langkah berikutnya adalah apabila kita berada di awal periode ke 5, dan dengan mengasumsikan bahwa kita hanya memperhatikan periode ke 5 dan ke 6 saja. Masalahnya sekarang menjadi berapa banyak barang yang harus diproduksi di awal periode ke 5 untuk meminimumkan biaya kebijakan (biaya produksi dan biaya inventori) pada periode ke 5 dan ke 6. Jika diawal periode ke 5 sudah terdapat K barang tersedia sebagai hasil inventori dari periode sebelumnya (periode ke 4), dimana K lebih kecil dari permintaan periode ke 5 (d_5 = 10), dengan demikian sedikitnya d_5-K atau 10-K satuan barang harus diproduksi; bila tidak demikian, jumlah minimum barang yang harus diproduksi adalah nol. Sejalan dengan itu, apabila sebanyak K satuan barang dari inventori pada periode sebelumnya, maka jumlah maksimum barang yang harus diproduksi untuk memenuhi permintaan sehingga di akhir periode ke 6 tak ada barang yang harus diinventori, adalah jumlah dari permintaan pada periode ke 5 dan periode ke 6 dikurangi dengan K. Sebagai konsekuensinya, jumlah optimal yang harus diproduksi di awal periode ke 5, jika inventori dari periode sebelumnya adalah K satuan barang, adalah nilai z yang menghasilkan f_5(K), dimana

$$f_5(K) = \min_{\max(0, d_5 - K) \le z \le d_5 + d_6 - K} \left[BP_5(z) + BI_5(K + z - d_5) + f_6(K + z - d_5) \right]$$

dimana $K = 0,1,2,\ldots,d_5 + d_6$

atau

$$f_5(K) = \min_{\max(0, 10 - K) \le z \le 10 + 4 - K} \left[BP_5(z) + BI_5(K + z - 10) + f_6(K + z - 10) \right]$$

dimana $K = 0,1,2,\ldots,14$

Sebagai contoh, untuk K = 0, maka

$$f_5(0) = \min_{10 \le z \le 14} \left[20 + 5z + (z-10) + f_6(z-10) \right]$$

$$= \min \begin{pmatrix} z = 10; & 20 + 50 + 0 + 40 = 110 \\ z = 11; & 20 + 55 + 1 + 35 = 111 \\ z = 12; & 20 + 60 + 2 + 30 = 112 \\ z = 13; & 20 + 65 + 3 + 25 = 113 \\ z = 14; & 20 + 70 + 4 + 0 = 94 \end{pmatrix} = 94$$

dan $x_5(0) = 14$ adalah nilai z yang memenuhi kriteria bila K = 0. Hal ini berarti bahwa bila tak ada inventori dari periode ke 4, maka biaya kebijakan minimum untuk periode ke 5 dan ke 6 adalah 94 satuan harga, dan banyaknya barang yang harus diproduksi adalah sebanyak 14 satuan barang.

Lagi, untuk K = 1 misalnya

$$f_5(1) = \min_{9 \le z \le 13} \left[20 + 5z + (z-9) + f_6(z-9) \right]$$

$$= \min \begin{pmatrix} z = 9; & 20 + 45 + 0 + 40 = 105 \\ z = 10; & 20 + 50 + 1 + 35 = 106 \\ z = 11; & 20 + 55 + 2 + 30 = 107 \\ z = 12; & 20 + 60 + 3 + 25 = 108 \\ z = 13; & 20 + 65 + 4 + 0 = 89 \end{pmatrix} = 89$$

dan $x_5(1) = 13$ adalah nilai z yang memenuhi kriteria bila K = 1. Hal ini berarti bahwa bila terdapat satu satuan barang dari inventori periode ke 4, maka biaya kebijakan minimum untuk periode ke 5 dan ke 6 adalah 89 satuan harga, dan banyaknya barang yang harus diproduksi adalah sebanyak 13 satuan barang.

Untuk K = 2 misalnya

$$f_5(2) = \min_{8 \le z \le 12} \left[20 + 5z + (z-8) + f_6(z-8) \right]$$

$$= \min\begin{pmatrix} z=8; & 20+40+0+40=100 \\ z=9; & 20+45+1+35=101 \\ z=10; & 20+50+2+30=102 \\ z=11; & 20+55+3+25=103 \\ z=12; & 20+60+4+0=84 \end{pmatrix} = 84$$

dan $x_5(2)=12$ adalah nilai z yang memenuhi kriteria bila K = 2. Hal ini berarti bahwa bila terdapat 2 satuan barang dari inventori periode ke 4, maka biaya kebijakan minimum untuk periode ke 5 dan ke 6 adalah 84 satuan harga, dan banyaknya barang yang harus diproduksi adalah sebanyak 12 satuan barang.

Untuk K = 3 misalnya

$$f_5(3) = \min_{7 \le z \le 11}\left[20+5z+(z-7)+f_6(z-7)\right]$$

$$= \min\begin{pmatrix} z=7; & 20+35+0+40=95 \\ z=8; & 20+40+1+35=96 \\ z=9; & 20+45+2+30=97 \\ z=10; & 20+50+3+25=98 \\ z=11; & 20+55+4+0=79 \end{pmatrix} = 79$$

dan $x_5(3)=11$ adalah nilai z yang memenuhi kriteria bila K = 3. Hal ini berarti bahwa bila terdapat 3 satuan barang dari inventori periode ke 4, maka biaya kebijakan minimum untuk periode ke 5 dan ke 6 adalah 79 satuan harga, dan banyaknya barang yang harus diproduksi adalah sebanyak 11 satuan barang.

Untuk K = 4 misalnya

$$f_5(4) = \min_{6 \le z \le 10}\left[20+5z+(z-6)+f_6(z-6)\right]$$

$$= \min \begin{pmatrix} z = 6; & 20 + 30 + 0 + 40 = 90 \\ z = 7; & 20 + 35 + 1 + 35 = 91 \\ z = 8; & 20 + 40 + 2 + 30 = 92 \\ z = 9; & 20 + 45 + 3 + 25 = 93 \\ z = 10; & 20 + 50 + 4 + 0 = 74 \end{pmatrix} = 74$$

dan $x_5(4) = 10$ adalah nilai z yang memenuhi kriteria bila K = 4. Hal ini berarti bahwa bila terdapat 4 satuan barang dari inventori periode ke 4, maka biaya kebijakan minimum untuk periode ke 5 dan ke 6 adalah 74 satuan harga, dan banyaknya barang yang harus diproduksi adalah sebanyak 10 satuan barang.

Untuk K = 5 misalnya

$$f_5(5) = \min_{5 \le z \le 9} \left[20 + 5z + (z - 5) + f_6(z - 5) \right]$$

$$= \min \begin{pmatrix} z = 5; & 20 + 25 + 0 + 40 = 85 \\ z = 6; & 20 + 30 + 1 + 35 = 86 \\ z = 7; & 20 + 35 + 2 + 30 = 87 \\ z = 8; & 20 + 40 + 3 + 25 = 88 \\ z = 9; & 20 + 45 + 4 + 0 = 69 \end{pmatrix} = 69$$

dan $x_5(5) = 9$ adalah nilai z yang memenuhi kriteria bila K = 5. Hal ini berarti bahwa bila terdapat 5 satuan barang dari inventori periode ke 4, maka biaya kebijakan minimum untuk periode ke 5 dan ke 6 adalah 69 satuan harga, dan banyaknya barang yang harus diproduksi adalah sebanyak 9 satuan barang.

Untuk K = 6 misalnya

$$f_5(6) = \min_{4 \le z \le 8} \left[20 + 5z + (z - 4) + f_6(z - 4) \right]$$

$$= \min \begin{pmatrix} z = 4; & 20 + 20 + 0 + 40 = 80 \\ z = 5; & 20 + 25 + 1 + 35 = 81 \\ z = 6; & 20 + 30 + 2 + 30 = 82 \\ z = 7; & 20 + 35 + 3 + 25 = 83 \\ z = 8; & 20 + 40 + 4 + 0 = 64 \end{pmatrix} = 64$$

dan $x_5(6) = 8$ adalah nilai z yang memenuhi kriteria bila K = 6. Hal ini berarti bahwa bila terdapat 6 satuan barang dari inventori periode ke 4, maka biaya kebijakan minimum untuk periode ke 5 dan ke 6 adalah 64 satuan harga, dan banyaknya barang yang harus diproduksi adalah sebanyak 8 satuan barang.

Untuk K = 7 misalnya

$$f_5(7) = \min_{3 \le z \le 7}\left[20 + 5z + (z-3) + f_6(z-3)\right]$$

$$= \min \begin{pmatrix} z = 3; & 20 + 15 + 0 + 40 = 75 \\ z = 4; & 20 + 20 + 1 + 35 = 76 \\ z = 5; & 20 + 25 + 2 + 30 = 77 \\ z = 6; & 20 + 30 + 3 + 25 = 78 \\ z = 7; & 20 + 35 + 4 + 0 = 59 \end{pmatrix} = 59$$

dan $x_5(7) = 7$ adalah nilai z yang memenuhi kriteria bila K = 7. Hal ini berarti bahwa bila terdapat 7 satuan barang dari inventori periode ke 4, maka biaya kebijakan minimum untuk periode ke 5 dan ke 6 adalah 59 satuan harga, dan banyaknya barang yang harus diproduksi adalah sebanyak 7 satuan barang.

Untuk K = 8 misalnya

$$f_5(8) = \min_{2 \le z \le 6}\left[20 + 5z + (z-2) + f_6(z-2)\right]$$

$$= \min \begin{pmatrix} z = 2; & 20 + 10 + 0 + 40 = 70 \\ z = 3; & 20 + 15 + 1 + 35 = 71 \\ z = 4; & 20 + 20 + 2 + 30 = 72 \\ z = 5; & 20 + 25 + 3 + 25 = 73 \\ z = 6; & 20 + 30 + 4 + 0 = 54 \end{pmatrix} = 54$$

dan $x_5(8) = 6$ adalah nilai z yang memenuhi kriteria bila K = 8. Hal ini berarti bahwa bila terdapat 8 satuan barang dari inventori periode ke 4, maka biaya kebijakan minimum untuk periode ke 5 dan ke 6 adalah 54 satuan harga, dan banyaknya barang yang harus diproduksi adalah sebanyak 6 satuan barang.

Untuk K = 9 misalnya

$$f_5(9) = \min_{1 \le z \le 5} \left[20 + 5z + (z-1) + f_6(z-1) \right]$$

$$= \min \begin{pmatrix} z = 1; & 20 + 5 + 0 + 40 = 65 \\ z = 2; & 20 + 10 + 1 + 35 = 66 \\ z = 3; & 20 + 15 + 2 + 30 = 67 \\ z = 4; & 20 + 20 + 3 + 25 = 68 \\ z = 5; & 20 + 25 + 4 + 0 = 49 \end{pmatrix} = 49$$

dan $x_5(9) = 5$ adalah nilai z yang memenuhi kriteria bila K = 9. Hal ini berarti bahwa bila terdapat 9 satuan barang dari inventori periode ke 4, maka biaya kebijakan minimum untuk periode ke 5 dan ke 6 adalah 49 satuan harga, dan banyaknya barang yang harus diproduksi adalah sebanyak 5 satuan barang.

Untuk K = 10 misalnya

$$f_5(10) = \min_{0 \le z \le 4} \left[20 + 5z + z + f_6(z) \right]$$

$$= \min \begin{pmatrix} z = 0; & 0 + 0 + 0 + 40 = 40 \\ z = 1; & 20 + 5 + 1 + 35 = 61 \\ z = 2; & 20 + 10 + 2 + 30 = 62 \\ z = 3; & 20 + 15 + 3 + 25 = 63 \\ z = 4; & 20 + 20 + 4 + 0 = 44 \end{pmatrix} = 40$$

dan $x_5(10) = 0$ adalah nilai z yang memenuhi kriteria bila K = 10. Hal ini berarti bahwa bila terdapat 10 satuan barang dari inventori dari periode ke 4, maka biaya kebijakan minimum untuk periode ke 5 dan ke 6 adalah 40 satuan harga, dan tak ada barang yang harus diproduksi.

Untuk K = 11 misalnya

$$f_5(11) = \min_{0 \le z \le 3} \left[20 + 5z + (z+1) + f_6(z+1) \right]$$

$$= \min \begin{pmatrix} z = 0; & 0 + 0 + 1 + 35 = 36 \\ z = 1; & 20 + 5 + 2 + 30 = 57 \\ z = 2; & 20 + 10 + 3 + 25 = 58 \\ z = 3; & 20 + 15 + 4 + 0 = 39 \end{pmatrix} = 36$$

dan $x_5(11) = 0$ adalah nilai z yang memenuhi kriteria bila K = 11. Hal ini berarti bahwa bila terdapat 11 satuan barang dari inventori dari periode ke 4, maka biaya kebijakan minimum untuk periode ke 5 dan ke 6 adalah 36 satuan harga, dan tak ada barang yang harus diproduksi.

Untuk K = 12 misalnya

$$f_5(12) = \min_{0 \le z \le 2} \left[20 + 5z + (z+2) + f_6(z+2) \right]$$

$$= \min \begin{pmatrix} z = 0; & 0 + 0 + 2 + 30 = 32 \\ z = 1; & 20 + 5 + 3 + 25 = 53 \\ z = 2; & 20 + 10 + 4 + 0 = 34 \end{pmatrix} = 32$$

dan $x_5(12) = 0$ adalah nilai z yang memenuhi kriteria bila K = 12. Hal ini berarti bahwa bila terdapat 12 satuan barang dari inventori dari periode ke 4, maka biaya kebijakan minimum untuk periode ke 5 dan ke 6 adalah 32 satuan harga, dan tak ada barang yang harus diproduksi.

Untuk K = 13 misalnya

$$f_5(13) = \min_{0 \le z \le 1}\left[20 + 5z + (z+3) + f_6(z+3)\right]$$

$$= \min\begin{pmatrix} z = 0; & 0+0+3+25 = 28 \\ z = 1; & 20+5+4+0 = 29 \end{pmatrix} = 28$$

dan $x_5(13) = 0$ adalah nilai z yang memenuhi kriteria bila K = 13. Hal ini berarti bahwa bila terdapat 13 satuan barang dari inventori dari periode ke 4, maka biaya kebijakan minimum untuk periode ke 5 dan ke 6 adalah 28 satuan harga, dan tak ada barang yang harus diproduksi.

Untuk K = 14 misalnya

$$f_5(14) = \min_{z=0}\left[20 + 5z + (z+4) + f_6(z+4)\right]$$

$$= \min\left(z = 0; \quad 0+0+4+0 = 4\right) = 4$$

dan $x_5(14) = 0$ adalah nilai z yang memenuhi kriteria bila K = 14. Hal ini berarti bahwa bila terdapat 14 satuan barang dari inventori dari periode ke 4, maka biaya kebijakan minimum untuk periode ke 5 dan ke 6 adalah 4 satuan harga, dan tak ada barang yang harus diproduksi.

Dengan demikian kita dapat membuat tabel seperti berikut:

Inventori periode 4 K	$f_5(K)$	$x_5(K)$
0	94	14
1	89	13
2	84	12
3	79	11
4	74	10
5	69	9
6	64	8
7	59	7
8	54	6
9	49	5
10	40	0
11	36	0
12	32	0
13	28	0
14	4	0

Langkah berikutnya adalah apabila kita berada di awal periode ke 4, dan dengan mengasumsikan bahwa kita hanya memperhatikan periode ke 4, ke 5 dan ke 6 saja. Masalahnya sekarang menjadi berapa banyak barang yang harus diproduksi di awal periode ke 4 untuk meminimumkan biaya kebijakan (biaya produksi dan biaya inventori) pada periode ke 4, ke 5 dan ke 6. Jika diawal periode ke 4 sudah terdapat K barang tersedia sebagai hasil inventori dari periode sebelumnya (periode ke 3), dimana K lebih kecil dari permintaan periode ke 4 (d_4 = 2), dengan demikian sedikitnya d_4-K atau 2-K satuan barang harus diproduksi; bila tidak demikian, jumlah minimum barang yang harus diproduksi adalah nol. Sejalan dengan itu, apabila sebanyak K satuan barang dari inventori pada periode sebelumnya, maka jumlah maksimum barang yang harus diproduksi untuk memenuhi permintaan sehingga di akhir periode ke 6 tak ada barang yang harus diinventori, adalah jumlah dari permintaan pada periode ke 4, ke 5 dan periode ke 6 dikurangi dengan K. Sebagai konsekuensinya, jumlah optimal yang harus diproduksi di awal periode ke 4, jika inventori dari periode sebelumnya adalah K satuan barang, adalah nilai z yang menghasilkan f_4(K), dimana

$$f_4(K) = \min_{\max(0, d_4 - K) \le z \le d_4 + d_5 + d_6 - K} \left[BP_4(z) + BI_4(K + z - d_4) + f_5(K + z - d_4) \right]$$

dimana $K = 0,1,2,\ldots,d_4 + d_5 + d_6$

atau

$$f_4(K) = \min_{\max(0, 2-K) \le z \le 2+10+4-K} \left[BP_4(z) + BI_4(K + z - 2) + f_5(K + z - 2) \right]$$

dimana $K = 0,1,2,\ldots,16$

Sebagai contoh, untuk K = 0, maka

$$f_4(0) = \min_{2 \le z \le 16} \left[BP_4(z) + BI_4(0 + z - 2) + f_5(0 + z - 2) \right]$$

$$= \min \begin{pmatrix} z=2;\ 20+5(2)+(0+2-2)+f_5(0+2-2) \\ z=3;\ 20+5(3)+(0+3-2)+f_5(0+3-2) \\ z=4;\ 20+5(4)+(0+4-2)+f_5(0+4-2) \\ z=5;\ 20+5(5)+(0+5-2)+f_5(0+5-2) \\ z=6;\ 20+5(6)+(0+6-2)+f_5(0+6-2) \\ z=7;\ 20+5(7)+(0+7-2)+f_5(0+7-2) \\ z=8;\ 20+5(8)+(0+8-2)+f_5(0+8-2) \\ z=9;\ 20+5(9)+(0+9-2)+f_5(0+9-2) \\ z=10;\ 20+5(10)+(0+10-2)+f_5(0+10-2) \\ z=11;\ 20+5(11)+(0+11-2)+f_5(0+11-2) \\ z=12;\ 20+5(12)+(0+12-2)+f_5(0+12-2) \\ z=13;\ 20+5(13)+(0+13-2)+f_5(0+13-2) \\ z=14;\ 20+5(14)+(0+14-2)+f_5(0+14-2) \\ z=15;\ 20+5(15)+(0+15-2)+f_5(0+15-2) \\ z=16;\ 20+5(16)+(0+16-2)+f_5(0+16-2) \end{pmatrix}$$

$$= \min \begin{pmatrix} z = 2;\ 20 + 10 + 0 + 94 = 124 \\ z = 3;\ 20 + 15 + 1 + 89 = 125 \\ z = 4;\ 20 + 20 + 2 + 84 = 126 \\ z = 5;\ 20 + 25 + 3 + 79 = 127 \\ z = 6;\ 20 + 30 + 4 + 74 = 128 \\ z = 7;\ 20 + 35 + 5 + 69 = 129 \\ z = 8;\ 20 + 40 + 6 + 64 = 130 \\ z = 9;\ 20 + 45 + 7 + 59 = 131 \\ z = 10;\ 20 + 50 + 8 + 54 = 132 \\ z = 11;\ 20 + 55 + 9 + 49 = 133 \\ z = 12;\ 20 + 60 + 10 + 40 = 130 \\ z = 13;\ 20 + 65 + 11 + 36 = 131 \\ z = 14;\ 20 + 70 + 12 + 32 = 134 \\ z = 15;\ 20 + 75 + 13 + 28 = 136 \\ z = 16;\ 20 + 80 + 14 + 4 = 118 \end{pmatrix} = 118$$

dan $x_4(0) = 16$ adalah nilai z yang memenuhi kriteria bila K = 0. Hal ini berarti bahwa bila tak ada inventori dari periode ke 4, maka biaya kebijakan minimum untuk periode ke 4, ke 5 dan ke 6 adalah 118 satuan harga, dan banyaknya barang yang harus diproduksi adalah sebanyak 16 satuan barang.

Untuk K = 1,

maka $f_4(1) = \min\limits_{1 \le z \le 15} \left[BP_4(z) + BI_4(1 + z - 2) + f_5(1 + z - 2) \right]$

$$= \min \begin{pmatrix} z = 1;\ 20 + 5(1) + (1 + 1 - 2) + f_5(1 + 1 - 2) \\ z = 2;\ 20 + 5(2) + (1 + 2 - 2) + f_5(1 + 2 - 2) \\ z = 3;\ 20 + 5(3) + (1 + 3 - 2) + f_5(1 + 3 - 2) \\ z = 4;\ 20 + 5(4) + (1 + 4 - 2) + f_5(1 + 4 - 2) \\ z = 5;\ 20 + 5(5) + (1 + 5 - 2) + f_5(1 + 5 - 2) \\ z = 6;\ 20 + 5(6) + (1 + 6 - 2) + f_5(1 + 6 - 2) \\ z = 7;\ 20 + 5(7) + (1 + 7 - 2) + f_5(1 + 7 - 2) \\ z = 8;\ 20 + 5(8) + (1 + 8 - 2) + f_5(1 + 8 - 2) \\ z = 9;\ 20 + 5(9) + (1 + 9 - 2) + f_5(1 + 9 - 2) \\ z = 10;\ 20 + 5(10) + (1 + 10 - 2) + f_5(1 + 10 - 2) \\ z = 11;\ 20 + 5(11) + (1 + 11 - 2) + f_5(1 + 11 - 2) \\ z = 12;\ 20 + 5(12) + (1 + 12 - 2) + f_5(1 + 12 - 2) \\ z = 13;\ 20 + 5(13) + (1 + 13 - 2) + f_5(1 + 13 - 2) \\ z = 14;\ 20 + 5(14) + (1 + 14 - 2) + f_5(1 + 14 - 2) \\ z = 15;\ 20 + 5(15) + (1 + 15 - 2) + f_5(1 + 15 - 2) \end{pmatrix}$$

$$= \min \begin{pmatrix} z = 1;\ 20 + 5 + 0 + 94 = 119 \\ z = 2;\ 20 + 10 + 1 + 89 = 120 \\ z = 3;\ 20 + 15 + 2 + 84 = 121 \\ z = 4;\ 20 + 20 + 3 + 79 = 122 \\ z = 5;\ 20 + 25 + 4 + 74 = 123 \\ z = 6;\ 20 + 30 + 5 + 69 = 124 \\ z = 7;\ 20 + 35 + 6 + 64 = 125 \\ z = 8;\ 20 + 40 + 7 + 59 = 126 \\ z = 9;\ 20 + 45 + 8 + 54 = 127 \\ z = 10;\ 20 + 50 + 9 + 49 = 128 \\ z = 11;\ 20 + 55 + 10 + 40 = 125 \\ z = 12;\ 20 + 60 + 11 + 36 = 126 \\ z = 13;\ 20 + 65 + 12 + 32 = 129 \\ z = 14;\ 20 + 70 + 13 + 28 = 131 \\ z = 15;\ 20 + 75 + 14 + 4 = 113 \end{pmatrix} = 113$$

dan $x_4(1) = 15$ adalah nilai z yang memenuhi kriteria bila K = 1. Hal ini berarti bahwa bila terdapat satu satuan barang dari inventori periode ke 3, maka biaya kebijakan minimum untuk periode ke 4, ke 5 dan ke 6 adalah 113 satuan harga, dan banyaknya barang yang harus diproduksi adalah sebanyak 15 satuan barang.

Untuk K = 2,
maka $f_4(2) = \min\limits_{0 \le z \le 14} \left[BP_4(z) + BI_4(2 + z - 2) + f_5(2 + z - 2) \right]$

$$
= \min \begin{pmatrix}
z = 0;\ 0 + 0 + 0 + 94 = 94 \\
z = 1;\ 20 + 5 + 1 + 89 = 115 \\
z = 2;\ 20 + 10 + 2 + 84 = 116 \\
z = 3;\ 20 + 15 + 3 + 79 = 117 \\
z = 4;\ 20 + 20 + 4 + 74 = 118 \\
z = 5;\ 20 + 25 + 5 + 69 = 119 \\
z = 6;\ 20 + 30 + 6 + 64 = 120 \\
z = 7;\ 20 + 35 + 7 + 59 = 121 \\
z = 8;\ 20 + 40 + 8 + 54 = 122 \\
z = 9;\ 20 + 45 + 9 + 49 = 123 \\
z = 10;\ 20 + 50 + 10 + 40 = 120 \\
z = 11;\ 20 + 55 + 11 + 36 = 122 \\
z = 12;\ 20 + 60 + 12 + 32 = 124 \\
z = 13;\ 20 + 65 + 13 + 28 = 126 \\
z = 14;\ 20 + 70 + 14 + 4 = 108
\end{pmatrix} = 94
$$

dan $x_4(2) = 0$ adalah nilai z yang memenuhi kriteria bila K = 2. Hal ini berarti bahwa bila terdapat 2 satuan barang dari inventori periode ke 3, maka biaya kebijakan minimum untuk periode ke 4, ke 5 dan ke 6 adalah 94 satuan harga, dan tak ada barang yang harus diproduksi.

Generalisasi

Generalisasi prosedur dapat terus dilakukan secara sekuensial. Untuk itu diperlukan rumusan tentang $f_i(K)$, yaitu biaya kebijakan minimum apabila terdapat K satuan barang yang berasal dari inventori periode ke (i-1) dengan z satuan barang harus diproduksi pada periode ke i, sehingga di akhir periode ke-n atau periode terakhir tak ada barang yang harus dimasukkan sebagai inventori.

$$f_i(K) = \min_{\max(0, d_5 - K) \le z \le \sum_{j=i}^{6} d_j - K} \left[BP_i(z) + BI_i(K + z - d_i) + f_{i+1}(K + z - d_i) \right]$$

dan

$$x_i(K) \text{ = nilai z yang menghasilkan } f_i(K)$$

Dari tabel rangkuman diperoleh bahwa dengan memperhatikan seluruh periode permintaan, maka biaya kebijakan minimum dari permasalahan tersebut diatas adalah sebesar 236 satuan biaya, dengan rincian barang yang harus diproduksi di awal setiap periode adalah sebagai berikut

$x_1(0) = 20$ dengan $d_1 = 8$
$x_2(20-d_1) = x_2(20-8) = x_2(12) = 0$ dengan $d_2 = 4$
$x_3(20-d_1-d_2) = x_3(20-8-4) = x_3(8) = 0$ dengan $d_3 = 6$
$x_4(20-d_1-d_2-d_3) = x_4(20-8-4-6) = x_4(2) = 0$ dengan $d_4 = 2$
$x_5(20-d_1-d_2-d_3-d_4) = x_5(20-8-4-6-2) = x_5(0) = 14$ dengan $d_5 = 10$
$x_6(14-d_5) = x_6(14-10) = x_6(4) = 0$

Rangkuman pengolahan biaya kebijakan optimum tiap periode dan banyaknya barang yang harus diproduksi tiap periode, bila K inventori berasal dari periode sebelumnya.

K	$f_6(K)$	$x_6(K)$	$f_5(K)$	$x_5(K)$	$f_4(K)$	$x_4(K)$	$f_3(K)$	$x_3(K)$	$f_2(K)$	$x_2(K)$
0	40	4	94	14	118	16	156	8	184	12
1	35	3	89	13	113	15	151	7	179	11
2	30	2	84	12	94	0	146	6	174	10
3	25	1	79	11	90	0	141	5	169	9
4	0	0	74	10	86	0	136	4	156	0
5	0	0	69	9	82	0	131	3	152	0
6	0	0	64	8	78	0	118	0	148	0
7	0	0	59	7	74	0	114	0	144	0
8	0	0	54	6	70	0	96	0	140	0
9	0	0	49	5	66	0	93	0	136	0
10	0	0	40	0	62	0	90	0	124	0
11	0	0	36	0	58	0	87	0	121	0
12	0	0	32	0	50	0	84	0	104	0
13	0	0	28	0	47	0	81	0	102	0
14	0	0	4	0	44	0	78	0	100	0
15	0	0	0	0	41	0	75	0	98	0
16	0	0	0	0	18	0	72	0	96	0
17	0	0	0	0	0	0	69	0	94	0
18	0	0	0	0	0	0	62	0	92	0

19	0	0	0	0	0	0	60	0	90	0
20	0	0	0	0	0	0	58	0	88	0
21	0	0	0	0	0	0	56	0	86	0
22	0	0	0	0	0	0	34	0	80	0
23	0	0	0	0	0	0	0	0	79	0
24	0	0	0	0	0	0	0	0	78	0
25	0	0	0	0	0	0	0	0	77	0
26	0	0	0	0	0	0	0	0	56	0
27	0	0	0	0	0	0	0	0	0	0

$f_1(0) = 236$ dan $x_1(0) = 20$

Program Komputer

Berikut ini dituliskan program untuk menyelesaikan masalah Penjadwalan Produksi yang baru saja dipelajari, yang dituliskan dengan bahasa Basic.

```
REM *****************************************************************************************************
REM *  PROGRAM PENJADWALAN PRODUKSI UNTUK MEMINIMUMKAN BIAYA PRODUKSI   *
REM *  DAN BIAYA INVENTORI SELAMA PERIODE TERTENTU DIMANA DI AKHIR PE            *
REM *  RIODE TAK ADA BARANG YANG MASUK INVENTORI.  SESUAIKAN FUNGSI              *
REM *  PRODUKSI P1(I,J) DAN BIAYA INVENTORI E1(I,J) DALAM PROGRAM INI                *
REM *                -= Ir. SIGIT NUGROHO, M.Sc., Ph.D. =-                                          *
REM *****************************************************************************************************
     CLS
     INPUT "JUMLAH PERIODE ="; N
     INPUT "MAKSIMUM INVENTORI ="; M1
     INPUT "MAKSIMUM PRODUKSI ="; M2
     INPUT "INVENTORI AWAL PERIODE PERTAMA ="; I0
     PRINT : PRINT
     DIM D(N), F(N, M1 + 1), X(N, M1 + 1), X1(N), P1(N, M2 + 2), E1(N, M2 + 1)
     FOR I = 1 TO N
       FOR J = 1 TO M2 + 1
         P1(I, J) = 8 + 3 * J
         E1(I, J) = .5 * J
       NEXT J
       P1(I, 0) = 0
       PRINT "PERMINTAAN KE"; I; : INPUT D(I)
     NEXT I
     M3 = M2 + 1
     N1 = N - 1
     M4 = M1 + 1
     D1 = D(N) + 1
     D2 = D(N)
     IF D(N) = 0 GOTO 240
     FOR K = 1 TO D2
       F(N, K) = P1(N, D(N) - K + 1)
       X(N, K) = D(N) - K + 1
     NEXT K
```

```
240    F(N, D1) = 0
       X(N, D1) = 0
       FOR I1 = 1 TO N1
         I = N - I1
         S = 0
         FOR J = 1 TO N
           S = S + D(J)
         NEXT J
         S = S + 1
         IF S < M4 GOTO 360
         M5 = M4
         GOTO 370
360        M5 = S
370        FOR K = 1 TO M5
           IF D(I) - K + 1 <= 0 GOTO 440
           L1 = D(I) - K + 1
           F(I, K) = P1(I, L1) + E1(I, 0) + F(I + 1, 1)
           X(I, K) = D(I) - K + 1
           L1 = L1 + 1
           GOTO 480
440          L1 = 0
           F(I, K) = E1(I, K - D(I) - 1) + F(I + 1, K - D(I))
           X(I, K) = 0
           L1 = L1 + 1
480          IF M2 > S - K GOTO 520
           IF M2 > D(I) + M4 - K GOTO 550
           M6 = M2
           GOTO 570
520          IF S - K > D(I) + M4 - K GOTO 550
           M6 = S - K
           GOTO 570
550          M6 = D(I) + M4 - K
           GOTO 570
570          IF L1 - 1 = M6 GOTO 640
           FOR Z = L1 TO M6
             H0 = P1(I, Z) + E1(I, K + Z - D(I) - 1) + F(I + 1, K + Z - D(I))
             IF F(I, K) <= H0 GOTO 630
             F(I, K) = H0
             X(I, K) = Z
630          NEXT Z
640        NEXT K
       NEXT I1
       X1(1) = X(1, I0 + 1)
       N5 = I0 + 1
       FOR I = 2 TO N
         N6 = X1(I - 1) - D(I - 1) + N5
         X1(I) = X(I, N6)
         N5 = N6
       NEXT I
       CLS : PRINT : PRINT
       PRINT "INVENTORI DENGAN "; N; "PERIODE, MAKSIMUM INVENTORI "; M1; "DAN"
       PRINT "MAKSIMUM PRODUKSI "; M2
       PRINT : PRINT
```

```
FOR I = 1 TO N
  PRINT "PERMINTAAN PERIODE KE "; I; "ADALAH ="; D(I)
NEXT I
PRINT : PRINT "TOTAL BIAYA YANG DIPERLUKAN ="; F(1, I0 + 1)
PRINT
FOR I = 1 TO N
  PRINT "PRODUKSI PERIODE KE "; I; "ADALAH ="; X1(I)
NEXT I
END
```

Latihan / Pekerjaan Rumah

Perhatikan permasalahan berikut. Misalkan
n = banyaknya periode = 4
I0 = inventori di awal periode ke-1 = 0
IN = inventori di akhir periode ke-n = 0

$$BP_i(j) = \begin{cases} 0 & untuk\, j = 0 \; dan \; i = 1,2,...,n \\ 8 + 3j & untuk \;\; j = 1,2,... \; dan \; i = 1,2,...,n \end{cases}$$ = Biaya untuk memproduksi j satuan barang dalam periode ke i.

$$BI_i(j) = 0{,}5\,j \; untuk \;\; j = 1,2,... \quad dan \quad i = 1,2,...,n$$ = Biaya inventori sebanyak j satuan barang di akhir periode ke i.

Sedangkan tabel permintaan barang untuk 6 periode ke depan adalah sebagai berikut

Periode (i)	1	2	3	4
Permintaan (d_i)	4	6	2	3

Tentukan jumlah barang yang harus diproduksi di awal setiap periode yang meminimumkan biaya total (*policy cost*) untuk 4 periode.

Penggantian Alat

Setiap perusahaan baik besar maupun kecil tentunya pernah mengalami masalah kapan harus mengganti peralatan, terutama untuk keperluan produksi atau yang dapat menghasilkan uang. Ambil contoh saja, dalam tulisan ini, akan disajikan permasalahan tentang truk milik perusahaan yang dapat diberdayakan untuk mencari uang.

Jika estimasi yang dapat dipercaya akan pemasukan atau pendapatan (*revenue*), biaya perawatan (*upkeep*), dan biaya penggantian baru (*replacement*) dapat diperoleh, teknik pemrograman dinamis dapat digunakan untuk menentukan secara realistis akan kebijakan penggantian yang optimal.

Permasalahan

Misalkan perusahaan memiliki truk yang sudah berumur 2 tahun di awal tahun 2006 (tahun 1), dan estimasi berikut dapat dilakukan (semua diukur dengan satuan biaya yang sama kecuali umur dalam tahun):

Truk tahun 2004 (tahun –1)

Umur	2	3	4	5	6
Pemasukan	10	8	8	6	4
Perawatan	3	3	4	4	5
Penggantian	25	26	27	28	29

Truk tahun 2006 (tahun 1)

Umur	0	1	2	3	4
Pemasukan	14	16	16	14	12
Perawatan	1	1	2	2	3
Penggantian	20	22	24	25	26

Truk tahun 2007 (tahun 2)

Umur	0	1	2	3
Pemasukan	16	14	14	12
Perawatan	1	1	2	2

Penggantian	20	22	24	25

Truk tahun 2008 (tahun 3)

Umur	0	1	2
Pemasukan	18	16	16
Perawatan	1	1	2
Penggantian	20	22	24

Truk tahun 2009 (tahun 4)

Umur	0	1
Pemasukan	18	16
Perawatan	1	1
Penggantian	21	22

Truk tahun 2010 (tahun 5)

Umur	0
Pemasukan	20
Perawatan	1
Penggantian	21

Jika tahun 2006 (tahun 1) diawali dengan penggunaan truk yang sudah berusia 2 tahun, keputusan apa yang harus diambil di awal periode tersebut dan diawal setiap periode hingga 4 periode (tahun) mendatang sehingga memaksimumkan pendapatan ?

Pemecahan Masalah

Untuk dapat menggunakan persamaan fungsional dari pemrograman dinamis, kita gunakan notasi berikut:

$r_i(t)$ = adalah pemasukan atau pendapatan dalam periode ke i dari sebuah truk yang dibuat dalam tahun (i-t) dan sudah berumur t tahun diawal periode ke-i.

$u_i(t)$ = biaya perawatan dalam periode ke-i untuk sebuah truk yang dibuat tahun (i-t) dan seudah berumur t tahun diawal periode ke-i.

$c_i(t)$ = biaya penggantian sebuah truk yang dibuat tahun (i-t) dan sudah berumur t tahun diawal periode ke-i.

a = faktor diskon. Untuk mempermudah, misal a = 1.
IT = umur truk yang digunakan diawal permasalahan. Dalam hal ini IT = 2.
$f_i(t)$ = keuntungan optimal untuk periode i, i+1, …, 5 dimana periode ke-i diawali dengan penggunaan truk yang sudah berumur t tahun.
$x_i(t)$ = keputusan yang harus dibuat di awal periode ke-i yang menghasilkan $f_i(t)$. Hanya ada dua keputusan: **Tetap** menggunakan truk yang lama, atau **Beli** truk yang baru.

Secara umum untuk i = 1,2,…,5 dan t = 1,2,…,(i-1), (i+IT-1) maka

$$f_i(t) = \max\begin{bmatrix} Beli; & r_i(0) - u_i(0) - c_i(t) + a\ f_{i+1}(1) \\ Tetap; & r_i(t) - u_i(t) + a\ f_{i+1}(t+1) \end{bmatrix}$$

Untuk dapat menggunakan persamaan ini, kita asumsikan bahwa $f_6(j) = 0$ untuk semua j. Juga, untuk i=1, t = IT saja.

Seperti teknik pemrograman dinamis yang lain, kita asumsikan bahwa periode atau tahun ke 5 sudah dicapai dengan truk dengan berbagai umur, yaitu 1, 2, 3, 4 atau 6 tahun, dan gunakan persamaan fungsional diatas. Dengan demikian untuk i=5

t=1 $\quad f_5(1) = \max\begin{bmatrix} Beli; & r_5(0) - u_5(0) - c_5(1) + f_6(1) \\ Tetap; & r_5(1) - u_5(1) + f_6(2) \end{bmatrix}$

$\quad = \max\begin{bmatrix} Beli; & 20 - 1 - 22 + 0 \\ Tetap; & 16 - 1 + 0 \end{bmatrix} = 15$

$x_5(1)$ = Tetap

t=2 $\quad f_5(2) = \max\begin{bmatrix} Beli; & r_5(0) - u_5(0) - c_5(2) + f_6(1) \\ Tetap; & r_5(2) - u_5(2) + f_6(3) \end{bmatrix}$

$\quad = \max\begin{bmatrix} Beli; & 20 - 1 - 24 + 0 \\ Tetap; & 16 - 2 + 0 \end{bmatrix} = 14$

$x_5(2)$ = Tetap

t=3 $\quad f_5(3) = \max\begin{bmatrix} Beli; & r_5(0) - u_5(0) - c_5(3) + f_6(1) \\ Tetap; & r_5(3) - u_5(3) + f_6(4) \end{bmatrix}$

$= \max\begin{bmatrix} Beli; & 20 - 1 - 25 + 0 \\ Tetap; & 12 - 2 + 0 \end{bmatrix} = 10$

$x_5(3)$ = Tetap

t=4 $\quad f_5(4) = \max\begin{bmatrix} Beli; & r_5(0) - u_5(0) - c_5(4) + f_6(1) \\ Tetap; & r_5(4) - u_5(4) + f_6(5) \end{bmatrix}$

$= \max\begin{bmatrix} Beli; & 20 - 1 - 26 + 0 \\ Tetap; & 12 - 3 + 0 \end{bmatrix} = 9$

$x_5(4)$ = Tetap

t=6 $\quad f_5(6) = \max\begin{bmatrix} Beli; & r_5(0) - u_5(0) - c_5(6) + f_6(1) \\ Tetap; & r_5(6) - u_5(6) + f_6(7) \end{bmatrix}$

$= \max\begin{bmatrix} Beli; & 20 - 1 - 29 + 0 \\ Tetap; & 12 - 3 + 0 \end{bmatrix} = 9$

$x_5(6)$ = Tetap

Kalau kita berada diawal tahun ke-4 dengan truk yang sudah berumur 1, 2, 3, atau 5 tahun, maka

$$f_4(t) = \max\begin{bmatrix} Beli; & r_4(0) - u_4(0) - c_4(t) + 1\ f_5(1) \\ Tetap; & r_4(t) - u_4(t) + 1\ f_5(t+1) \end{bmatrix}$$

untuk t=1, 2, 3, 5

t=1 $\quad f_4(1) = \max\begin{bmatrix} Beli; & 18 - 1 - 22 + 15 \\ Tetap; & 16 - 1 + 14 \end{bmatrix} = 29$

$x_4(1)$ = Tetap

t=2 f$_4$(2) = $\max\begin{bmatrix} Beli; & 18-1-24+15 \\ Tetap; & 14-2+10 \end{bmatrix} = 22$

x$_4$(2) = Tetap

t=3 f$_4$(3) = $\max\begin{bmatrix} Beli; & 18-1-25+15 \\ Tetap; & 14-2+9 \end{bmatrix} = 21$

x$_4$(3) = Tetap

t=5 f$_4$(5) = $\max\begin{bmatrix} Beli; & 18-1-28+15 \\ Tetap; & 6-4-1 \end{bmatrix} = 4$

x$_4$(5) = Beli

Kalau kita berada diawal tahun ke-3 dengan truk yang sudah berumur 1, 2 atau 4 tahun, maka

$$f_3(t) = \max\begin{bmatrix} Beli; & r_3(0)-u_3(0)-c_3(t)+1\ f_4(1) \\ Tetap; & r_3(t)-u_3(t)+1\ f_4(t+1) \end{bmatrix}$$

untuk t=1, 2, 4

t=1 f$_3$(1) = $\max\begin{bmatrix} Beli; & 18-1-22+29 \\ Tetap; & 14-1+22 \end{bmatrix} = 35$

x$_3$(1) = Tetap

t=2 f$_3$(2) = $\max\begin{bmatrix} Beli; & 18-1-24+29 \\ Tetap; & 16-2+21 \end{bmatrix} = 35$

x$_3$(2) = Tetap

t=4 f$_3$(4) = $\max\begin{bmatrix} Beli; & 18-1-27+29 \\ Tetap; & 8-4+4 \end{bmatrix} = 19$

x$_3$(4) = Beli

Untuk periode atau tahun ke-2, maka untuk t = 1 dan 3 kita peroleh t=1 $\quad f_2(1) = \max\begin{bmatrix} Beli; & 16-1-22+35 \\ Tetap; & 16-1+35 \end{bmatrix} = 50$

$x_2(1)$ = Tetap

t=3 $\quad f_2(3) = \max\begin{bmatrix} Beli; & 16-1-26+35 \\ Tetap; & 8-3+19 \end{bmatrix} = 24$

$x_2(3)$ = Tetap atau Beli

Akhirnya, kita sampai ke tahun pertama dimana kita hanya miliki truk yang berumur 2 tahun. Apakah kita tetap menggunakan truk tersebut atau dengan membeli yang baru sehingga secara keseluruhan keuntungan selama empat tahun berikutnya akan maksimum? Untuk melihat itu, perlu diperhatikan

t=2 $\quad f_1(2) = \max\begin{bmatrix} Beli; & 14-1-25+50 \\ Tetap; & 10-3+24 \end{bmatrix} = 38$

$x_1(2)$ = Beli

Sehingga dapat disimpulkan , bahwa untuk mendapatkan biaya kebijakan maksimum kita dapat sarikan dalam tabel berikut

Tahun / Periode	Umur Truk	Keputusan
1	2	$x_1(2)$: Beli
2	1	$x_2(1)$: Tetap
3	2	$x_3(2)$: Tetap
4	3	$x_4(3)$: Tetap
5	4	$x_5(4)$: Tetap

Program Komputer

```
REM ************************************************************************
REM *                    PROGRAM PENGGANTIAN ALAT                *
REM *                                                            *
REM *                 Ir. SIGIT NUGROHO, M.Sc., Ph.D.            *
REM ************************************************************************
CLS
DIM F(11, 21), R(10, 21), U(10, 21), C(10, 21)
DIM POLICY(10, 21), XSTAR(10)
INPUT "JUMLAH PERIODE ="; N
INPUT "UMUR INCUMBENT ALAT ="; IT
INPUT "FAKTOR DISKON ="; A
NP1 = N + 1
NM1 = N - 1
ITPN1 = IT + N + 1
FOR I = 1 TO N
      FOR T = 1 TO I
      REM       PRINT "REVENUE PERIODE "; I; " PADA USIA ALAT "; T; : INPUT
R(I, T)
      READ R(I, T)
      NEXT T
      REM PRINT "REVENUE PERIODE "; I; " PADA USIA ALAT "; I + IT; : INPUT R(I,
I + IT)
      READ R(I, I + IT)
NEXT I
DATA 14, 10
DATA 16, 16, 8
DATA 18, 14, 16, 8
DATA 18, 16, 14, 14, 6
DATA 20, 16, 16,12, 12, 4

FOR I = 1 TO N
      FOR T = 1 TO I
            X(I, T) = R(I, T)
      NEXT T
      X(I, I + IT) = R(I, I + IT)
NEXT I
GOSUB 1000
FOR I = 1 TO N
      FOR T = 1 TO I
            REM PRINT "UPKEEP PERIODE "; I; " PADA USIA ALAT "; T; : INPUT U(I,
T)
            READ U(I, T)
      NEXT T
```

```
            REM PRINT "UPKEEP PERIODE "; I; " PADA USIA ALAT "; I + IT; : INPUT U(I, I
+ IT)
            READ U(I, I + IT)
      NEXT I
      DATA 1, 3
      DATA 1, 1, 3
      DATA 1, 1, 2, 4
      DATA 1, 1, 2, 2, 4
      DATA 1, 1, 2, 2, 3, 5


      FOR I = 1 TO N
            FOR T = 1 TO I
                  X(I, T) = U(I, T)
            NEXT T
            X(I, I + IT) = U(I, I + IT)
      NEXT I
      GOSUB 1000
      FOR I = 1 TO N
            FOR T = 1 TO I
                  REM PRINT "COST ALAT BARU PERIODE "; I; " PADA USIA ALAT "; T; :
INPUT C(I, T)
                  READ C(I, T)
            NEXT T
            REM PRINT "COST ALAT BARU PERIODE "; I; " PADA USIA ALAT "; I + IT; :
INPUT C(I, I + IT)
            READ C(I, I + IT)
      NEXT I
      DATA 20, 25
      DATA 20, 22, 26
      DATA 20, 22, 24, 27
      DATA 21, 22, 24, 25, 28
      DATA 21, 22, 24, 25, 26, 29
      FOR I = 1 TO N
            FOR T = 1 TO T
                  X(I, T) = C(I, T)
            NEXT T
            X(I, I + IT) = C(I, I + IT)
      NEXT I
      GOSUB 1000
      FOR T = 2 TO 14
            F(NF1, T) = 0
      NEXT T
      FOR II = 1 TO NM1
            I = N - II + 1
            FOR T = 2 TO I
                  GOSUB 2000
```

```
            NEXT T
            T = I + IT
            GOSUB 2000
        NEXT II
        I = 1
        T = IT + 1
        GOSUB 2000
9       IF POLICY(I, T) = 2 GOTO 12
        XSTAR(I) = 1
        T = 2
11      IF I = N GOTO 14
        I = I + 1
        GOTO 9
12      XSTAR(I) = 2
        T = T + 1
        GOTO 11
14      CLS
        PRINT "UNTUK PROSES "; N; " PERIODE"
        PRINT "DIAWALI DENGAN MESIN BERUMUR "; IT; " TAHUN"
        PRINT "DIPEROLEH OPTIMAL RETURN ="; F(1, IT + 1)
        PRINT
        PRINT
        FOR I = 1 TO N
            IF XSTAR(I) = 1 GOTO 16
            PRINT "AWAL PERIODE "; I; " TETAP PAKAI, TIDAK BELI"
            GOTO 15
16          PRINT "AWAL PERIODE "; I; " BELI MESIN BARU"
15      NEXT I
        GOTO 5000

1000    DIM NUM(10), FORM(9)
        JT = IT - 1
        FOR I = 1 TO N
            FORM(5) = FX
            PRINT I
            CHECK = 0
            FOR K = 1 TO I
                IF (K MOD 7) = 1 GOTO 245
                INC = (K - 1) MOD 7
245             FORM(4) = NUM0
250             PRINT X(I, K)
                CHECK = CHECK + 1
                ITEST = I / 7 + 1
                IJTEST = (I + JT) / 7
                IF ITEST - 1 = IJTEST GOTO 247
                IF I < 7 * ITEST AND I + JT > 7 * IJTEST AND I = K GOTO 220
247             IF I + JT = 7 * IJTEST AND K = I GOTO 225
```

```
                 IF (CHECH MOD 7) <> 0 GOTO 210
                 FORM(5) = ZX
                 GOTO 210
220                KJT = JT - 7 * IJTEST + I
                 FORM(4) = NUM(KJT)
                 FORM(5) = ZX
REM                  PRINT X(I, I + IT)
                 GOTO 20
225  REM              PRINT X(I, I + IT)
                 GOTO 20
210          NEXT K
           INK = (I + JT) MOD 7
           FORM(4) = NUM(INK)
REM            PRINT X(I, I + IT)
20     NEXT I
      RETURN

2000    PURCH = R(I, 1) - U(I, 1) - C(I, T) + A * F(I + 1, 2)
      KEEP = R(I, T) - U(I, T) + A * F(I + 1, T + 1)
      IF PURCH > KEEP GOTO 3
      F(I, T) = KEEP
      POLICY(I, T) = 2
      GOTO 4
3      F(I, T) = PURCH
      POLICY(I, T) = 1
4      RETURN

5000    END
```

Masalah Penugasan

Misalkan perusahaan memiliki m pekerjaan yang harus diselesaikan dan memiliki m pekerja yang dapat mengerjakan setiap pekerjaan tersebut, sedang yang membedakan adalah waktu penyelesaian tiap pekerjaan dari tiap pekerja. Permasalahan yang harus diselesaikan adalah mengalokasikan tiap pekerjaan ini ke tiap pekerja, dimana setiap pekerja pasti dapat satu pekerjaan sedemikian rupa sehingga total waktunya minimum ?

Dengan memisalkan x_{ij} bernilai 0 apabila perkerjaan i tidak dikerjakan oleh pekerja j dan bernilai 1 apabila pekerjaan i dikerjakan oleh pekerja j, serta c_{ij} merupakan nilai keefektifan yang berhubungan dengan penugasan pekerjaan i untuk pekerja j. (i=1,2,…,m dan j=1,2,…,m).

Dengan notasi matematik, permasalahan penugasan (*Assignment Problem*) ini dapat dituliskan sebagai berikut :

Minimumkan (atau maksimumkan) $z = \sum_{i=1}^{m}\sum_{j=1}^{m} c_{ij} x_{ij}$

dengan kendala $\sum_{j=1}^{m} x_{ij} = 1$ untuk i = 1,2,…,m

$\sum_{i=1}^{m} x_{ij} = 1$ untuk j = 1,2,…,m

serta x_{ij} = 0 atau 1 untuk semua i dan j.

Sudah barang tentu ini dapat diselesaikan dengan metode yang biasa digunakan untuk pemrograman linier, seperti Simplex, misalnya.

Hungarian Method

Metode ini sebagai hasil dari apa yang dikerjakan oleh matematikawan Hungaria D Konig yang telah membuktikan dalil penting bagi pengembangan metode ini. Sebagai dasar, metode ini secara berturut-turut memodifikasi baris dan kolom dari matriks keefektifan sampai sedikitnya ada satu komponen nol dalam tiap baris dan kolom sehingga penugasan secara lengkap yang bersesuaian dengan angka-angka nol ini dapat dilakukan. Penugasan lengkap ini akan menjadi penugasan yang optimal, bila dikembalikan ke permasalahan semula. Metode ini akan selalu konvergen ke suatu penugasan optimal dalam sejumlah langkah (terhingga). Basis dari metode ini adalah fakta bahwa sebuah konstanta dapat ditambahkan atau dikurangkan dari sembarang baris atau kolom tanpa harus merubah gugus penugasan optimal. Sebagai contoh, jika baris ke-i dikurangi dengan 4 semua, dan kolom ke-j ditambah 5 semua, maka fungsi tujuannya menjadi

$$\text{Minimumkan } z = \sum_{i=1}^{m}\sum_{j=1}^{m} c_{ij}x_{ij} - 4\sum_{j=1}^{m} x_{ij} + 5\sum_{i=1}^{m} x_{ij}$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{m} c_{ij}x_{ij} - 4 + 5$$

Dapat dibuat hal yang lebih umum lagi apabila jika baris ke-i dikurangi dengan a_i semua, dan kolom ke-j ditambah b_j semua, maka fungsi tujuannya menjadi

$$\text{Minimumkan } z = \sum_{i=1}^{m}\sum_{j=1}^{m} c_{ij}x_{ij} - \sum_{i=1}^{m} a_i \sum_{j=1}^{m} x_{ij} + \sum_{j=1}^{m} b_j \sum_{i=1}^{m} x_{ij}$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{m} c_{ij}x_{ij} - \sum_{i=1}^{m} a_i + \sum_{j=1}^{m} b_j$$

Kita ambil teladan berikut:

Misalkan terdpat empat Profesor (Ahmad, Badu, Chepot, dan Deon) yang dapat mengajar tiap mata kuliah berikut : Manajemen Pemasaran, Manajemen Organisasi, Manajemen Produksi, dan Manajemen Sains. Namun demikian, karena latar belakang dan pengalaman yang tak sama, rata-rata waktu perminggu untuk menyiapkan perkuliahan tersebut tentu tak sama. Karena jurusan ini sangat berorientasi penelitian, maka ketua jurusan ingin menentukan profesor siapa harus mengajar apa sehingga waktu yang diperlukan untuk mempersiapkan perkuliahan seminimal mungkin, sehingga banyak waktu yang dapat digunakan untuk penelitian. Tabel waktu persiapan dari permasalahan tersebut dapat dilihat sebagai berikut

Profesor	Mata Kuliah			
	Manajemen Pemasaran	Manajemen Organisasi	Manajemen Produksi	Manajemen Sains
Ahmad	2	10	9	7
Badu	15	4	14	8
Chepot	13	14	16	11
Deon	4	15	13	9

Langkah pertama dari metode Hungaria adalah memodifikasi matrriks keefektifan diatas (waktu) untuk mendapatkan sedikitnya satu nol di setiap baris dan setiap kolom dengan harapan nantinya dapat membuat penugasan lengkap yang berhubungan dengan nol dari matriks yang tereduksi (nilainya). Untuk itu tiap nilai dari tiap baris dikurangi dengan nilai terendah dari baris yang bersangkutan. Hasilnya menjadi

Profesor	Mata Kuliah			
	Manajemen Pemasaran	Manajemen Organisasi	Manajemen Produksi	Manajemen Sains
Ahmad	0	8	7	5
Badu	11	0	10	4
Chepot	2	3	5	0
Deon	0	11	9	5

Hal ini menunjukkan bahwa pada kolom Manajemen Produksi, belum diperoleh satupun nilai nol, oleh karenanya proses serupa juga perlu dilakukan untuk pengolahan kolom.

Profesor	Mata Kuliah			
	Manajemen Pemasaran	Manajemen Organisasi	Manajemen Produksi	Manajemen Sains
Ahmad	0	8	2	5
Badu	11	0	5	4
Chepot	2	3	0	0
Deon	0	11	4	5

Langkah kedua dari metode ini adalah dengan cara membuat penugasan lengkap yang hanya memiliki elemen 0. Dimulai dari baris pertama, nilai 0 diberi tanda Δ atau X. Beri tanda 0 dengan Δ apabila penugasan diberikan dari profesor dari baris tersebut untuk mata kuliah pada kolom yang bersangkutan. Beri tanda X angka nol yang lain pada kolom yang sama. Ulangi hal ini hingga tiap baris tidak memiliki 0 yang ditandai atau sedikitnya dua angka 0. Hasil proses berikutnya adalah

Profesor	Mata Kuliah			
	Manajemen Pemasaran	Manajemen Organisasi	Manajemen Produksi	Manajemen Sains
Ahmad	**0** Δ	8	2	5
Badu	11	**0** Δ	5	4
Chepot	2	3	**0** Δ	**0** X
Deon	**0** X	11	4	5

Karena penugasan secara lengkap tak dapat dibuat dengan hanya menggunakan angka nol saja, maka tarik garis yang digunakan untuk menutup angka nol ini baik horizontal ataupun vertikal.

a. Cek semua baris dimana penugasan belum dilakukan (*baris ke-empat*)
b. Cek kolom-kolom yang belum ditandai yang memiliki 0 dalam baris yang sudah ditandai (*kolom pertama*)
c. Cek baris-baris yang belum ditandai yang memiliki penugasan dalam kolom yang sudah ditandai (*baris pertama*)
d. Ulangi dua tahap sebelum ini, b dan c, hingga tak ada pengecekan lagi
e. Gambar garis melalui semua baris yang belum ditandai dan kolom yang ditandai (*baris kedua, baris ketiga, dan kolom pertama*)

Profesor	Mata Kuliah			
	Manajemen Pemasaran	Manajemen Organisasi	Manajemen Produksi	Manajemen Sains
Ahmad	**0** Δ	8	2	5
Badu	11	**0** Δ	5	4
Chepot	2	3	**0** Δ	**0** X
Deon	**0** X	11	4	5

Jika matriks keefektifan memiliki m baris dan m kolom, maka minimum baris diperlukan untuk menutupi minimal sekali setiap 0 dalam matriks tereduksi haruslah m, jika penugasan lengkap dapat dibuat dengan total keefektifan 0. Hingga tahap ini, baru tiga garis (ditunjukkan dengan dua baris dan satu kolom yang diarsir) yang diperlukan untuk menutup semua angka 0 dari metriks tereduksi, sehingga jawaban layak dan lengkap belum dapat dibuat.

f. Periksa elemen-elemen dalam matriks tereduksi yang tidak memiliki sedikitnya satu satu garis yang melaluinya. Misalkan *k* merupakan elemen terkecil. Kurangkan *k* dari tiap elemen di tiap baris yang mengandung elemen-elemen tak tertutup. Saat ini elemen yang dimaksud adalah 2. Karenanya kurangkan dengan dua untuk semua elemen tak tertutup.

-2 Δ	6	0	3
11	**0** Δ	5	4
2	3	**0** Δ	**0** X
-2 X	9	2	3

g. Pengurangan tersebut mengakibatkan beberapa elemen memiliki nilai negatif. (kolom 1), karenanya, tambahkan setiap elemen dalam kolom tersebut dengan 2.

0 Δ	6	0	3
13	**0** Δ	5	4
4	3	**0** Δ	**0** X
0 X	9	2	3

Dapat diperlihatkan bahwa, jika penugasan lengkap mencakup elemen 0 hanya dapat dibuat dengan matriks tereduksi seperti tahap diatas, total keefektifan dengan menggunakan penugasan yang sama dengan matriks aslinya akan minimum.

0 X	6	**0 ∆**	3
13	**0 ∆**	5	4
4	3	**0 X**	**0 ∆**
0 ∆	9	2	3

Untuk menentukan penugasan secara lengkap, prosedur umum untuk membuat penugasan diulang dengan menggunakan matriks keefektifan baru dalam tahap *g*. Lebih spesifik lagi

- Buat penugasan dalam baris kedua
- Buat penugasan dalam baris keempat dan beri tanda X angka 0 pada baris pertama kolom pertama
- Buat penugasan dalam baris pertama dan beri tanda X angka 0 pada baris ketiga kolom ketiga
- Buat penugasan dalam baris ketiga

Penugasan optimal lengkap telah dibuat, dengan hasil seperti berikut

Ahmad mengajar Manajemen Produksi	9
Badu mengajar Manajemen Organisasi	4
Chepot mengajar Manajemen Sains	11
Deon mengajar Manajemen Pemasaran	4

Dengan waktu persiapan minimum sebesar 9 + 4 + 11 + 4 = 28 satuan waktu.

Program Komputer

```
420 REM ****************************************************************
421 REM  *  PROGRAM MODEL PENUGASAN UNTUK MENGALOKASIKAN N PROYEK/TUGAS *
422 REM  *     UNTUK N KONTRAKTOR/PEKERJA GUNA OPTIMASI BIAYA/WAKTU    *
423 REM  *            -= Ir. SIGIT NUGROHO, M.Sc., Ph.D. =-           *
424 REM ****************************************************************
425 CLS : INPUT "BANYAKNYA TUGAS ="; N
426 PRINT : PRINT "KODE 0 = MINIMISASI"
427 PRINT "KODE 1 = MAKSIMISASI"
430 PRINT : INPUT "PILIH KODE OPTIMISASI (0/1) ="; K
431 IF (K = 0 OR K = 1) GOTO 433
432 PRINT "Oooops ... Tolong Ulang Masukkan Kode": GOTO 430
433 IF K = 0 THEN A$ = "MINIMUM"
434 IF K = 1 THEN A$ = "MAKSIMUM"
435 DIM M(N, N), A(N, N), B(N, N), R(N), C(N), Y(N), Z(N)
436 DIM R1(N), C1(N)
```

```
437 PRINT
440 FOR I = 1 TO N
445 FOR J = 1 TO N
450 PRINT "PERUSAHAAN "; I; " TUGAS "; J; : INPUT M(I, J)
455 A(I, J) = M(I, J)
460 NEXT J
461 NEXT I
462 PRINT : PRINT "PERUSAHAAN", "TUGAS", "BIAYA"
464 FOR I = 1 TO N
465 FOR J = 1 TO N
466 PRINT I, J, M(I, J)
467 NEXT J: NEXT I
475 GOSUB 505
480 PRINT : PRINT "JAWABAN "; A$
483 PRINT "DARI PERMASALAHAN ADALAH ="; T5
484 PRINT : PRINT : PRINT
485 FOR I = 1 TO N
490 PRINT "PERUSAHAAN "; I, "MENDAPAT TUGAS PROYEK "; R1(I)
495 NEXT I
500 END
505 IF K = 1 GOTO 540
510 FOR I = 1 TO N
515 FOR J = 1 TO N
520 B(I, J) = A(I, J)
525 NEXT J
530 NEXT I
535 GOTO 565
540 FOR I = 1 TO N
545 FOR J = 1 TO N
550 B(I, J) = -A(I, J)
555 NEXT J
560 NEXT I
565 FOR I = 1 TO N
570 H = B(I, 1)
575 FOR J = 1 TO N
580 IF B(I, J) <= H THEN H = B(I, J)
590 NEXT J
595 FOR L = 1 TO N
600 B(I, L) = B(I, L) - H
605 NEXT L
610 NEXT I
615 FOR J = 1 TO N
620 P = B(1, J)
625 FOR I = 1 TO N
630 IF B(I, J) <= P THEN P = B(I, J)
640 NEXT I
645 FOR S = 1 TO N
650 B(S, J) = B(S, J) - P
655 NEXT S
660 NEXT J
665 FOR I = 1 TO N
670 R1(I) = 0
```

```
675 C1(I) = 0
680 NEXT I
685 N0 = 0
690 L0 = 0
695 Z0 = 0
700 FOR I = 1 TO N
705 Z1 = 0
710 IF R1(I) <> 0 GOTO 780
715 FOR J = 1 TO N
720 IF C1(J) <> 0 GOTO 750
725 IF B(I, J) <> 0 GOTO 750
730 Z1 = Z1 + 1
735 Z0 = Z0 + 1
740 N1 = J
745 N2 = I
750 NEXT J
755 IF Z1 <> 1 GOTO 780
760 R1(I) = N1
765 C1(N1) = 1
770 N0 = N0 + 1
775 L0 = L0 + 1
780 NEXT I
785 FOR I = 1 TO N
790 N5 = 0
795 IF C1(I) <> 0 GOTO 860
800 FOR J = 1 TO N
805 IF R1(J) <> 0 GOTO 830
810 IF B(J, I) <> 0 GOTO 830
815 N5 = N5 + 1
820 Z0 = Z0 + 1
825 N6 = J
830 NEXT J
835 IF N5 <> 1 GOTO 860
840 C1(I) = 1
845 R1(N6) = I
850 N0 = N0 + 1
855 L0 = L0 + 1
860 NEXT I
865 IF N0 = N GOTO 1145
870 IF L0 <> 0 GOTO 690
875 IF Z0 = 0 GOTO 900
880 R1(N2) = N1
885 N0 = N0 + 1
890 C1(N1) = 1
895 GOTO 690
900 FOR I = 1 TO N
905 Y(I) = 0: Z(I) = 0
915 IF R1(I) <> 0 GOTO 925
920 Y(I) = I
925 NEXT I
930 N8 = 0
935 FOR I = 1 TO N
```

```
940 IF Y(I) = 0 GOTO 975
945 FOR J = 1 TO N
950 IF Z(J) <> 0 GOTO 970
955 IF B(I, J) <> 0 GOTO 970
960 Z(J) = J
965 N8 = N8 + 1
970 NEXT J
975 NEXT I
980 IF N8 = 0 GOTO 1030
985 FOR I = 1 TO N
990 IF Z(I) <= 0 GOTO 1020
995 FOR J = 1 TO N
1000 IF Z(I) <> R1(J) GOTO 1015
1005 Y(J) = J
1010 N8 = N8 + 1
1015 NEXT J
1020 NEXT I
1025 IF N8 <> 0 GOTO 930
1030 Z8 = 9999999
1035 FOR I = 1 TO N
1040 IF Y(I) = 0 GOTO 1070
1045 FOR J = 1 TO N
1050 IF Z(J) <> 0 GOTO 1065
1055 IF B(I, J) >= Z8 GOTO 1065
1060 Z8 = B(I, J)
1065 NEXT J
1070 NEXT I
1075 FOR I = 1 TO N
1080 FOR J = 1 TO N
1085 IF Y(I) < 0 GOTO 1130
1090 IF Y(I) = 0 GOTO 1115
1100 IF Z(J) <> 0 GOTO 1125
1105 B(I, J) = B(I, J) - Z8
1110 GOTO 1125
1115 IF Z(J) <= 0 GOTO 1125
1120 B(I, J) = B(I, J) + Z8
1125 NEXT J
1130 NEXT I
1140 GOTO 665
1145 T5 = 0
1150 FOR I = 1 TO N
1155 K5 = R1(I)
1160 T5 = T5 + A(I, K5)
1165 NEXT I
1170 RETURN
1175 END
```

Latihan

Seorang manajer harus memesan 5 truk dari delapan truk yang dapat dipesan di delapan lokasi yang berbeda untuk mengangkut barang siap kirim. Biaya transportasi dari lokasi truk ke tempat pemuatan barang, disajikan seperti berikut

Truk	Tempat Pemuatan				
	A	B	C	D	E
1	300	290	280	290	210
2	250	310	290	300	200
3	180	190	300	190	180
4	320	180	190	240	170
5	270	210	190	250	160
6	190	200	220	190	140
7	220	300	230	180	160
8	260	190	260	210	180

Tentukan penugasan dari tiap truk ke tempat pemuatan yang mana, agar total biaya yang dikeluarkan seminimum mungkin.

RUTE PERJALANAN BISNIS

Seorang pelaku bisnis yang berada di kota tertentu, sebut saja “A”, ingin berkunjung ke setiap N-1 kota lainnya sekali dan hanya sekali, dan kemudian kembali ke kota dimana ia tinggal atau “A”. Bagaimanakah urutan perjalanan yang harus ia lakukan untuk meminimumkan total jarak yang harus ditempuhnya. Biasanya jarak dari satu kota ke kota lain ini akan berpengaruh terhadap biaya perjalanan dari satu kota ke kota lainnya.

Misalkan D(I,J) merupakan jarak dari kota I ke kota J untuk I=1,2,…,N dan J=1,2,…,N, dimana N adalah banyaknya kota yang ada dalam permasalahan. Dalam praktek dapat saja $D(I,J) \neq D(J,I)$. Kita misalkan $D(I,I)=\infty$ untuk I=1,2,…,N. Suatu *tour* atau perjalanan merupakan rute lengkap jika melewati N kota dimana tak ada kota yang dikunjungi lebih dari satu kali. Jika pelaku bisnis tersebut mengunjungi kota tertentu dankembali ke kota tersebut di kemudian hari, kota tersebut membentuk suatu *subtour*. Sudah tentu ini tak dapat terjadi jika rutenya layak, yaitu bila tiap kota hanya dikunjungi sekali dan hanya sekali.

Berikut ini adalah *Eastman Algorithm* yang digunakan untuk memecahkan masalah tersebut diatas.

Langkah 1 Misalkan CLUB (Current Least Upper Bound) atau Batas Atas Terkecil Saat ini sebagai jawab optimal permasalahan. Berikan nilai misalnya = 10^{10}.

Langkah 2 Selesaikan masalah ini sebagai masalah penugasan, dimana D(I,J) merupakan unsur matrik keefektifan. Jawaban ini merupakan batas bawah jawab optimal permasalahan. Jika sedikitnya satu *subtour* terjadi, lanjutkan langkah 3, dan bila tidak maka jawab dalam masalah penugasan juga merupakan jawab optimal dari permasalahan ini.

Langkah 3 Pilih sebuah *subtour* dan *k* adalah banyaknya busur atau lengkung atau penghubung dalam subtour terpilih. Eastman memilih subtour dengan jumlah busur terkecil. Semua subtour pada nodus ini dapat diabaikan.

Langkah 4 Cabangkan menjadi *k* sub masalah. Jika subtournya adalah $I_1 \rightarrow I_2 \rightarrow \ldots \rightarrow I_k \rightarrow I_1$, maka untuk sub masalah 1 misalkan $D(I_1,I_2)=\infty$, untuk sub masalah 2 misalkan $D(I_2,I_3)=\infty$, …,untuk sub masalah k misalkan $D(I_k,I_1)=\infty$, dst.

Langkah 5 Selesaikan seluruh k masalah penugasan yang baru. Setiap jawaban merupakan batas bawah sub masalah yang sesuai.

Langkah 6 Jika terdapat satu atau lebih jawab yang layak dari Langkah 5 dan jika total jarak terkecil untuk jawab layak ini, sebut saja JTK, lebih kecil dari CLUB, maka nilai CLUB=JTK. Bila tidak nilai CLUB tidak berubah.

Langkah 7 Jika CLUB lebih kecil dari batas bawah semua sub masalah yang tak tereksplorasi, maka jawaban yang sesuai terhadap CLUB merupakan jawab optimal dari permasalah, karenanya SELESAI; Bila tidak pergi ke langkah 8. Yang dimaksud dengan masalah yang belum tereksplorasi adalah sub masalah yang belum dibagi lagi menjadi sub masalah yang lebih jauh.

Langkah 8 Dari gugus semua sub masalah tak layak yang belum tereksplorasi dengan batas kurang dari CLUB, pilih sub masalah dengan batas bawah terkecil untuk pencabangan lebih lanjut. Pergi ke langkah 3.

Teladan

Misalkan tabel jarak perjalanan bisnis untuk 5 kota adalah sebagai berikut

Dari Kota	Ke kota				
	1	2	3	4	5
1	∞	1	7	4	3
2	2	∞	6	3	4
3	1	6	∞	2	1
4	1	5	4	∞	6
5	7	5	4	5	∞

Langkah 1

Dengan menggunakan *Hungarian Method* kita selesaikan masalah ini sebagaimana menyelesaikan masalah penugasan.

$$\begin{bmatrix} \infty & 1 & 7 & 4 & 3 \\ 2 & \infty & 6 & 3 & 4 \\ 1 & 6 & \infty & 2 & 1 \\ 1 & 5 & 4 & \infty & 6 \\ 7 & 5 & 4 & 5 & \infty \end{bmatrix} \xrightarrow{baris} \begin{bmatrix} \infty & 0 & 6 & 3 & 2 \\ 0 & \infty & 4 & 1 & 2 \\ 0 & 5 & \infty & 1 & 0 \\ 0 & 4 & 3 & \infty & 5 \\ 3 & 1 & 0 & 1 & \infty \end{bmatrix}$$

$$\xrightarrow{kolom} \begin{bmatrix} \infty & [0] & 6 & 2 & 2 \\ 0 & \infty & 4 & [0] & 2 \\ 0 & 5 & \infty & 0 & [0] \\ [0] & 4 & 3 & \infty & 5 \\ 3 & 1 & [0] & 0 & \infty \end{bmatrix}$$

Dengan demikian penugasan optimalnya adalah 1→2, 2→4, 3→5, 4→1, dan 5→3 dengan total jarak 10. Dalam penugasan ini terdapat dua subtour, yaitu 1→2→4→1 dan 3→5→3.

Langkah 2

Karena jawab masalah penugasan dalam Langkah 1 tidak menghasilkan suatu tour, 10 merupakan batas bawah semua jawab layak masalah perjalanan bisnis ini.

Langkah 3

Subtour dengan jumlah lengkung 2 adalah 3→5→3 sedangkan 1→2→4→1 memiliki 3 lengkung. Dengan demikian *k* = 2.

Langkah 4

Cabangkan menjadi 2 sub masalah, satu dengan D(3,5)=∞ dan satunya dengan D(5,3)=∞.

Langkah 5

Misalkan D(3,5)=∞ dan selesaikan permasalahan penugasan berikut

$$\begin{bmatrix} \infty & 1 & 7 & 4 & 3 \\ 2 & \infty & 6 & 3 & 4 \\ 1 & 6 & \infty & 2 & \infty \\ 1 & 5 & 4 & \infty & 6 \\ 7 & 5 & 4 & 5 & \infty \end{bmatrix} \xrightarrow{baris} \begin{bmatrix} \infty & 0 & 6 & 3 & 2 \\ 0 & \infty & 4 & 1 & 2 \\ 0 & 5 & \infty & 1 & \infty \\ 0 & 4 & 3 & \infty & 5 \\ 3 & 1 & 0 & 1 & \infty \end{bmatrix}$$

$$\xrightarrow{kolom} \begin{bmatrix} \infty & [0] & 6 & 2 & 0 \\ 0 & \infty & 4 & 0 & [0] \\ 0 & 5 & \infty & [0] & \infty \\ [0] & 4 & 3 & \infty & 3 \\ 3 & 1 & [0] & 0 & \infty \end{bmatrix}$$

Penugasan optimalnya merupakan suatu tour dengan total jarak 12. Penugasan atau rute yang harus ditempuh adalah 1→2→5→3→4→1. Karena ini merupakan jawab yang layak dari permasalahan perjalanan bisnis, nilai tersebut juga merupakan batas atas terkecil dari jawab optimal yang diperoleh hingga disini. Tak ada pencabangan lebih lanjut dari nodus ini, karena jawabnya layak.

Untuk D(5,3)=∞ dan dengan cara yang sama kita selesaikan permasalahan penugasan

$$\begin{bmatrix} \infty & 1 & 7 & 4 & 3 \\ 2 & \infty & 6 & 3 & 4 \\ 1 & 6 & \infty & 2 & 1 \\ 1 & 5 & 4 & \infty & 6 \\ 7 & 5 & \infty & 5 & \infty \end{bmatrix} \xrightarrow{baris} \begin{bmatrix} \infty & 0 & 6 & 3 & 2 \\ 0 & \infty & 4 & 1 & 2 \\ 0 & 5 & \infty & 1 & \infty \\ 0 & 4 & 3 & \infty & 5 \\ 3 & 1 & 0 & 1 & \infty \end{bmatrix}$$

$$\xrightarrow{kolom} \begin{bmatrix} \infty & [0] & 3 & 3 & 2 \\ [0] & \infty & 1 & 1 & 2 \\ 0 & 5 & \infty & 1 & [0] \\ 0 & 4 & [0] & \infty & 5 \\ 2 & 0 & \infty & [0] & \infty \end{bmatrix}$$

Penugasan optimalnya memiliki 2 sub tour 1→2→1 sedangkan 3→5→4→3 dengan total jarak 13.

Langkah 6

Karena jawaban untuk sub masalah dengan D(3,5)= ∞ layak dengan nilai 12 lebih kecil dari batas bawah pada sub masalah lainnya, maka nilai tersebut (12) merupakan jawab optimal dari masalah perjalanan bisnis diatas. Tak perlu kita mengecek lebih jauh dari sub masalah yang muncul dari sub tour lainnya pada nodus 1 (1→2→4→1).

Pekerjaan Rumah

Dengan menggunakan cara atau lengkah seperti diatas, carilah rute yang harus ditempuh pelaku bisnis untuk berkeliling ke 10 kota, berangkat dari kota 1 dan berakhir di kota 1 sehingga total jaraknya minimum. Setiap kota hanya dikunjungi sekali dan hanya sekali. Berikut ini adalah tabel jarak dari dan ke 10 kota tersebut

Dari Kota	Ke Kota									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	∞	184	292	449	670	516	598	618	881	909
2	184	∞	195	310	540	357	514	434	697	964
3	292	195	∞	215	380	232	434	493	719	955
4	449	310	215	∞	288	200	566	787	790	1020
5	670	540	380	288	∞	211	436	814	632	974
6	516	357	232	200	211	∞	381	642	697	952
7	598	514	434	566	436	381	∞	295	224	541
8	618	434	493	787	814	642	295	∞	320	341
9	881	697	719	790	632	697	224	320	∞	318
10	909	964	955	1020	974	952	541	341	318	∞

MASALAH PEMUATAN BARANG

Masalah pemuatan barang sering muncul dalam perusahaan pengiriman barang. Dalam menyelesaikan masalah ini, kita akan pelajari bagaimana kita harus memuat barang atau barang-barang apa saja yang harus dimuat, sesuai dengan prioritas. Disini sebagai prioritas adalah **nilai barang** tersebut. Selain itu kita juga harus melihat berat barang itu sendiri, yang tentunya merupakan kendala karena kapasitas muat kendaraan akan barang-barang tersebut.

Bila berat barang I dinotasikan dengan W(I) dan nilai barang ke I tersebut adalah V(I), barang mana yang harus dimuat untuk memaksimumkan nilai muatan apabila maksimum berat yang diperbolehkan adalah W ? Diasumsikan bahwa tiap barang tak dapat dibagi. Secara matematis, masalah tersebut adalah menentukan nilai X(I), I=1,2,3,…,N yang memaksimumkan $Z = \sum_{I=1}^{N} X(I).V(I)$ dengan kendala $\sum_{I=1}^{N} X(I).W(I) \leq W$ serta X(I) = 0 apabila barang ke I tidak termuat dan X(I) = 1 apabila barang ke I termuat.

Notasi

Untuk menyelesaikan masalah diatas perlu kiranya kita berikan notas yang digunakan:

N	=	Jumlah barang
W(I)	=	Berat barang ke I (I=1,2,…,N)
V(I)	=	Nilai barang ke I (I=1,2,…,N)
B(K)	=	Nilai batas pada nodus K
SETI(K)	=	Gugus barang yang termasuk dalam nodus K
SETE(K)	=	Gugus barang yang tak termasuk pada nodus K

SETF(K)	=	Gugus barang yang bebas pada nodus K
Z0	=	Batas bawah terbesar sekarang pada gugus semua jawab yang layak
WL	=	Batas berat maksimum

Langkah 1 Jika $\min_{I}[W(I)] > WL$ maka tak ada jawab yang layak, artinya tak ada barang yang dapat dimuat. BERHENTI. Jika tidak pergi ke langkah 2

Langkah 2 Jika $\sum_{I=1}^{N} W(I) \leq WL$ maka semua barang dapat dimuat dan masalahnya jelas dan mudah. BERHENTI. Jika tidak pergi ke langkah 3

Langkah 3 Urutkan barang berdasarkan urutan menurun dari rasio V(I)/W(I) dan pergi ke langkah 4

Langkah 4 $B(1) = 10^{10}$

SETI(1) = ϕ ϕ adalah himpunan kosong

SETE(1) = ϕ

SETF(1) = {1,2,…,N}

Pergi ke langkah 5

Langkah 5 Misalkan K merupakan nodus terminal dengan batas terbesar. Suatu nodus merupakan terminal jika nodus tersebut tidak memiliki cabang yang muncul darinya. Percabangan berikutnya akan mengambil tempat pada nodus K. Untuk awalnya K bernilai 1. Pergi ke langkah 6

Langkah 6 Jika SETF(K)= ϕ, jawab optimal diperoleh dari indeks-indeks yang ada di dalam SETI(K). Nilai maksimum kargonya adalah

$$\max(Z) = \sum_{I \in SETI(K)} V(I)$$

. BERHENTI. Jika SETF(K)$\neq \phi$, lanjutkan ke langkah 7.

Langkah 7 Cabangkan barang dengan maks[V(I)/W(I)] untuk seluruh I yang menjadi anggota SETF(K).

	Misalkan ISTAR merupakan barang. Satu cabang memasukkan ISTAR dan cabang lain tidak memasukkan ISTAR. Lanjutkan langkah 8.
Langkah 8	Perhatikan nodus dimana ISTAR pertama kali tak termasuk. K=K+1 SETI(K)=SETI(K-1) SETE(K)=SETE(K-1) $\cup$ ISTAR COUNT=1 Lanjutkan ke langkah 9
Langkah 9	Hitung batas atas B(K) dengan pertama kali memuat semua barang dalam gugus SETI(K) dan kemudian lanjutkan secara berurut (nilai maksimum) memuat sebanyak mungkin tiap item yang menjadi anggota SETF(K) hingga total berat persis sama dengan WL. Nilai total dari barang yang dimuat dengan B(K). Pergi ke langkah 10.
Langkah 10	Jika COUNT=1, pergi ke langkah 11; bila tidak, pergi ke langkah 13.
Langkah 11	Perhatikan nodus dimana ISTAR termasuk. K=K+1 SETI(K) = SETI(K-2) $\cup$ ISTAR SETE(K) = SETE(K-2) COUNT = 0 Lanjutkan langkah 12
Langkah 12	Uji kelayakan jawaban yang sesuai dengan nodus K dengan memeriksa apakah $\sum_{I \in SETI(K)} W(I) \le WL$. Jika uji gagal, maka B(K)= -10^{10} dan lanjutkan ke langkah 13; jika tidak pergi ke langkah 9.
Langkah 13	SETF(K-1) = SETF(K-2)-ISTAR SETF(K) = SETF(K-2)-ISTAR

Lanjutkan ke Langkah 5

Teladan

Perhatikan masalah pemuatan barang-barang dengan WL = 100 seperti tertera pada tabel berikut:

Nomor Barang	Berat	Nilai
1	40	40
2	50	60
3	20	10
4	10	9
5	20	3
6	40	20
7	30	60

Langkah 1.

Karena minimum berat barang adalah 10 < 100 berarti sedikitnya satu barang dapat dimuat.

Langkah 2.

Karena total semua berat barang adalah 210 > 100, maka tak semua barang dapat dimuat.

Langkah 3.

Urutkan semua barang. Urutan baru adalah sebagai berikut:

Urutan baru	Nomor Barang	Berat	Nilai	Rasio Nilai/berat
1	7	30	60	2
2	2	50	60	6/5
3	1	40	40	1
4	4	10	9	9/10
5	6	40	20	½
6	3	20	10	½
7	5	20	3	3/20

Langkah 4.

Z0 = - 10^{10}; B(1) = 10^{10} SETI(1) = ϕ SETE(1) = ϕ
SETF(1) = {1,2,3,4,5,6,7}
Lanjutkan ke langkah 5

Langkah 5.

Nodus awal merupakan nodus terminal; lanjutkan ke langkah 6.

Langkah 6.

SETF(1) $\neq \phi$, lanjutkan ke langkah 7

Langkah 7.

ISTAR = 1, karena V(1)/W(1) = 2 = $\underset{I \in SETF(1)}{maks} [V(I)/W(I)]$

Langkah 8.

K = K + 1 = 1 +1 = 2
SETI(2) = SETI(1) = ϕ
SETE(2) = SETE(1) $\cup$ ISTAR = {1} tak termasuk ISTAR
COUNT = 1
Nodus 2 merupakan nodus dimana ISTAR = 1 tak termasuk

Langkah 9.

SETI(2) kosong dan barang 1 tak termasuk, karenanya muat sebanyak mungkin barang berawal dari barang paling berharga (barang 2) hingga total berat persis sama dengan WL = 100.

Muat barang 2, berat 40, nilai 60
Muat barang 3, berat 40, nilai 40
Muat barang 4, berat 10, nilai 9
Total **100** **109**

Dengan demikian B(2) = 109.

Langkah 10.

COUNT = 1, karenanya pergi ke Langkah 11

Langkah 11.

Misalkan nodus dimana ISTAR = 1 termasuk (nodus 3).
K = K + 1 = 3
SETI(3) = SETI(1) $\cup$ ISTAR = {1}
SETE(3) = SETE(1) = ϕ
COUNT = 0
Lanjutkan ke langkah 12

Langkah 12.

$$\sum_{I \in SETI(3)} W(I) = \sum_{I \in \{1\}} W(I) = W(1) = 30 \leq 100 = WL$$

lanjutkan ke langkah 9A

Langkah 9A

Hitung B(3) pada nodus 3
Muat barang 1, berat 30, nilai 60
Muat barang 2, berat 50, nilai 60
Muat ½ barang 3, berat 20, nilai 20
Total **100** **140**

Dengan demikian B(3) = 140.

Langkah 10A.

COUNT = 0, lanjutkan ke langkah 13

Langkah 13

SETF(2) = SETF(1) – {1} = {2,3,4,5,6,7}
SETF(3) = SETF(1) – {1} = {2,3,4,5,6,7}
Lanjutkan ke langkah 5A

Langkah 5A.

Karena nodus 3 memiliki batas terbesar, maka K = 3

Langkah 6A

Karena SETF(3) = {2,3,4,5,6,7} lanjutkan ke langkah 7A

Langkah 7A

ISTAR = 2, karena V(2)/W(2) = 6/5 = $\underset{I \in SETF(3)}{maks} [V(I)/W(I)]$

Langkah 8A

K = K + 1 = 3 + 1 = 4
SETI(4) = SETI(3) = {1}
SETE(4) = SETE(3) $\cup$ ISTAR = {2}
COUNT = 1

Nodus 4 adalah nodus dimana barang 1 terasuk dan barang 2 tidak.

Langkah 9B

Muat barang dalam gugus SETI(4) (barang 1), dan sebanyak mungkin barang yang dimulai dengan barang yang paling berharga setelah barang nomor 2), hingga total barang mencapai 100.

Muat barang 1, berat 30, nilai 60
Muat barang 3, berat 40, nilai 40
Muat barang 4, berat 10, nilai 9
Muat ½ barang 5, berat 20, nilai 10
Total **100** **119**
Dengan demikian B(4) = 199

Langkah 10B

COUNT = 1, dengan demikian pergi ke 11B

Langkah 11B

Perhatikan nodus dimana ISTAR = 2 termasuk (nodus 5)
K = K + 1 = 5
SETI(5) = SETI(3) $\cup$ ISTAR = {1,2}
SETE(5) = SETE(3) = ϕ COUNT = 0

Lanjutkan ke langkah 12B

Langkah 12B

$$\sum_{I \in SETI(5)} W(I) = \sum_{I \in \{1,2\}} W(I) = W(1) + W(2) = 30 + 50 = 80 \leq 100 = WL$$

karenanya pergi ke langkah 9C

Langkah 9C

Hitung B(5) pada nodus 5.
Muat barang 1, berat 30, nilai 60
Muat barang 2, berat 50, nilai 60
Muat ½ brng 3, berat 20, nilai 20
Total 100 140
Dengan demikian B(5) = 140

Langkah 10C

COUNT = 0 lanjutkan langkah 13C

Langkah 13C

SETF(4) = SETF(3) – {1,2} = {3,4,5,6,7}
SETF(5) = SETF(3) – {1,2} = {3,4,5,6,7}
Pergi ke langkah 5C

Dan seterusnya. Akhirnya jawab layak yang optimal didapatkan dengan cara memuat barang 1, 2, dan 6 dengan nilai total 130. Bila digunakan penomoran barang awal, barang-barang tersebut adalah barang nomor 2, 3, dan 7.

Program Komputer

Dengan menggunakan bahasa BASIC, dibawah ini diberikan program untuk menyelesaikan permasalahan pemuatan barang (Loading Problem).

```
   REM *************************************************************
   REM *                                                           *
   REM *   PROGRAM MASALAH PEMUATAN BARANG                         *
   REM *     Disusun oleh : Dr. Ir. Sigit Nugroho, M.Sc.           *
   REM *                                                           *
   REM *************************************************************
   CLEAR
   DIM STATUS(500, 25), ITEM(25), W(25), V(25), WT(500), B(500), RATIO(25)
1  READ N, MAXNOD, WL
   FOR I = 1 TO N
      READ ITEM(I), W(I), V(I)
   NEXT I
   REM *******************************************************
   REM JIKA BERAT BARANG MINIMUM MELEBIHI KAPASITAS ANGKUT
   REM MAKA TAK ADA JAWABAN, TAK ADA BARANG YANG DAPAT
DIANGKUT
   REM *******************************************************
   AMINWT = W(1)
   FOR I = 2 TO N
      IF W(I) < AMINWT THEN AMINWT = W(I)
   NEXT I
   IF AMINWT <= WL THEN GOTO 2
   PRINT "TAK ADA JAWABAN YANG LAYAK"
   GOTO 17
   REM *******************************************************
   REM JIKA TOTAL SELURUH BARANG KURANG ATAU SAMA DENGAN
   REM KAPASITAS ANGKUT, MAKA SEMUA BARANG DAPAT DIANGKUT
   REM *******************************************************
2  TOT = W(1)
   FOR I = 2 TO N
      TOT = TOT + W(I)
   NEXT I
   IF TOT > WL GOTO 3
   PRINT "SEMUA BARANG DAPAT DIANGKUT"
   GOTO 17
```

```
   REM ***********************************************************
   REM MENGURUTKAN ITEM SECARA MENURUN BERDASARKAN RASIO
NILAI
   REM TERHADAP BERAT BARANG
   REM ***********************************************************
3  FOR I = 1 TO N
      RATIO(I) = V(I) / W(I)
   NEXT I
   M = N - 1
4  IFLAG = 0
   FOR I = 1 TO M
      IF RATIO(I) >= RATIO(I + 1) GOTO 5
      SAVE = RATIO(I + 1)
      RATIO(I + 1) = RATIO(I)
      RATIO(I) = SAVE
      ISAVE = ITEM(I + 1)
      ITEM(I + 1) = ITEM(I)
      ITEM(I) = ISAVE
      IFLAG = 10
5  NEXT I
   IF IFLAG > 0 GOTO 4
   REM ***********************************************************
   REM INISIALISASI BATAS ATAS NODUS 1 DAN GUGUS YANG TERMASUK,
   REM TIDAK TERMASUK, DAN BEBAS PADA NODUS 1
   REM ***********************************************************
   B(1) = 1000000000
   FOR I = 1 TO MAXNOD
      FOR J = 1 TO N
         STATUS(I, J) = 1
      NEXT J
   NEXT I
   REM ***********************************************************
   REM ATUR K=1 DAN CABANGKAN DARI NODUS K. L NODUS SAAT INI
   REM ***********************************************************
   L = 1
   K = 1
   GOTO 8
   REM ***********************************************************
   REM CARI NODUS DENGAN BATAS TERBESAR, NAMAKAN K
   REM ***********************************************************
6  K = 1
   BIG = B(1)
   FOR KK = 2 TO L
```

```
    IF B(KK) <= BIG GOTO 7
    BIG = B(KK)
    K = KK
7  NEXT KK
   REM ********************************************************
   REM JIKA SEMUA BARANG TERMASUK ATAU TIDAK TERMASU (BUKAN
   REM BEBAS) PADA NODUS SEKARANG INI, JAWAB OPTIMAL SUDAH
   REM DIPEROLEH, BILA TIDAK CABANGKAN DENGAN RATIO MAKSIMUM
   REM V(I)/W(I)
   REM ********************************************************
8  FOR II = 1 TO N
    IF STATUS(K, ITEM(II)) = 1 GOTO 10
   NEXT II
   FOR I = 1 TO N
    PRINT "STATUS("; K; ","; ITEM(I); ")="; STATUS(K, ITEM(I))
   NEXT I
   PRINT "NILAI OPTIMAL TERCAPAI"; B(K)
   PRINT "BERAT BARANG"; WT(K)
   PRINT "ITEM YANG DIMUAT"
   FOR J = 1 TO N
    IF STATUS(K, J) <> -1 GOTO 9
    PRINT J
9  NEXT J
   GOTO 17
10 ISTAR = ITEM(II)
   L = L + 1
   IF L > MAXNOD GOTO 16
   B(K) = -100000000
   FOR J = 1 TO N
    STATUS(L, J) = STATUS(K, J)
    STATUS(L + 1, J) = STATUS(K, J)
   NEXT J
   STATUS(L, ISTAR) = 0
   STATUS(L + 1, ISTAR) = -1
   COUNT = 1
   REM ********************************************************
   REM MENGHITUNG BATAS ATAS NODUS HINGGA JUMLAH BERAT
BARANG
   REM TEPAT SAMA DENGAN KAPASITAS ANGKUT SECARA SEKUENSIAL
   REM ********************************************************
   B(L) = 0
   WT(L) = 0
```

```
   FOR J = 1 TO N
      IF STATUS(L, J) <> -1 GOTO 11
      B(L) = B(L) + V(J)
      WT(L) = WT(L) + W(J)
11  NEXT J
12  FOR J = 1 TO N
      IF STATUS(L, ITEM(J)) <> 1 GOTO 14
      IF (WT(L) + W(ITEM(J))) > WL GOTO 13
      WT(L) = WT(L) + W(ITEM(J))
      B(L) = B(L) + V(ITEM(J))
      GOTO 14
13      DIFF = WT(L) + W(ITEM(J)) - WL
      B(L) = B(L) + (1 - DIFF / W(ITEM(J))) * V(ITEM(J))
      WT(L) = WT(L) + (1 - DIFF / W(ITEM(J))) * W(ITEM(J))
      GOTO 18
14  NEXT J
   REM ********************************************************
   REM JIKA SEMUA CABANG DARI NODUS K SUDAH DI EKSPLORASI
   REM LAKUKAN LANGKAH .....
   REM ********************************************************
18  IF COUNT <> 1 GOTO 6
   COUNT = 0
   L = L + 1
   IF L > MAXNOD GOTO 16
   B(L) = 0
   WT(L) = 0
   FOR J = 1 TO N
      IF STATUS(L, J) <> -1 GOTO 15
      B(L) = B(L) + V(J)
      WT(L) = WT(L) + W(J)
15  NEXT J
   IF WT(L) <= WL GOTO 12
   B(L) = -1000000000
   GOTO 6
16  PRINT "JUMLAH MAXIMUM NODUS TERLAMPAUI"
   GOTO 17
   DATA 7, 50, 100
   DATA 1, 40, 40
   DATA 2, 50, 60
   DATA 3, 20, 10
   DATA 4, 10, 9
   DATA 5, 20, 3
   DATA 6, 40, 20
```

```
   DATA 7, 30, 60
17  END
```

Model Inventori Deterministik

Salah satu masalah yang paling mengganggu dalam produksi dan penjualan produk adalah pengendalian inventori. Banyak perusahaan yang gagal tiap tahunnya karena ketidakmampuan mengendalikan masalah inventori ini. Apakah itu disebabkan karena bahan dasar yang digunakan untuk produksi ataupun dapat berupa hasil atau produk yang menunggu untuk dijual, masalahnya muncul apabila terlalu sedikit atau terlalu banyak item ditahan atau disimpan di inventori. Untuk keperluan ini, item berada di inventori apabila item tersebut menunggu untuk *digunakan* atau *dijual*. Jumlah terbanyak masalah muncul karena terlalu banyak item berada di inventori. Pengukuran biasanya dilakukan dengan menaikkan taraf inventori item yang sering habis (item yang sering digunakan untuk aktifitas produksi); namun demikian, item yang memiliki laju penggunaan atau penjualan yang lambat (item yang jarang digunakan) cenderung diabaikan, dan seringkali kebanyakan masuk di inventori.

Beberapa alasan penyimpanan barang (inventori) diantaranya adalah:

- Penyimpanan barang dilakukan agar perusahaan dapat memenuhi pesanan pembeli dalam waktu relatif singkat
- Pada saat barang sukar diperoleh di pasar, kecuali pada saat musim tiba
- Penyimpanan barang juga dilakukan untuk menekan harga pokok per unit barang.

Faktor-faktor yang dapat mempengaruhi besarnya tingkat persediaan

1. Biaya persediaan barang (*Inventory cost*). Biaya yang berkaitan dengan pemilikan barang dapat dibedakan ke dalam
   a *Holding* atau *Carrying cost*, yaitu biaya yang dikeluarkan karena memelihara atau menyimpan barang; atau

opportunity cost karena melakukan investasi dalam bentuk barang dan bukan investasi lainnya.

b *Ordering cost*, yaitu biaya yang dikeluarkan untuk memesan barang dari supplier untuk mengganti barang yang telah dijual.

c *Stock out cost*, yaitu biaya yang timbul karena kehabisan barang pada saat diperlukan.

2. Sejauh mana permintaan barang oleh pembeli dapat diketahui. Jika permintaan barang dapat diketahui, maka perusahaan dapat menentukan berapa kebutuhan barang dalam suatu periode.
3. Lama penyerahan barang antara saat dipesan dengan barang tiba, atau disebut sebagai "*lead time*" atau "*delivery time*".
4. Terdapat atau tidak kemungkinan untuk menunda pemenuhan pesanan dari pembeli atau disebut sebagai "backlogging" atau "backordering".
5. Kemungkinan diperolehnya diskonto untuk pembelian dalam jumlah besar. Dengan menerima diskonto untuk pembelian dalam jumlah besar, total biaya persediaan barang akan berkurang. Tetapi pembelian dalam jumlah besar akan meningkatkan biaya penyimpanan atau holding cost. Sedangkan pembelian kurang dari jumlah minimum tidak memperoleh diskonto, tetapi biaya pesanan akan meningkat. Dengan demikian terdapat pertimbangan untung rugi dalam keputusan untuk mengambil diskonto atau tidak.

Misalkan kita konsentrasikan saja pada pengendalian satu tipe produk yang sedang ditahan atau disimpan di inventori dengan tipe produk lainnya. Apabila taraf inventori dari satu tipe produk rendah atau produk habis, kita asumsikan ruangan tersedia dan dapat digunakan untuk produk lain. Apabila taraf inventori dari suatu produk mencapai nilai tertentu, dua pilihan dapat dilakukan:

- Pemesanan dapat dilakukan dengan pihak luar untuk pengiriman berikutnya.
- Proses produksi dapat dimulai untuk memproduksi secukupnya untuk menaikkan inventori ke taraf tertentu

Dalam tiap kasus, pemesanan lengkap akan bahan cadangan dapat diterima kedalam inventori sekaligus atau dapat dilakukan secara parsial. Jika pemesanan dilakukan dengan pihak luar, maka biaya pemesanan c_0 harus diperhitungkan setiap pemesanan terjadi. Sebaliknya, bila produk dihasilkan oleh pabrik itu sendiri, c_0 merupakan biaya yang digunakan dalam persiapan produksi. Hal ini berasumsi bahwa produksi dari produk tertentu hanya mengambil sebagian dari waktu produksi yang ada. Inheren dengan pernyataan ini adalah kenyataan bahwa laju produksi lebih besar dari laju penggunaan atau laju penjualan. Pemesanan yang belum dapat dipenuhi yang dapat dipenuhi kemudian hari disebut dengan pemesanan tunda (*backorder*).

Bilamana pemesanan tunda diperbolehkan, perusahaan perlu menentukan

a Jumlah pesanan tunda yang diperbolehkan sebelum menjalankan produksi atau sebelum menerima pesanan sebelumnya dengan pihak luar
b Jumlah satuan untuk produksi bilamana produksi telah mulai *atau* banyaknya satuan pesanan dari pihak luar

Pada dasarnya, masalah dalam pengendalian inventori adalah meminimumkan jumlah biaya yang berkenaan dengan penanganan inventori, yang meliputi

a *Ordering cost*
b *Inventory holding cost*
c *Backordering cost*, jika ada.

Kunci untuk meminimumkan biaya inventori adalah menentukan kapan memesan, berapa banyak dipesan, dan berapa banyak pemesanan tunda diperbolehkan, jika ada. Jika permintaan diketahui dan waktu menerima pesanan (*lead time*) konstan, maka kapan memesan bukan merupakan masalah.

Dalam bab ini, diperlukan **asumsi**:

1. Hanya satu jenis produk yang dianalisa meskipun berbagai tipe disimpan dalam inventori untuk digunakan atau dijual.
2. Periode perencanaan satu tahun.
3. Permintaan produk diketahui dan konstan selama setahun.
4. *Lead time* diketahui dan konstan (waktu antara rekuisisi dan penerimaan pesanan)

Laju Pengiriman Tak Terhingga Tanpa Pemesanan Tunda

Dalam bagian ini kita asumsikan

- Pesanan keseluruhan (lengkap) dikirim sekali (laju pengiriman takterhingga)
- Pesanan yang takterpenuhi merupakan penjualan yang hilang (tidak diperkenankan pemesanan tunda)
- Keempat asumsi umum dipenuhi.

Dengan demikian, hanya terdapat dua biaya yang berkenaan dengan model inventori: *ordering cost* dan *inventory holding cost*.

Analisis produk tunggal dalam inventori adalah sah dalam kasus yang jarang terjadi, tetapi umumnya tidak sah karena ruang dan biaya satu tipe produk berhubungan dengan tipe produk lainnya. Namun demikian, studi pengendalian satu tipe produk akan membentuk basis pengendalain inventori berbagai produk. Sama halnya, asumsi bahwa permintaan produk yang diketahui dan konstan mungkin hanya sah apabila produk adalah bahan dasar yang diperlukan dalam proses produksi.

Pada dasarnya, terdapat dua macam barang inventori: barang yang akan digunakan dalam produksi dan barang yang akan dijual. Barang yang ditahan di inventori yang akan digunakan dalam produksi disebut sebagai bahan dasar. Jika pihak manajemen perusahaan membolehkan cadangan (*stock*) bahan dasar dihabiskan, beberapa kondisi dapat terjadi

- Material substitusi mungkin akan dipakai

- Keadaan darurat mungkin terjadi untuk mendapatkan suplai material baru
- Perusahaan dapat beralih untuk memproduksi produk lainnya
- Proses produksi dapat berhenti total

Untuk masalah inventori yang akan dipelajari, kita gunakan notasi berikut

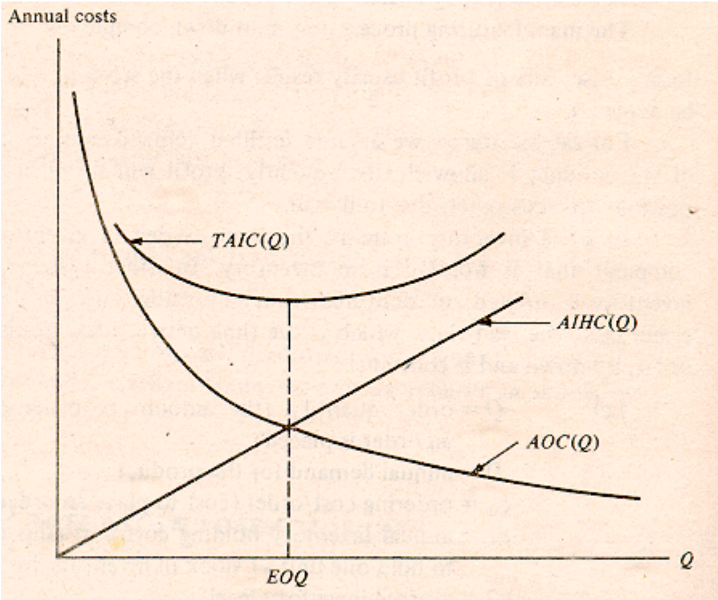
Q	=	Jumlah pesanan (banyaknya pesanan tiap kali melakukan pemesanan)
D	=	Permintaan produk tahunan
c_0	=	Biaya pemesanan tiap melakukan pesanan
c_1	=	Biaya holding-inventori tahunan per unit
$Q/2$	=	Rata-rata taraf inventori
$N = \dfrac{D}{Q}$	=	Banyaknya pesanan per tahun
AOC(Q)	=	$c_0.(D/Q)$
		Biaya pesanan tahunan
AIHC(Q)	=	$c_1.(Q/2)$
	=	Biaya holding-inventori tahunan
TAIC(Q)	=	Total biaya inventori tahunan
	=	AOC(Q)+AIHC(Q)
	=	$c_0.(D/Q)+c_1.(Q/2)$
EOQ	=	Economic Order Quantity atau Jumlah Pesanan Ekonomis, yaitu nilai Q yang meminimumkan TAIC(Q)

Untuk mendapatkan EOQ dapat digunakan cara-cara berikut:

- Cari turunan pertama TAIC(Q) terhadap Q; buatlah hasil turunan ini sama dengan nol; kemudian selesaikan untuk mendapatkan nilai Q.
- Cari Q dengan cara menyelesaikan persamaan AOC(Q) = AIHC(Q)

Untuk kasus ini diperoleh

$$Q = \sqrt{\frac{2c_0 D}{c_1}}$$



Sumber: Gillet, 1982

Teladan:

Dari sebuah perusahaan kecil yang bergerak dalam pembuatan *sleeping bag* diperoleh keterangan bahwa diperkirakan perusahaan dapat memproduksi 1500 *sleeping bag* tersebut selama setahun bila tersedia bahan dan ada permintaan. Bahan dasar tiap *sleeping bag* seharga Rp 48 ribu. Bila diasumsikan bahwa barang tersebut diproduksi dengan laju yang konstan selama 300 hari kerja dalam setahun. Diperkirakan bahwa AIHC untuk bahan dasar tiap barang sebesar 22% dari biaya bahan dasarnya. Sebagai tambahan, biaya sebesar Rp 25 ribu dikenakan setiap kali memesan bahan dasar. *Lead-time* misalnya 7 hari. Dengan demikian

$$TAIC(Q) = 25\left(\frac{1500}{Q}\right) + 22\% * 48 * \frac{Q}{2}$$

Untuk mendapatkan EOQ, pertama kali kita gunakan pendekatan nilai TAIC(Q) dengan cara enumerasi atau daftar hubungan antara Q dan TAIC(Q) mulai Q=7 dan seterusnya hingga TAIC(K)>TAIC(K-1), maka Q=K-1 merupakan EOQ. Kita ambil saja potongan hubungan nilai tersebut sebagai berikut

Q	**TAIC(Q)**
80	891,150
81	890,643
82	890,277
83	890,047
84	***889,949***
85	889,976
86	890,127
87	890,394
88	890,776
89	891,268
90	891,867

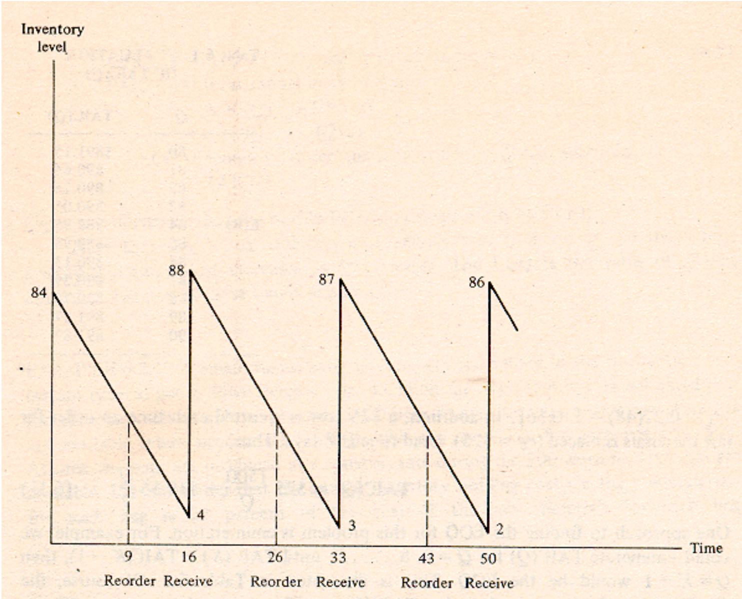
Dari tabel terlihat bahwa EOQ (*Economic Order Quantity*) nya adalah 84, yang berarti bahwa sebanyak 84 satuan bahan yang cukup dipesan tiap melakukan pemesanan. Sebagai tambahan, total biaya dengan *policy* seperti ini adalah

TC = biaya bahan untuk 1500 *sleeping bag* + TAIC(84)
= 1500(48) + 889.949
= 72889.949 (ribu Rp)

Juga dapat diperoleh

N = (1500)/(84) = 17.85 pesanan per tahun

Karena nilai N bukan bilangan bulat, maka di beberapa tahun 17 pesanan dilakukan, sedang di tahun lainnya 18 pesanan.



Sumber: Gillet, 1982

Perlu diketahui bahwa *reorder* atau pemesanan kembali dilakukan 7 hari sebelum waktu dimana bahan dasar kurang dari 5. Laju penggunaan bahan ini adalah 5 (=1500/300). Kita asumsikan pada tahun tersebut diawali dengan bahan untuk membuat 84 *sleeping bag*. Setelah 16 hari tentunya, persediaan bahan tinggal 4, dengan demikian pesanan bahan sebanyak 84 harus dilakukan di hari ke-9 agar bahan tersebut dapat diterima di hari ke-16 yang selanjutnya siap digunakan untuk proses berikutnya. Tentunya di hari ke-16 bahan yang tersedia menjadi 88. Pada hari ke-33 (17 hari kemudian) taraf inventori menjadi 3 dengan demikian pemesanan kembali sudah harus dilakukan pada hari ke-26 sebanyak 84 agar dapat diterima pada hari ke-33 dan seterusnya proses berlanjut dalam tahun tersebut.

Apabila pembuat bahan dasar untuk membuat *sleeping bag* tersebut akan memberikan diskon sebesar 5% bila pesanan dilakukan dalam jumlah besar, yaitu 750 untuk setiap kali pemesanan. Haruskah tawaran ini diterima ?

Dengan tawaran ini, tentunya harga bahan dasar per sleeping bag menjadi 45.60 ribu Rp. Dengan c_0 = 25 ; c_1 = 10.03 ; Q = 750 ; D = 1500 yang menghasilkan

$$\begin{aligned} TC^* &= 1500(45.60) + 25(1500/750) + (10.03)(750)/2 \\ &= 72211.250 \end{aligned}$$

yang tentunya jauh lebih kecil dari TC = 72889.949 Sebaiknya, tawaran tersebut diterima.

Laju Pengiriman Terhingga Tanpa Pemesanan Tunda

Seringkali pesanan keseluruhan stock tidak datang sekaligus. Hal ini terjadi mungkin karena semua pesanan tidak tersedia untuk dikirim, sehingga pesanan tersebut harus dikirim parsial sebagaimana diproduksi atau diperoleh dari pihak lain. Mungkin dapat juga diakibatkan karena permintaan terlalu banyak bila dikirim sekaligus. Untuk mudahnya, diasumsikan bahwa untuk *lead-time* tertentu, stock datang dengan laju yang tetap hingga pesanan keseluruhan terpenuhi atau diterima. Juga diasumsikan bahwa laju datangnya pesanan tersebut lebi besar dari laju penjualan atau penggunaan, dengan demikian menghindari permintaan yang tak terpenuhi. A > U.

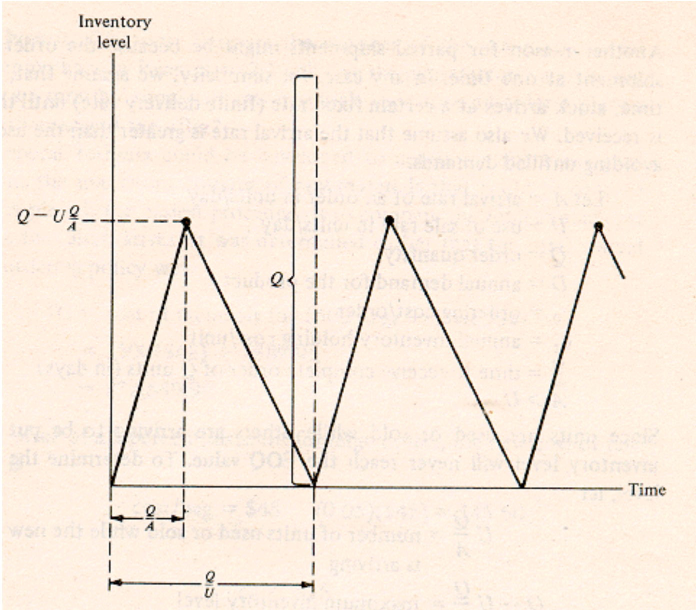
A	=	Laju kedatangan pesanan (satuan per hari)
U	=	Laju penggunaan atau penjualan (satuan per hari)
Q	=	Jumlah pesanan (banyaknya pesanan tiap kali melakukan pemesanan)
D	=	Permintaan produk tahunan
c_0	=	Biaya pemesanan tiap melakukan pesanan
c_1	=	Biaya holding-inventori tahunan per unit
Q/A	=	Berapa kali atau waktu hingga pesanan lengkap sebanyak Q satuan diterima (hari)

Karena adanya barang yang digunakan atau dijual, sedang yang datang harus ditempatkan sebagai inventori, maka taraf inventori tak akan pernah mencapai nilai EOQ. Untuk menentukan taraf rata-rata inventori,

$U\dfrac{Q}{A}$	=	Banyaknya barang digunakan atau dijual pada saat stock baru datang
$Q-U\dfrac{Q}{A}$	=	Taraf inventori maksimum
$\dfrac{1}{2}\left[Q-U\dfrac{Q}{A}\right]$	=	Taraf rata-rata inventori
AOC(Q)	=	$c_0.(D/Q)$
		Biaya pesanan tahunan
AIHC(Q)	=	
	=	$\dfrac{c_1}{2}\left[Q-U\dfrac{Q}{A}\right]$
		Biaya holding-inventori tahunan
TAIC(Q)	=	Total biaya inventori tahunan
	=	AOC(Q)+AIHC(Q)
	=	$c_0.(D/Q)+\dfrac{c_1}{2}\left[Q-U\dfrac{Q}{A}\right]$
EOQ	=	Economic Order Quantity atau Jumlah Pesanan Ekonomis, yaitu nilai Q yang meminimumkan TAIC(Q)

Untuk kasus dengan laju pengiriman terhingga EOQ diperoleh pada

$$Q=\sqrt{\frac{2c_0D}{\left(1-\dfrac{U}{A}\right)c_1}}$$



Sumber: Gillet, 1982

Teladan

Dengan menggunakan permasalahan yang sama seperti pada bagian terdahulu, tanpa diskon, tetapi dengan laju laju pengiriman atau kedatangan 30 per hari, maka

$$Q = \sqrt{\frac{2(25)(1500)}{(10.56)(1-5/30)}} = 92.3$$

dengan

TC = 1500(48) + (25)(1500)/92 + (10.56)(92/2)(1-5/30)
= 72812.410

Laju Pengiriman Takterhingga dengan Pemesanan Tunda

Dalam dua bagian terdahulu, diasumsikan bahwa apabila ada permintaan yang tak dapat dipenuhi dianggap hilang. Namun demikian, permintaan yang tak terpenuhi untuk item yang dapat dijual tidak selalu berarti hilang dalam penjualan. Seringkali, pelanggan akan menunggu item yang dipesannya datang, atau memperbolehkannya memesan barang yang dimaksud. Permintaan yang belum/tidak terpenuhi yang dapat dipenuhi dikemudian hari dikenal sebagai *backorder* atau pemesanan tunda. Sebagaimana diharapkan, umumnya terdapat biaya yang berkenaan dengan pemesanan tunda. Jelasnya, jika semua permintan dapat dipesan tunda, jika tak ada biaya pesan tunda ini, maka tak perlu ada inventori. Kita dapat mengambil pesanan dan kemudian menunggu hingga waktu paling ekonomis untuk memesan untuk pengiriman mendatang/berikutnya. Namun ini kasus yang jarang terjadi. Jika suatu item dapat dipesantunda, umumnya ada biaya yang berkenaan dengan

- Kehilangan kemauan baik (*Loss of goodwill*)
- Penundaan yang berulang-ulang dalam pengiriman
- Tatabuku
- Kehilangan dalam bentuk uang tunai yang mungkin tersedia guna keperluan segera

Masalah dasar selanjutnya adalah menentukan berapa banyak harus dipesan setiap melakukan pemesanan dan berapa banyak pesantunda yang diperbolehkan sebelum menerima pengiriman baru. Kita asumsikan

- Pesan tunda diperbolehkan
- Pesanan lengkap dikirimkan sekaligus
- Keempat asumsi terdahulu dipenuhi

Misalkan

Q	=	Jumlah pesanan (banyaknya pesanan tiap kali melakukan pemesanan)
D	=	Permintaan produk tahunan
B	=	Banyaknya pesantunda diperbolehkan sebelum mengisi atau melengkapi inventori
c_0	=	Biaya pemesanan tiap melakukan pesanan
c_1	=	Biaya holding-inventori tahunan per unit

c_2 = Biaya pesantunda tahunan per unit (biaya sekali pesantunda untuk setahun)

t_1 = Waktu dari penerimaan pesanan hingga taraf inventori mencapai nol lagi

t_2 = Waktu dari taraf inventori nol hingga pesanan yang baru diterima

t_3 = Waktu diantara dua pemesanan berturutan

LT = Lead-time

$N = \dfrac{D}{Q}$ = *Banyaknya pesanan per tahun*

Terdapat relasi, yang dapat dilihat dari gambar, sebagai berikut

$$\frac{t_1}{t_3} = \frac{Q - B}{Q}$$

$$\frac{t_2}{t_3} = \frac{B}{Q}$$

Perlu dicatat bahwa taraf rata-rata inventori selama waktu t_1 adalah $\dfrac{(Q-B)}{2}$ satuan. Karena taraf inventori adalah 0 selama waktu t_2, maka biaya *holding-inventori* tahunan diberikan oleh persamaan

$$AIHC(Q,B) = c_1\left(\frac{Q-B}{2}\right)\frac{t_1}{t_3} = \frac{c_1(Q-B)^2}{2Q}$$

Sementara itu, pesantunda hanya terjadi dalam waktu t_2, sehingga biaya pesantunda tahunan adalah

$$ABC(Q,B) = c_2\,\frac{B}{2}\frac{t_2}{t_3} = \frac{c_2 B}{2}\frac{B}{Q} = \frac{c_2 B^2}{2Q}$$

Biaya pemesanan tahunannya

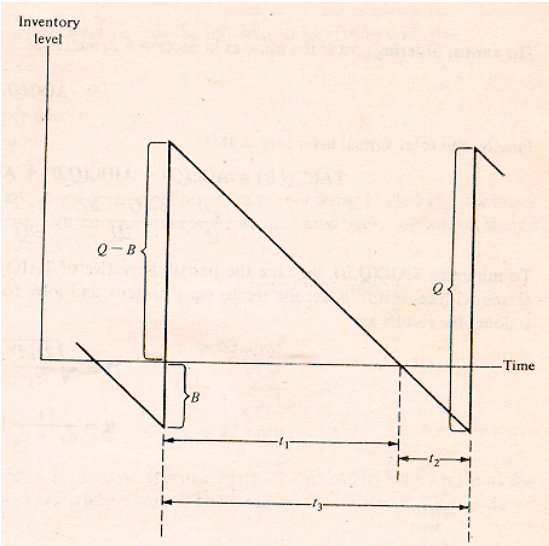
$$AOC(Q) = \frac{c_0 D}{Q}$$

Akhirnya, total biaya inventori tahunan dapat diformulasikan sebagai berikut

$$TAIC(Q,B) = AOC(Q) + AIHC(Q,B) + ABC(Q,B)$$

atau

$$TAIC(Q,B) = \frac{c_0 D}{Q} + \frac{c_1 (Q-B)^2}{2Q} + \frac{c_2 B^2}{2Q}$$



Sumber: Gillet, 1982

Untuk meminimumkan TAIC(Q,B), digunakan turunan parsial TAIC(Q,B) terhadap Q dan terhadap B, buat nilainya sama dengan nol, dan kemudian selesaikan untuk mendapatkan nilai Q dan B yang akhirnya diperoleh

$$Q = \sqrt{\frac{2c_0 D}{c_1} \frac{c_1 + c_2}{c_2}}$$

serta

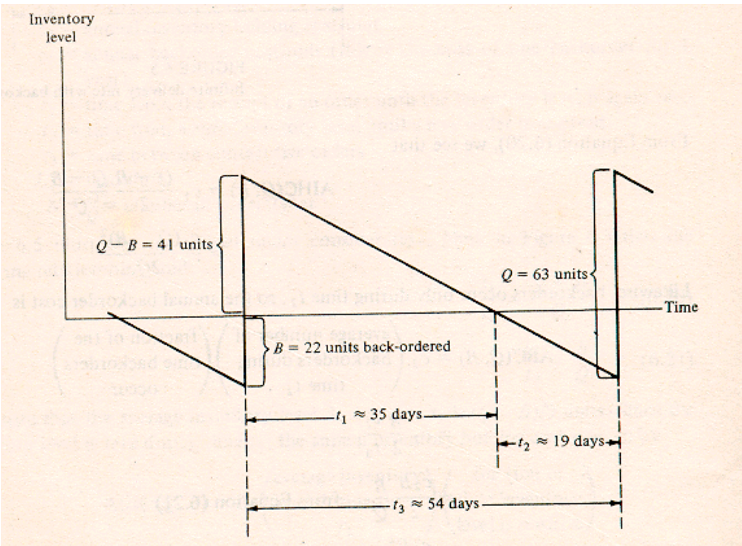
$$B = \frac{c_1}{c_1 + c_2} Q$$

Teladan

Misalkan pengecer memiliki informasi seperti berikut

D	=	350 unit per tahun
c_0	=	Rp 50 ribu per pemesanan
c_1	=	Rp 13.75 ribu per satuan
c_2	=	Rp 25 ribu per satuan
LT	=	5 hari

Untuk meminimumkan total biaya inventori tahunan apabila backordering diperbolehkan, berapa banyak harus dipesan tiap melakukan pemesanan, dan berapa banyak backorder diperbolehkan ?



Sumber: Gillet, 1982

$$Q = \sqrt{\frac{2(50)(350)}{13.75}\frac{13.75 + 25}{25}} \approx 63 \text{ satuan}$$

$$B = \frac{13.75}{13.75 + 25}(63) \approx 22 \text{ satuan}$$

Dengan demikian, kebijakan optimalnya adalah dengan memperbolehkan sekitar 22 backorder sebelum memenuhi inventori yang besarnya sekitar 63 unit.

Laju Pengiriman Terhingga dengan Pemesanan Tunda

Pada bagian ini sama seperti pada kasus laju pengiriman takterhingga kecuali kita asumsikan bahwa tiap pesanan untuk stock selebihnya datang secara parsial dengan laju konstan tiap harinya hingga pesanan lengkap diterima. Dengan demikian, semua asumsi dipenuhi kecuali satu yang baru saja disebutkan.

A	=	Laju kedatangan/pengiriman per hari
U	=	Laju penggunaan atau penjualan per hari
Q	=	Jumlah pesanan (banyaknya pesanan tiap kali melakukan pemesanan)
D	=	Permintaan produk tahunan
B	=	Banyaknya pesantunda diperbolehkan sebelum mengisi atau melengkapi inventori
c_0	=	Biaya pemesanan tiap melakukan pesanan
c_1	=	Biaya holding-inventori tahunan per unit
c_2	=	Biaya pesantunda tahunan per unit (biaya sekali pesantunda untuk setahun)
t_1	=	Waktu dari inventori nol hingga pesanan secara lengkap atau keseluruhan diterima
t_2	=	Waktu dari penerimaan pesanan

		lengkap hingga taraf inventori mencapai nol lagi
t_3	=	Waktu dari saat pemesanan tunda dimulai hingga pesanan baru mulai masuk
t_4	=	Waktu dari pesanan baru masuk hingga semua pesantunda dipenuhi (taraf inventori kembali lagi ke nol)
t_5	=	$\sum_{i=1}^{4} t_i = t_1 + t_2 + t_3 + t_4$
LT	=	Lead-time
$N = \frac{D}{Q}$	=	Banyaknya pesanan per tahun

Dengan notasi diatas, kita peroleh

$$AOC(Q) = \frac{c_0 D}{Q}$$

$$AIHC(Q,B) = c_1 \frac{t_1 + t_2}{t_5} \frac{Q - t_1 U - B}{2}$$

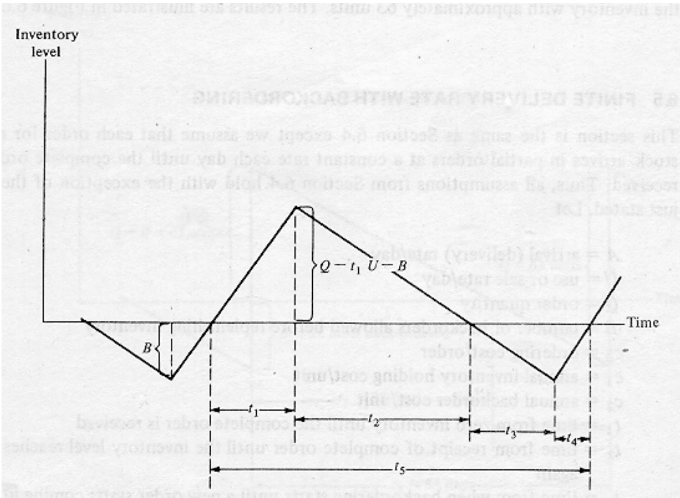
$$ABC(Q,B) = c_2 \frac{t_3 + t_4}{t_5} \frac{B}{2}$$

$$TAIC(Q,B) = \frac{c_0 D}{Q} + \frac{c_1 A}{2Q(A-U)} \left(Q \frac{A-U}{A} - B \right)^2 + \frac{c_2 AB^2}{2Q(A-U)}$$

Akhirnya nilai Q dan B yang meminimumkan TAIC(Q,B) adalah

$$Q = \sqrt{\frac{2c_0 D}{c_1 \left(1 - \frac{U}{A}\right)} \frac{c_1 + c_2}{c_2}}$$

$$B = \frac{c_1}{c_1 + c_2}\left(1 - \frac{U}{A}\right)Q$$



Sumber: Gillet, 1982

Teladan

Dengan menggunakan informasi yang sama pada bagian sebelum ini, dengan tambahan informasi bahwa pesanan datang secara parsial dengan laju 10 per hari. Berapa nilai Q dan B yang harus dipakai untuk meminimumkan total biaya inventori tahunan ?

A = 10 per hari

U = 350/300 = 1.1667 satuan per hari

Dengan demikian

$$Q = \sqrt{\frac{2(50)(350)}{13.75\left(1 - \frac{1.1667}{10}\right)}\left(\frac{13.75 + 25}{25}\right)} \approx 67 \text{ satuan}$$

$$B = \frac{13.75}{13.75 + 25}\left(1 - \frac{1.1667}{10}\right)(67) \approx 21 \text{ satuan}$$

dengan TAIC(67,21)

$$= \frac{(50)(350)}{67} + \left[\frac{13.75(10)}{2(67)(10-1.1667)}\right]\left[(67)\left(\frac{10-1.1667}{10}\right)-21\right]^2 + \frac{25(10)(21)^2}{2(67)(10-1.1667)}$$

= 261.19 + 169.36 + 93.14

= 523.69

Masalah Pengurutan

Masalah pengurutan yang paling terkenal adalah menentukan urutan dimana dua atau lebih pekerjaan harus diproses pada satu atau lebih mesin sehingga mengoptimalkan beberapa kriteria ukuran keefektifan. Permasalahan tersebut mungkin memiliki beberapa kendala, seperti berapa lama setiap pekerjaan dapat diselesaikan, urutan proses untuk tiap pekerjaan pada tiap mesin, atau variabel waktu proses. Ukuran keefektifan dapat berupa *total elapsed time* antara mulainya pekerjaan pertama diproses pada mesin 1 dan selesainya pekerjaan terakhir pada mesin terakhir.

Misalkan terdapat masalah pengurutan sederhana dimana n pekerjaan yang menunggu diselesaikan dimana setiap pekerjaan harus diproses pada dua mesin sedemikian rupa sehingga toal waktu yang digunakan untuk memproses pekerjaan pertama pada mesin pertama hingga pekerjaan terakhir mesin kedua minimum.

Kita asumsikan bahwa waktu proses untuk tiap pekerjaan diketahui dan segera setelah proses di mesin 1 selesai, dilanjutkan proses di mesin 2 apabila mesin ini tidak sedang melakukan proses atau kosong, bila tidak harus menunggu. Tidak ada atau tidak diperbolehkan saling mendahului dari pemrosesan satu jenis pekerjaan ke jenis pekerjaan lainnya. Dengan demikian pekerjaan yang diselesaikan pada urutan ke-3 di mesin pertama, harus juga diselesaikan pada urutan ke-3 di mesin kedua.

Pengurutan dengan 2 Mesin

Berikut ini akan disajikan algoritma pengurutan pekerjaan yang harus diselesaikan dengan menggunakan 2 mesin dengan kriteria seperti yang telah dijelaskan.

Misalkan

T(I,J)	=	Waktu yang diperlukan untuk menyelesaikan pekerjaan ke-I dengan menggunakan mesin Mj I=1,2,…,N dan J=1,2
TIM1(K)	=	Waktu pekerjaan ke-K dalam urutan mulai diproses pada mesin 1 dengan K = 1,2,…,N
TOM1(K)	=	Waktu pekerjaan ke-K dalam urutan selesai diproses dengan menggunakan mesin 1 dengan K = 1,2,…,N
TIM2(K)	=	Waktu pekerjaan ke-K dalam urutan mulai diproses pada mesin 2 dengan K = 1,2,…,N
TOM2(K)	=	Waktu pekerjaan ke-K dalam urutan selesai diproses dengan menggunakan mesin 2 dengan K = 1,2,…,N
TET	=	*Total Elapsed Time* dari awal pekerjaan pertama diproses pada mesin 1 dan akhir pekerjaan terakhir pada mesin 2
ITM1	=	Waktu *idle* pada mesin 1
ITM2	=	Waktu *idle* pada mesin 2

Langkah 1

Pilih elemen T(I,J) terkecil, sebut saja T(I0,J0) dari pekerjaan yang belum diproses. Jika terdapat dua atau lebih elemen yang memiliki T(I,J) terkecil, pilih salah satu sembarang.

Langkah 2

Jika J0 sama dengan 1, berarti bahwa waktu proses tercepat yang terpilih ada pada mesin 1 (M1), karenanya letakkan pekerjaan I0 paling depan dari suatu urutan; bila tidak waktu pemrosesan tercepat yang terpilih berada pada mesin 2 (M2) sehingga pekerjaan I0 harus ditempatkan di bagian terakhir dari urutan.

Langkah 3

Hilangkan pekerjaan I0 dari pembicaraan lebih lanjut.

Langkah 4

Jika semua pekerjaan telah diurutkan, pergi ke langkah 5, bila tidak ke langkah 1.

Langkah 5

Hitung waktu tiap pekerjaan dalam urutan keluar dari mesin 1 (M1)
TOM1(K)=TOM1(K-1)+waktu proses pekerjaan ke-K pada mesin M1
untuk K = 1,2,3,…,N dimana TOM1(0) = 0

Langkah 6

Hitung waktu tiap pekerjaan dalam urutan dimulai pada mesin 1
TIM1(1) = 0
TIM1(K) = TOM1(K-1) untuk K = 2,3,4,…,N

Langkah 7

Secara iteratif hitung waktu tiap pekerjaan mulai dan berakhir pada mesin M2.
TIM2(1) = TOM1(1)
TOM2(K) = TIM2(K) + waktu proses pekerjaan ke-K pada mesin M2
TIM2(K+1) = maks[TOM1(K+1), TOM2(K)] untuk K = 1,2,…,N dan
TIM2(N+1) = 0

Langkah 8

Hitung *total elapsed time* (TET) untuk memproses semua pekerjaan dengan menggunakan dua mesin
TET = TOM2(N)

Langkah 9

Hitung *idle time* untunk mesin M1 dan M2
ITM1 = TOM2(N) – TOM1(N)

$$ITM2 = TOM1(1) + \sum_{K=2}^{N}\left[TIM2(K) - TOM2(K-1)\right]$$

Langkah 10

Selesai
ILUSTRASI

Misalkan ada 10 pekerjaan yang harus diproses dengan menggunakan dua mesin, M1 dan M2 yang harus diproses secara berurutan. Pekerjaan dan lama waktu prosesnya untuk tiap mesin tersaji pada tabel berikut:

Pekerjaan	Mesin 1	Mesin 2
1	20	4
2	10	12
3	3	5
4	10	8
5	5	6
6	2	12
7	8	4
8	7	10
9	3	6
10	4	1

Apabila pengerjaan dari pekerjaan yang ada dengan menggunakan mesin dan urutan 1-2-3-4-5-6-7-8-9-10 maka dapat kita lihat tabel dibawah ini

Pekerjaan	Mesin 1 (M1)		Mesin 2 (M2)		Idle Time
	Masuk	Keluar	Masuk	Keluar	
1	0	20	20	24	20
2	20	30	30	42	6
3	30	33	42	47	0
4	33	43	47	55	0
5	43	48	55	61	0
6	48	50	61	73	0
7	50	58	73	77	0
8	58	65	77	87	0
9	65	68	87	93	0
10	68	72	93	94	0

Dari data diatas terlihat bahwa Total *idel time* untuk mesin kedua = 26

Dapat dibandingkan apabila urutan pengerjaannya adalah 6-3-9-5-8-2-4-1-7-10, yang menghasilkan tabel

Pekerjaan	Mesin 1 (M1)		Mesin 2 (M2)		Idle Time
	Masuk	Keluar	Masuk	Keluar	
6	0	2	2	14	2
3	2	5	14	19	0
9	5	8	19	25	0
5	8	13	25	31	0
8	13	20	31	41	0
2	20	30	41	53	0
4	30	40	53	61	0
1	40	60	61	65	0
7	60	68	68	72	3
10	68	72	72	73	0

Dengan urutan ini *Idle time* untuk mesin kedua berkurang dari 26 menjadi hanya 5.

TELADAN PENYELESAIAN.

Misalkan terdapat delapan pekerjaan yang harus diselesaikan oleh dua teknisi secara berurutan dengan deskripsi sebagai berikut

No	Barang yang perlu direparasi	Teknisi	
		Pendeteksi Kerusakan	Perbaikan Kerusakan
1	Seterika	4	6
2	Toaster	8	3
3	Radio	7	6
4	Mixer	8	4
5	TV	2	6
6	Tape Recorder	1	5
7	Video Player	3	7
8	VCD Player	9	2

Langkah pengerjaan:
Nilai T(I,J) terkecil adalah T(6,1) = 1

Barang yang direparasi ke-6 yaitu Tape Recorder harus diperiksa untuk dideteksi kerusakannya terlebih dahulu oleh teknisi yang bertugas untuk itu.
Hilangkan barang ke-6 dari tabel.
Urutan parsial pengerjaan itu adalah

6							

Selanjutnya nilai minimum T(I,J) berikutnya adalah T(5,1) = T(8,2) = 2
Barang ke-5 yaitu TV berada di urutan sesudah barang ke-6 atau istilahnya sedekat mungkin dengan ujung teknisi pertama yang bertugas mendeteksi kerusakan.
Hilangkan barang ke-5 dari tabel
Urutan parsial pengerjaan itu adalah

6	5						

Selanjutnya nilai minimum T(I,J) tersisa adalah T(8,2) = 2
Barang ke-8 yaitu VCD Player dialokaksikan paling akhir atau istilahnya sedekat mungkin dengan ujung teknisi kedua yang bertugas memperbaiki kerusakan.
Hilangkan barang ke-2 dari tabel
Urutan parsial pengerjaan itu adalah

6	5						8

Bila kita ikuti algoritma yang ada, maka pengisian parsial dari pengurutan barang yang harus diperiksa sehingga meminimumkan total waktu adalah berturut-turut seperti berikut

6	5	7					8

kemudian

6	5	7				2	8

lalu

6	5	7			4	2	8

selanjutnya

6	5	7	1		4	2	8

dan akhirnya kita dapatkan urutan optimalnya

6	5	7	1	3	4	2	8

Sekarang dapat dilihat tabel *elapsed time* nya

Barang ke	Teknisi 1		Teknisi 2		Idle Time
	Masuk	Keluar	Masuk	Keluar	
6	0	1	1	6	1
5	1	3	6	12	0
7	3	6	12	19	0
1	6	10	19	25	0
3	10	17	25	31	0
4	17	25	31	35	0
2	25	33	35	38	0
8	33	42	42	44	4

Dari tabel ini dapat diperoleh informasi:

Idle time teknisi pertama = waktu barang ke-8 selesai dikerjakan teknisi kedua dikurang dengan waktu barang ke-8 selesai diperiksa kerusakannya oleh teknisi pertama. Dalam hal ini 44-42 = 2.

Idle time teknisi kedua = jumlah dari seluruh idle time yang ada pada kolom tabel paling kanan. Dalam hal ini hanya barang ke-6 dan barang ke-8 yang menghasilkan idle time untuk teknisi kedua. 1+4=5.

Pengurutan dengan 3 Mesin

Jawaban optimal untuk permasalahan pengurutan N tugas dengan menggunakan M mesin tidak diperoleh hingga tahun 1967. Jawaban secara umum tidak diperoleh karena terlalu rumit; namun demikian, untuk M = 3, jika minimal salah satu kondisi di bawah ini dipenuhi, jawabnya dapat langsung diperoleh:

1. Waktu proses tercepat mesin pertama sedikitnya sama dengan waktu proses terlama dari mesin kedua
2. Waktu proses tercepat mesin ketiga sedikitnya sama dengan waktu proses terlama dari mesin kedua

Jika sedikitnya salah satu kondisi tersebut dipenuhi, maka prosedur yang dapat digunakan untuk menjawab permasalahn N tugas adalah

1. Jumlahkan waktu proses mesin pertama dan kedua untuk tiap tugas atau pekerjaan.
2. Jumlahkan waktu proses mesin kedua dan ketiga untuk tiap tugas atau pekerjaan.
3. Gunakan jumlah kedua prosedur diatas seakan-akan menyelesaikan permasalahan N tugas atau pekerjaan dengan menggunakan dua mesin. Prosedur yang dipakai sama seperti pada penyelesaian dengan menggunakan dua mesin.
4. Urutan optimal untuk dua mesin pada langkah ketiga diatas juga merupakan urutan optimal permasalahan dengan menggunakan tiga mesin aslinya.

Langkah 1

Misalkan S(I,1) = T(I,1) + T(I,2) dan S(I,2) = T(I,2) + T(I,3) untuk I = 1,2,…,N

Dimana T(I,J) adalah waktu proses pekerjaan I pada mesin J. S(I,J) menunjukkan menunjukkan waktu proses pekerjaan I pada mesin J dalam permasalahan yang baru N pekerjaan dengan 2 mesin.

Langkah 2

Cari jawab permasalahan baru dari N pekerjaan dengan 2 mesin dengan menggunakan algoritma yang telah dijelaskan terdahulu.

Langkah 3

Hitung waktu tiap pekerjaan selesai dikerjakan dengan mesin 1

TOM1(K) = TOM1(K-1) + waktu proses pekerjaan ke-K pada mesin 1 dimana K = 1,2,…,N dengan TOM1(0) = 0

Langkah 4

Hitung waktu tiap pekerjaan mulai pada mesin 1

TIM1(1) = 0

TIM1(K) = TOM1(K-1) untuk K = 2,3,…,N

Langkah 5
Secara iteratif hitung waktu tiap pekerjaan mulai dan selesai pada mesin 2
TIM2(1) = TOM1(1)
TOM2(K) = TIM2(K) + waktu proses pekerjaan ke-K pada mesin 2
TIM2(K+1) = maks[TOM1(K+1),TOM2(K)] untuk K = 1,2,…,N
dengan TIM2(N+1)=0.

Langkah 6
Secara iteratif hitung waktu tiap pekerjaan mulai dan selesai pada mesin 3
TIM3(1) = TOM2(1)
TOM3(K) = TIM3(K) + waktu proses pekerjaan ke-K pada mesin 3
TIM3(K+1) = maks[TOM2(K+1),TOM3(K)] untuk K = 1,2,…,N
dengan TIM3(N+1)=0.

Langkah 7
Hitung *Total elapsed time* untuk memproses semua pekerjaan : TET = TOM3(N)

Langkah 8
Hitung *idle time* untuk mesin 1, 2, dan 3.
ITM1 = TOM3(N) – TOM1(N)

$$ITM2 = TOM1(1) + \sum_{K=2}^{N}\left[TIM2(K) - TOM2(K-1)\right] + \left[TOM3(N) - TOM2(N)\right]$$

$$ITM3 = TOM2(1) + \sum_{K=2}^{N}\left[TIM3(K) - TOM3(K-1)\right]$$

Langkah 9
Selesai

ILUSTRASI

Misalkan terdapat 5 pekerjaan yang harus dikerjakan dengan menggunakan mesin secara berurutan (Mesin 1, Mesin 2, dan Mesin 3). Waktu proses untuk tiap mesin dapat dilihat sebagai berikut:

Peker jaan	Mesin		
	1	2	3
1	4	5	5
2	2	2	6
3	8	3	8
4	10	3	9
5	5	4	7

Peker jaan	Mesin	
	1	2
1	9	10
2	4	8
3	11	11
4	13	12
5	9	11

Dengan menggunakan algoritma yang digunakan untuk menyelesaikan permasalahan N pekerjaan dengan 2 mesin, maka urutan optimalnya adalah : 2-5-1-3-4

Elapsed time dari urutan optimal diatas dapat dilihat sebagai berikut

Peker jaan	Mesin 1		Mesin 2		Mesin 3	
	Msk	Kel	Msk	Kel	Msk	Kel
2	0	2	2	4	4	10
5	2	7	7	11	11	18
1	7	11	11	16	18	23
3	11	19	19	22	23	31
4	19	29	29	32	32	41

Sehingga

ITM1 = TOM3(N) – TOM1(N) = 41 – 29 = 12 unit merupakan *idle time* untuk mesin pertama.

$$ITM2 = TOM1(1) + \sum_{K=2}^{N}\left[TIM2(K) - TOM2(K-1)\right] + \left[TOM3(N) - TOM2(N)\right]$$

= 2 + [(7-4)+(11-11)+(19-16)+(29-22)]+(41-32) = 24 unit merupakan *idle time* untuk mesin kedua, dan

$$ITM3 = TOM2(1) + \sum_{K=2}^{N}\left[TIM3(K) - TOM3(K-1)\right]$$

= 4 + [(11-10)+(18-18)+(23-23)+(32-31)] = 6 unit waktu merupakan *idle time* untuk mesin ketiga.

Masalah Transportasi

Tidak jarang sebuah perusahaan memiliki pabrik untuk komoditas tertentu di beberapa tempat. Produk barang tersebut dikirim ke beberapa lokasi atau *warehouse* yang akhirnya didistribusikan ke pengguna. Misalkan barang tersebut di produksi di sejumlah *m* pabrik dan dikirimkan ke sejumlah *n* warehouses. Apabila biaya untuk mengangkut satu unit (ton, kuintal, satuan, dll) barang dari pabrik ke-I (I= 1,2,3,…,m) sampai warehouse ke-J (J= 1,2,3,…,n), jika tiap pabrik mempunyai kapasitas produksi tertentu dan tiap warehouse mempunyai daya permintaan tertentu, maka masalah transportasi ini adalah menentukan banyaknya satuan yang harus dikirim dari tiap pabrik ke tiap warehouse guna meminimumkan biaya pengiriman dimana semua permintaan dapat terpenuhi.

Misalkan

x_{ij} = Banyaknya barang yang dikirim dari pabrik ke-I (I=1,2,…,m) ke warewaouse ke-J (J=1,2,…,n)

c_{ij} = Besarnya biaya untuk mengirim satu unit barang dari pabrik ke-I (I=1,2,…,m) ke warewaouse ke-J (J=1,2,…,n)

d_j = Besarnya satuan permintaan barang warehouse ke-J (J=1,2,…,n)

s_i = Besarnya kapasitas produksi pabrik ke-I (I=1,2,…,m)

Dalam bentuk matriks, permasalahan tersebut dapat disajikan sebagai berikut

PABRIK	WAREHOUSE				Kapasitas Produksi
	D_1	**D_2**	**…**	**D_n**	
S_1	c_{11}	c_{12}	…	c_{1n}	s_1
S_2	c_{21}	c_{22}	…	c_{2n}	s_2
…			…		…
S_m	c_{m1}	c_{m2}	…	c_{mn}	s_m
Permintaan	d_1	d_2	…	d_n	

Permasalahan dalam bentuk program linier adalah

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

dengan kendala $\sum_{j=1}^{n} x_{ij} = s_i$ (untuk i=1,2,…,m) yang menyatakan banyaknya total barang yang dikirim dari pabrik ke-i untuk semua warehouse harus sama dengan kapasitas produksi pabrik ke-i; $\sum_{i=1}^{m} x_{ij} = d_j$ (untuk j=1,2,…,n) yang menyatakan banyaknya total barang yang dikirim dari semua pabrik ke warehouse ke-j; serta semua x_{ij} tidak negatif.

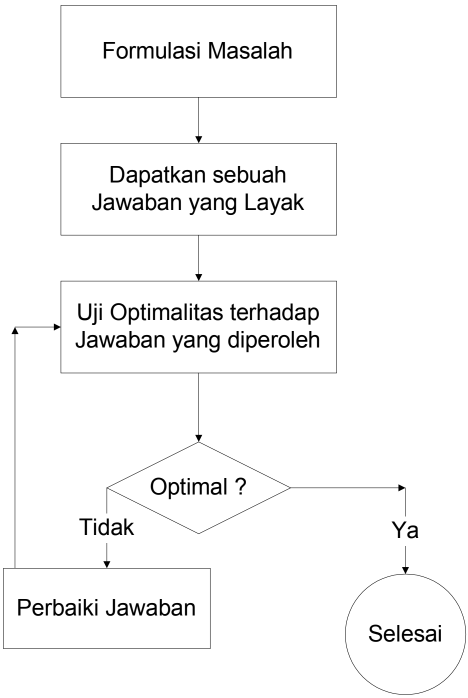
Jawaban optimal dari permasalahan alokasi ini harus merupakan jawaban yang layak atau *feasible solution*, yaitu jawaban yang tak melanggar setiap kendala atau batasan yang dipakai. Agar memenuhi seluruh kendala yang ada, maka kondisi yang perlu untuk menyelesaikan masalah transportasi ini adalah bahwa total semua produksi dari seluruh pabrik sama dengan total permintaan dari seluruh warehouse. Dengan kata lain

$$\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j$$

Jika kondisi ini tak dipenuhi, maka jawaban layak tak akan didapat.

Kondisi kedua adalah jawaban permasalahan ini harus terdiri tepat sebanyak m+n-1 alokasi secara independen. Sebagai contoh apabila matriknya memiliki m=5 dan n=3 maka jawaban yang layak dapat dilakukan dengan mengalokasikan sebanyak 5+3-1=7 sel tak kosong. Matriks alokasi yang mengandung m+n-1 sel tak kosong disebut sebagai jawaban layak dasar (*basic feasible solution*).

Struktur dari permasalahan transportasi ini mirip dengan struktur permasalahan atau model penugasan (*Assignment Model*). Dalam masalah penugasan, jumlah titik suplai harus sama dengan jumlah titik permintaan (m=n); sedangkan dalam masalah



transportasi ukuran $m \neq n$.

ALOKASI AWAL

Apabila matriks biaya sudah diformulasikan, tahap berikutnya adalah membuat matriks alokasi seperti berikut. Permintaan dan penawaran menggunakan satuan ribu ton, sedangkan biaya per ton adalah ribu rupiah.

	D_1	D_2	D_3	D_4	Supply
S_1	5	6	11	4	**51**
S_2	8	7	10	10	**55**
Demand	**26**	**27**	**29**	**24**	**106**

Beberapa metode untuk menentukan alokasi awal akan disajikan, seperti: Metode *Northwest Corner* dan *Vogel Approximation*.

Northwest Corner Method

Metode ini sebetulnya merupakan suatu prosedur yang sistematis memberikan alokasi awal untuk memenuhi batas kendala. Dengan diawali dari pengalokasian atau pengisian kotak paling kiri atas (northwest = barat laut) dari matriks di atas, nilai batas baris dan kolom untuk sel ini dibandingkan, yang lebih kecil dialokasikan sebagai nilai sel ini. Batas baris 51 dan batas kolom 26, oleh karenanya sebanyak 26 satuan sebagai nilai x11 yaitu banyaknya satuan barang yang harus dikirim dari S_1 ke D_1. Alokasi ini memenuhi seluruh permintaan D_1. Sel berikutnya yang harus diisi adalah sel tepat sebelah kanannya. Karena pengalokasian dari S_1 ke D_1, maka akan tersisa sebanyak 51-26=25 satuan barang yang masih dapat dikirim dari S_1. Permintaan D_2 ada sebanyak 27 satuan barang, oleh karenanya sebanyak 25 satuan barang dikirim ke D_2. Untuk pengiriman dari S_1 sudah lengkap, karenanya lanjutkan dengan pengiriman dari S_2. Karena tak ada lagi barang yang perlu dikirim ke D_1, maka alokasi berikutnya adalah ke D_2 dengan cara melengkapi kekurangan permintaan D_2 yaitu 27-25=2 satuan barang. Dengan demikian, sisa barang yang dapat dikirim dari S_2 tinggal 53 satuan. Permintaan D_3 sebanyak 29 satuan barang masih lebih kecil dari 53, karenanya permintaan ini dipenuhi dari S_3, yang akhirnya menyisakan 24 satuan. Sisa ini dialokasikan ke D_4 yang melengkapi menjadi suatu jawaban awal yang layak. m+n-1 = 2+4-1 = 5.

	D_1	D_2	D_3	D_4	Supply
S_1	5 26	6 25	11	4	51
S_2	8	7 2	10 29	10 24	55
Demand	26	27	29	24	106

Total biaya transportasi alokasi awal dengan menggunakan metode Northwest Corner ini 26000*5000 + 25000*6000 + 2000*7000 + 29000*10000 + 24000*10000 = 824000000

Least Cost Method

Metode ini menggunakan biaya transportasi sebagai acuan untuk pengalokasian. Sel yang mempunyai biaya terendah dipilih terlebih dahulu. Banyaknya alokasi tentunya dilihat dari ketersediaan permintaan pada kategori tersebut atau kapasitas produksi yang tersedia yang dapat digunakan untuk kebutuhan itu. Bila kita lihat informasi dari permasalahan yang kita miliki, maka alokasi pertama adalah sebanyak 24 satuan (ribu ton) dari S_1 ke D_4 yang menyisakan 27 satuan barang yang masih dapat dialokasikan dari S_1. Alokasi berikutnya adalah dari S_1 ke D_1 karena memiliki biaya terkecil saat ini, setelah biaya c_{14} dikeluarkan sebagai akibat dialokasikannya barang dari S_1 ke D_4. Permintaan 26 dapat dipenuhi dari S_1 yang akhirnya menyisakan satu satuan barang untuk D_2, dan seterusnya proses ini dilakukan hingga diperoleh hasil seperti berikut

	D_1	D_2	D_3	D_4	Supply
S_1	5 26	6 1	11	4 24	51
S_2	8	7 26	10 29	10	55
Demand	26	27	29	24	106

Total biaya transportasi pada alokasi awal dengan menggunakan metode biaya terkecil ini adalah = 26000*5000 + 1000*6000 + 24000*4000 + 26000*7000 + 29000*10000 = 704000000

Vogel Approximation Method

Prosedur lain yang dapat digunakan untuk menentukan alokasi awal adalah metode pendekatan Vogel atau *Vogel Approximation Method* yang juga dikenal dengan *penalty method* atau metode penalti. Dengan prosedur ini, tiap alokasi dilakukan atas dasar *opportunity cost* atau penalti yang dikenakan jika alokasi tersebut tidak dipilih. Keuntungan dari prosedur ini adalah memberikan jawaban awal yang dekat dengan jawaban optimalnya bila dibandingkan dengan metode *Northwest Corner*.

	D_1	D_2	D_3	D_4	Supply	
S_1	5	6	11	4	51	[1]
S_2	8	7	10	10	55	[1]
Demand	26	27	29	24	106	
	[3]	[1]	[1]	[6]		

Angka dalam [ ] menunjukkan beda biaya terkecil dan terkecil berikutnya dalam satu baris yang sama atau satu kolom yang sama. Sebagai contoh, penalti sebesar [3] dalam kolom D_1 merupakan opportunity cost mengirim D_1 dari S_2 dengan biaya 8 (ribu rupiah) daripada mengirim ke D_1 dari S_1 dengan biaya 5 (ribu rupiah) per ton nya. Hal yang sama juga berlaku untuk baris yang sama. Penalty tertinggi adalah [6] didapat di kolom D_4 sebagai hasil pengiriman barang ke D_4 dari S_1 dengan biaya pengiriman 4 (ribu rupiah) per ton. Jika sel ini tak digunakan, maka setiap ton kiriman ke D_4 akan terkena penalti sebesar 6 (ribu rupiah). Dengan demikian alokasi pertama adalah ke D_4 dari S_1 sebanyak 24 satuan. Hasilnya dapat dilihat sebagai berikut

	D_1	D_2	D_3	D_4	Supply	
S_1	5	6	11	4 24	~~51~~ 27	[1]
S_2	8	7	10	10	55	[1]
Demand	26	27	29	24	~~106~~ 82	
	[3]	[1]	[1]			

Alokasi tersebut akan menyisakan sebanyak 27 (ribu ton) yang dapat dikirim dari S_1 ke lokasi lain. Selanjutnya penalty juga perlu direvisi. Sekarang penalti terbesar adalah [3] pada kolom D_1 dengan demikian alokasikan sebanyak mungkin sehingga batas baris dan kolom dipenuhi dari sel D_1 yang ada yang memiliki biaya terkecil. Dengan demikian kita dapat alokasikan sebanyak 26 (ribu ton) barang ke D_1 dari S_1, karena S_1 masih memiliki 27 (ribu ton) barang yang dapat dikirimkan. Sehingga diperoleh

	D$_1$	D$_2$	D$_3$	D$_4$	Supply	
S$_1$	5 26	6	11	4 24	~~51~~ ~~27~~ 1	~~[1]~~ [5]
S$_2$	8	7	10	10	55	~~[1]~~ [3]
Demand	~~26~~ 0	27	29	24	~~106~~ 82 56	
		[1]	[1]			

Sekarang penalti terbesar adalah [5] untuk baris S$_1$ yang masih menyisakan 1 (ribu ton) yang dapat dikirimkan ke tempat lainnya. Sisa ini harus dikirimkan ke D$_2$ yang memiliki biaya transportasi terendah (setelah dikoreksi) dari S$_1$ yaitu sebesar 6 (ribu rupiah), akhirnya lengkaplah sudah pengiriman barang dari S$_1$.

	D$_1$	D$_2$	D$_3$	D$_4$	Supply	
S$_1$	5 26	6 1	11	4 24	~~51~~ ~~27~~ ~~1~~	~~[1]~~ ~~[5]~~
S$_2$	8	7	10	10	55	~~[1]~~ [3]
Demand	~~26~~ 0	~~27~~ 26	29	24	~~106~~ 82 ~~56~~ 55	

Selanjutnya sudah dapat dipastikan bahwa alokasi akhirnya adalah 26 (ribu ton) harus dikirim dari S$_2$ ke D$_2$ dan 29 (ribu ton) harus dikirim dari S$_2$ ke D$_3$. Demikianlah alokasi awal dengan menggunakan metode pendekatan Vogel.

	D1	D2	D3	D4	Supply
S1	5 26	6 1	11	4 24	51
S2	8	7 26	10 29	10	55
Demand	26	27	29	24	106

Total biaya transportasi pada alokasi awal dengan menggunakan metode pendekatan Vogel ini adalah = 26000*5000 + 1000*6000 + 24000*4000 + 26000*7000 + 29000*10000 = 704000000

Pengujian Optimalitas

Apabila kita telah dapatkan sebuah jawab layak dasar atau *basic feasible solution* selanjutnya yang ingin kita lakukan adalah menguji apakah jawaban kita tersebut sudah optimal, dalam hal ini minimum. Untuk itu kita perlu mengevaluasi sel yang tak terisi Dalam skala yang besar tentunya akan terlalu banyak memakan waktu. Metode MODI atau *Modified Distribution* merupakan teknik yang lebih efisien untuk menyelesaikan masalah tersebut.yang memungkinkan kita membandingkan secara relatif keuntungan alokasi alternatif untuk semua sel kosong secara bersamaan.

Langkah pertama dari metode MODI adalah menghitung nilai baris yang dinotasikan dengan B_i dan kolom yang dinotasikan dengan K_j. Baris-baris dan Kolom-kolom ini tidak memiliki besaran mutlak; nilai-nilainya merupakan jawaban pada saat itu. Dengan demikian mereka dapat dihitung dengan menggunakan persamaan berikut, untuk sel-sel yang terisi :

$$c_{ij} = B_i + K_j$$

Dengan menggunakan nilai-nilai yang didapat dari persamaan ini (melalui sel-sel terisi), kita dapatkan semua nilai B_i dan K_j. Karena sel terisi hanya sebanyak 5 (2+4-1) sedangkan banyaknya variabel ada 6 (2+4, 2 untuk baris dan 4 untuk kolom), maka jawaban yang unik atau khas untuk sistem persamaan diatas tentu tidak kita dapatkan.

Langkah berikutnya adalah mentransfer nilai-nilai yang kita dapatkan ini baik B_i maupun K_j untuk sel-sel yang kosong, sebagai biaya tak langsung. Gunakan untuk semua sel yang kosong

$$c_{ij}^* = B_i + K_j$$

Perbedaan antara nilai sesungguhnya pada permasalahan awal dengan nilai ini (biaya tak langsung) khususnya untuk sel-sel yang kosong, disebut dengan *opportunity cost* untuk sel-sel kosong.

$$oc_{ij} = c_{ij} - c_{ij}^*$$

Jika biaya langsung c_{ij} untuk sel yang tak terisi tidak lebih rendah dari biaya tak langsung c_{ij}^* yang menggambarkan jawaban saat ini, maka penggeseran alokasi ke sel tersebut, tidak akan mengurangi biaya transportasi total.

Sehingga, dalam masalah minimisasi, jika semua *opportunity cost* untuk sel-sel kosong tidak negatif, maka jawaban optimal dari permasalahan telah dicapai.

Untuk metode *Northwest Corner*

$B_1 + K_1 = 5$
$B_1 + K_2 = 6$
$B_2 + K_2 = 7$
$B_2 + K_3 = 10$
$B_2 + K_4 = 10$

Dengan memisalkan $B_1 = 0$, maka berturut-turut diperoleh $K_1 = 5$, $K_2 = 6$, $B_2 = 1$, $K_3 = 9$ dan $K_4 = 9$. Untuk sel-sel yang kosong

$c_{13}^* = B_1 + K_3 = 0 + 9 = 9$
$c_{14}^* = B_1 + K_4 = 0 + 9 = 9$
$c_{21}^* = B_2 + K_1 = 1 + 5 = 6$

Kita dapat periksa

$oc_{13} = 11 – 9 = +2$

oc_{14} = 4 – 9 = -5

oc_{21} = 8 – 6 = +2

Terlihat bahwa sel 14 memiliki *opportunity cost* yang negatif, artinya kita masih dapat lakukan revisi pengalokasian, dengan cara mengalokasikan sebanyak-banyaknya ke sel tersebut. Revisi dapat dilakukan dengan merealokasi dari sel S_1-D_2 ke sel S_1-D_4 sebanyak 24 satuan sehingga menyisakan 1 satuan di sel S_1-D_2, pengalokasi ke D_2 sebanyak 24 satuan dari S_1 menyebabkan pengiriman dari S_2 ke D_4 turun sebanyak 24 satuan, secara bersamaan tentunya hal ini menambah sel S_2-D_2 sebanyak 24 satuan, berubah dari 2 satuan menjadi 26 satuan. Hasil akhir ini tentunya sama dengan alokasi awal metode aproximasi Vogel.

Bila kita verifikasi alokasi awal metode Vogel, kita peroleh

$B_1 + K_1 = 5$

$B_1 + K_2 = 6$

$B_1 + K_4 = 4$

$B_2 + K_2 = 7$

$B_2 + K_3 = 10$

Dengan memisalkan $B_1 = 0$, maka diperoleh $K_1 = 5$, $K_2 = 6$, $K_4 = 4$, $B_2 = 1$ dan $K_3 = 9$.

Sehingga diperoleh

c_{13}^{*} = 0 + 9 = 9

c_{21}^{*} = 1 + 5 = 6

c_{24}^{*} = 1 + 4 = 5

dan diperoleh

oc_{13} = 11 – 9 = +2

oc_{21} = 8 – 6 = +2

oc_{24} = 10 – 5 = +5

yang semuanya sudah bernilai positif. Hal ini menunjukkan bahwa alokasi Vogel dengan data yang kita miliki diatas, selain merupakan jawaban yang layak, jawaban tersebut juga sudah merupakan jawaban yang optimum.

Teknik Evaluasi & Review Proyek

Keberhasilan proyek berskala besar sangatlah tergantung dari kualitas perencanaan, penjadwalan, dan pengendalian berbagai tahapan proyek. Kecuali jika beberapa tipe alat perencanaan dan koordinasi digunakan, banyaknya tahapan tidak perlu terlalu besar, sebelum manajemen kehilangan kendali. Salah satu alat riset operasi yang digunakan pada proyek berskala besar untuk membantu manajemen dalam pengekspedisian dan pengendalian penggunaan personil, material, fasilitas, dan waktu adalah teknik evaluasi dan review program atau *Program Evaluaiton and Review Technique* **(PERT)**. Teknik ini digunakan untuk menunjukkan daerah kritis di dalam proyek sehingga penyesuaian seperlunya dapat dilakukan untuk memenuhi waktu penyelesaian proyek sesuai dengan jadwal. PERT dapat diaplikasikan untuk riset berskala besar dan proyek pengembangan yang memiliki derajat ketidakpastian yang besar, seperti pengembangan produk baru dan pemasaran.

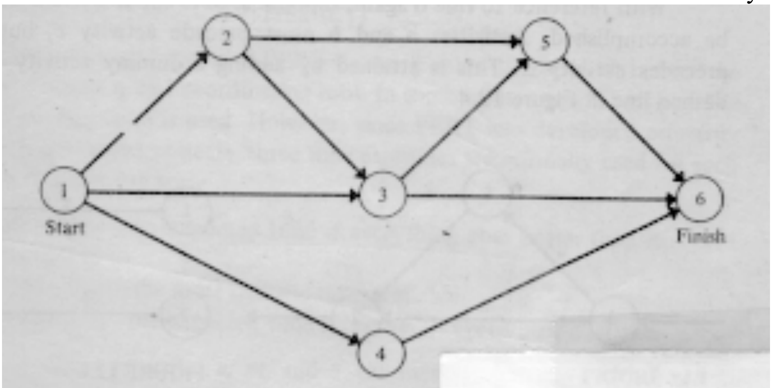
PERT dikembangkan pada thun 1958-1959 sebagai suatu alat riset dan pengembangan program U.S. Navy Polaris Missile. Program Polaris ini selesai 18-24 bulan lebih dini dari yang dijadwalkan. Sejak 1959 PERT telah dipakai dengan sukses dalam setiap industri berskala besar, seperti: industri komputer, industri perfilman dan militer.

Metode Jalur Kritis atau *Critical Path Method* **(CPM)** merupakan alat perencanaan dan pengkoordinasian lainnya. Alat ini dikembangkan dalam industri konstruksi dimana pengalaman sebelumnya digunakan untuk memperoleh estimasi waktu dan biaya berbagai fase proyek. Pengembangan CPM ini disponsori oleh E.I. du Pont de Nemours & Company dengan the Sparry-Rand Corporation. Pertaman kali diaplikasikan pada pembangunan pabrik kimia dan kemudian dalam penghentian produksi pabrik tersebut untuk turun mesin dan maintenance.

Jaringan PERT

Tahap pertama dalam analisis PERT adalah mengidentifikasi subproyek atau tugas yang harus diselesaikan sebelum keseluruhan proyek selesai. Setiap tugas atau pekerjaan yang selanjutnya disebut sebagai **aktivitas** memerlukan informasi waktu dan sumberdaya lainnya. Aktivitas ini menggambar pekerjaan yang harus diselesaikan. Sumberdaya dapat berbentuk tenaga kerja, uang, material, fasilitas, dan/atau ruang. Aktivitas harus jelas sehingga tidak semua aktivitas dapat diselesaikan dalam waktu yang sama. Beberapa aktivitas belum dapat dimulai apabila aktivitas lainnya belum selesai.

Hubungan dari semua aktivitas proyek dapat direpresentasikan dengan suatu jaringan dengan gambar lingkaran atau nodus dan anak panah. Anak panah melambangkan **aktivitas**, dan nodus menggambarkan awal dan selesainya suatu aktivitas. Ekor anak panah melambangkan mulainya suatu aktivitas dan kepala anak panah melambangkan akhir suatu aktivitas. Lingkaran atau nodus disebut dengan **kejadian**. Hal ini menggambarkan selesainya kegiatan yang masuk dan awal kegiatan yang dimulai dari nodus ini. Kejadian tidak memerlukan informasi waktu dan sumberdaya; kejadian ini hanya merupakan tonggak atau pertanda atau batas waktu yang menyatakan selesainya semua kegiatan dan mulainya kegiatan lainnya. Suatu kejadian dikatakan selesai apabila semua aktivitas yang mendahuluinya (dinotasikan dengan anak panah yang menuju ke lingkaran ini) telah selesai dikerjakan. Aktivitas yang berawal dari suatu kejadian tak dapat dimulai bilamana kejadian tersebut belum selesai. Aktivitas yang dimulai dari kejadian i dan berakhir pada kejadian j dinamakan dengan *aktivititas* (i, j).



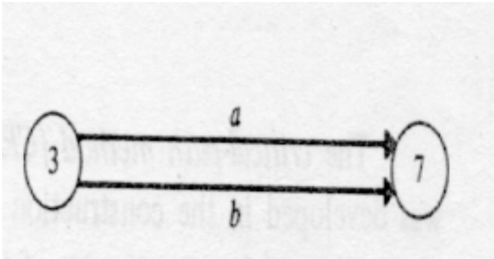
Sumber: Gillet, 1982

Dalam gambar diatas, kejadian 1 menggambarkan awal sebuah proyek, dan kejadian 6 menggambarkan selesainya suatu proyek. Kejadian 3 tidak dapat dikatakan selesai sebelum aktivitas (1,3) dan (2,3) semuanya selesai. Juga aktivitas(3,6) tak dapat dimulai bilamana kejadian 3 belum selesai. Kejadian 6 dikatakan selesai, apabila aktivitas (5,6), (3,6) dan (4,6) semuanya selesai dikerjakan.

Petunjuk dan aturan untuk membangun jaringan yang menggambarkan hubungan antar aktivitas:

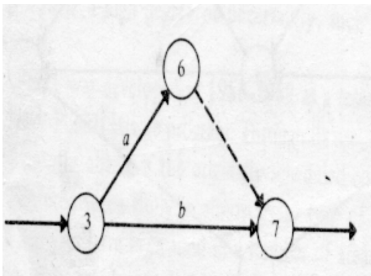
1. Kejadian 1 melambangkan awal suatu proyek. Semua aktivitas yang tidak diawali oleh aktivitas lainnya, harus diawali dengan aktivitas 1 ini.
2. Kejadian M melambangkan akhir suatu proyek, dimana M adalah maksimum jumlah kejadian.
3. Aktivitas (i, j) dimulai dari kejadian i dan berakhir pada kejadian j.
4. Untuk setiap Aktivitas (i, j), $i < j$.
5. Untuk j tertentu, semua aktivitas (i, j) harus diselesaikan terlebih dahulu sebelum kejadian j dikatakan selesai.
6. Setiap aktivitas (i, j) harus khas.

Implikasi dari aturan ke-6: Jika dua atau lebih aktivitas berawal dari kejadian i dan berakhir pada kejadian j, kejadian boneka (*dummy* *events*) dan aktivitas boneka dengan waktu nol perlu dibuat. Sebagai gambaran, misalkan dua aktivitas yang berbeda, *a* dan *b*, yang berawal dari kejadian 3 dan berakhir pada kejadian 7. Hal ini tak boleh digambarkan seperti pada gambar dibawah ini.



Sumber: Gillet, 1982

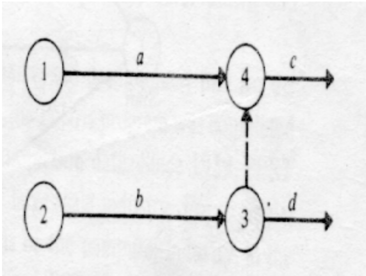
Untuk itu diperlukan sebuah kejadian boneka 6 dan aktivitas boneka (6,7), seperti tertera pada gambar berikut



Sumber: Gillet, 1982

Garis putus-putus dari kejadian 6 ke kejadian 7 melambangkan aktivitas boneka. Kejadian boneka 6 melambangkan selesainya aktivitas *a* dan kejadian 7 melambangkan selesainya aktivitas baik *a* dan *b*.

Gambaran lebih lanjut dengan aturan 6, misalkan aktivitas *a, b, c* dan *d* merupakan empat aktivitas yang harus diselesaikan. Aktivitas *a* dan *b* harus mendahului aktivitas *c*, tetapi hanya aktivitas *b* yang mendahului aktivitas *d*. Hal tersebut dapat dilakukan dengan cara menambah aktivitas boneka yang ditunjukkan dengan garis terputus seperti ilustrasi di bawah ini.

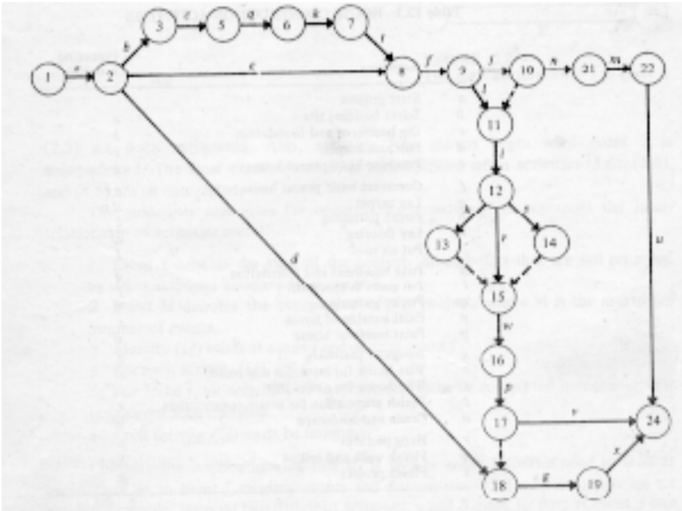


Sumber: Gillet, 1982

Teladan sebuah proyek dalam membangun sebuah rumah

Kegiatan		Kegiatan yang mendahului
a	Awal proyek	
b	Memilih lokasi	a
c	Menggali lantai dasar dan pondasi	b
d	Membeli karpet	a
e	Membeli bahan dasar rumah	a
f	Membangun bahan dasar rumah	e,t
g	Memasang karpet	d,p
h	Menyelesaikan saluran air	i
i	Memasang lantai	j,l
j	Memasang atap	f
k	Mengecor lantai dasar dan pondasi	q
l	Memasang tangga lantai dasar	f
m	Membersihkan kotoran	n
n	Mengecat bagian luar rumah	j
p	Mengecat bagian dalam rumah	w
q	Memasang pipa air	c
r	Memasang jalur telepon dan radio dan TV	i
s	Memasang jalur listrik	i
t	Menyiapkan persiapan konstruksi rumah	k
u	Meratakan dan lanskap	m
v	Memasang gambar pada dinding	p
w	Menyelesaikan pekerjaan dinding dan plafon	h,r,s
x	Selesai proyek	g

Jaringan PERT yang lengkap dapat disajikan pada gambar berikut:



Sumber: Gillet, 1982

Estimasi Waktu Aktivitas (ET)

Estimasi waktu untuk tiap aktivitas dalam jaringan PERT harus diberikan sebelum jaringan tersebut merupakan alat perencanaan dan pengkoordinasian yang bermanfaat. Dalam beberapa kasus, estimasi tunggal terbaik dari tiap aktivitas mungkin digunakan. Namun demikian, karena PERT dikembangkan terutama untuk penelitian dan pengembangan proyek, tiga estimasi waktu umumnya digunakan untuk tiap aktivitas. Mereka adalah:

- *a* = waktu optimis yaitu waktu yang diharapkan jika segala sesuatu berjalan lebih baik tanpa adanya penundaan.
- *m* = waktu yang sangat mungkin yaitu waktu yang paling realistis untuk menyelesaikan aktivitas tersebut.
- *b* = waktu pesimis yaitu waktu yang diperkirakan hanya jika segala sesuatunya berjalan dengan tidak semestinya.

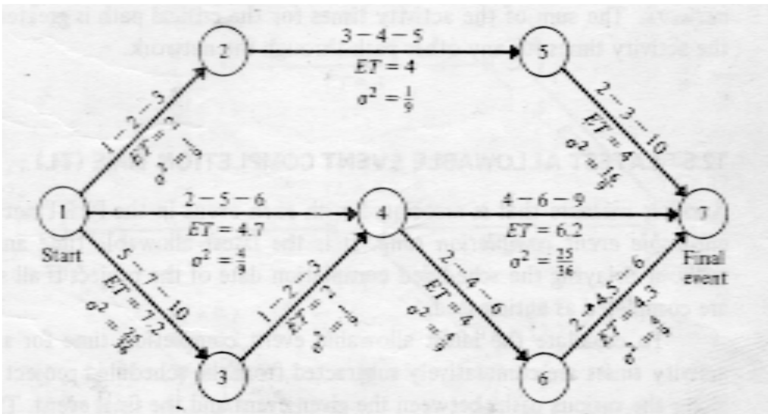
Berdasarkan tiga nilai estimasi *a, m,* dan *b* rataan waktu penyelesaian tiap kegiatan diestimasi dengan

$$ET = \frac{a + 4m + b}{6}$$

dan ragam waktu penyelesaian tiap kegiatan juga dapat diestimasi dengan

$$\sigma_{ET}^2 = \left(\frac{b-a}{6}\right)^2$$

Ketiga nilai estimasi waktu untuk tiap aktivitas terlihat dalam format *a-m-b* diatas anak panah, sedangkan nilai estimasi rataan dan ragam waktu penyelesaian tiap kegiatan di bawah anak panah.



Sumber: Gillet, 1982

Harapan Waktu Penyelesaian Kejadian Tercepat

Tiap aktivitas dalam jaringan PERT memiliki harapan waktu penyelesaian atau rata-rata waktu penyelesaian ET yang dihitung berdasarkan pada nilai estimasi tunggal atau estimasi majemuk berdasarkan *a-m-b*. Dengan demikian, rataan waktu aktivitas dapat digunakan untuk menghitung rataan waktu penyelesaian tiap kejadian. Lebih khusus lagi, untuk setiap kejadian waktu aktivitas dijumlahkan untuk setiap jalur yang mungkin berawal dari kejadian tempat dimulainya aktivitas ke kejadian tertentu, dan jumlah terbesar merupakan harapan waktu penyelesaian kejadian tercepat. Tidak diperlukan menjumlah waktu aktivitas untuk semua kemungkinan jalur untuk tiap kejadian. Untuk itu, kita definisikan

ET(I,J) = rataan waktu penyelesaian aktivitas(I,J)

TE(J) = harapan waktu penyelesaian kejadian tercepat kejadian J

Untuk sebuah nilai J tertentu, sebut saja J^*, maka $TE(J^*)$ diberikan oleh formula

$$TE(J^*) = \max_I \left[TE(I) + ET(I, J^*)\right]$$

dimana fungsi dievaluasi untuk semua I dimana aktivitas dalam bentuk (I,J^*).

Sebagai teladan kita evaluasi gambar diatas

TE(**1**) = 0

TE(2) = TE(1) + ET(1,2) = 0+2 = 2

TE(**3**) = TE(1) + ET(1,3) = 0+7,2 = 7,2

TE(**4**) = $\max\begin{Bmatrix} TE(1)+ET(1,4) \\ TE(3)+ET(3,4) \end{Bmatrix} = \max\begin{Bmatrix} 0+4,7 \\ 7,2+2 \end{Bmatrix}$ = 9,2

TE(5) = TE(2) + ET(2,5) = 2+4 = 6

TE(**6**) = TE(4) + ET(4,6) = 9,2+4 = 13,2

TE(**7**) = $\max\begin{Bmatrix} TE(5)+ET(5,7) \\ TE(4)+ET(4,7) \\ TE(6)+ET(6,7) \end{Bmatrix} = \max\begin{Bmatrix} 6+4 \\ 9,2+6,2 \\ 13,2+4,3 \end{Bmatrix}$ = 17,5

Nilai TE untuk tiap kejadian menunjukkan waktu tercepat dari tiap kejadian yang dapat diselesaikan jika setiap aktivitas J pada setiap jalur sebelum kejadian tertentu diselesaikan dalam tepat ET(J) satuan waktu. Jalur yang diikuti untuk menghitung nilai TE untuk kejadian terakhir disebut sebagai **jalur kritis jaringan**. Jumlah waktu aktivitas untuk jalur kritis lebih besar dari jumlah waktu aktivitas untuk sembarang jalur yang ada dalam jaringan.

Waktu Penyelesaian Kejadian Terlama yang Diperbolehkan (TL)

Ukuran lain yang berkenaan dengan tiap kejadian dalam jaringan PERT adalah waktu kejadian terlama yang masih diperbolehkan (*latest allowable event completion time*). Merupakan waktu terlama yang masih diperobolehkan sebuah kejadian dapat muncul tanpa menunda waktu penyelesaian proyek bilamana semua kejadian yang mendahului selesai seperti yang diharapkan.

Untuk menghitung waktu penyelesaian kejadian terlama yang masih diperbolehkan dari suatu kejadian, waktu kejadian secara kumulatif dikurangkan dari waktu penyelesaian proyek terjadwal lewat berbagai jalur antara kejadian yang dimaksud dan kejadian terakhir. Hasil terkecil dari pengolahan ini merupakan waktu penyelesaian kejadian terlama yang diperbolehkan. Misalkan didefinisikan

TL(I) = waktu penyelesaian kejadian terlama yang diperbolehkan untuk kejadian I

maka untuk nilai I tertentu, katakanlah I^*, maka

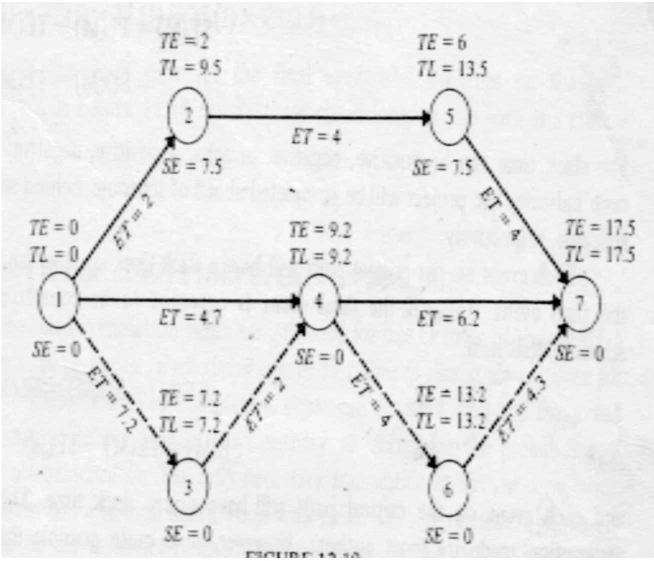
$$TL(I^*) = \min_J \left[TL(J) - ET(I^*, J)\right]$$

dimana J terdefinisi untuk semua aktivitas dalam bentuk (I^*, J). Dengan demikian, untuk M kejadian, persamaan diatas merupakan persamaan rekursif dimana kita dapat memulai menghitung TL(M) kemudian TL(M-1), TL(M-2), hingga TL(1). Waktu penyelesaian terlama yang diperbolehkan untuk kejadian terakhir sama dengan waktu penyelesaian tercepatnya. Namun demikian, TL(M) umumnya merupakan waktu penyelesaian terjadwal dari suatu proyek (SD).

Nilai TL pada gambar berikut dihitung sebagai berikut:
TL(7) = TE(7)
TL(6) = TL(7) – ET(6,7) = 17,5 – 4,3 = 13,2
TL(5) = TL(7) – ET(5,7) = 17,5 – 4,0 = 13,5
TL(4) = min{[TL(7)-ET(4,7)],[TL(6)-ET(4,6)]} = min{(17,5-6,2),(13,2-4,0)} = 9,2
TL(3) = TL(4) – ET(3,4) = 9,2 – 2,0 = 7,2

TL(2) = TL(5) – ET(2,5) = 13,5 – 4,0 = 9,5

TL(1) = min{[TL(2)-ET(1,2)],[TL(3)-ET(1,3)],[TL(4)-ET(1,4)]}
= min{[9,2-2,0],[7,2-7,2],[9,2-4,7]} = 0



Sumber: Gillet, 1982

Waktu Longgar Kejadian (SE)

Waktu longgar kejadian atau *Event Slack Time* untuk tiap kejadian adalah lamanya waktu suatu kejadian dapat ditunda tanpa harus mempengaruhi waktu penyelesaian yang sudah dijadwalkan. Waktu longgar untuk kejadian J diberikan oleh

$$SE(J) = TL(J) – TE(J)$$

Kejadian yang memiliki waktu longgar yang “sedikit” atau “kecil” dikatakan kejadian “kritis” dan harus diperhatikan dalam artian dimonitor secara hati-hati. Untuk menolong waktu longgar suatu kejadian yang “kecil”, sumberdaya dapat dipindahkan dari aktivitas yang mendahuluinya yang memiliki waktu longgar lebih besar ke aktivitas yang mempengaruhi TE untuk kejadian yang memiliki waktu longgar “kecil”. Sebagai contoh dari gambar diatas, kejadian

2 dan 5 memiliki waktu longgar yang lebih besar, sehingga dimungkinkan untuk memindahkan sumberdaya dari aktivitas (1,2), (2,5), dan/atau (5,7) ke aktivitas pada jalur kritis. Dengan demikian akan dapat "mempercepat" aktivitas pada jalur kritis dan "memperlambat" aktivitas diantara kejadian 2, 5 dan 7. Secara umum, prosedur ini akan menurunkan harapan waktu penyelesaian tercepat dari proyek. Waktu longgar untuk tiap kejadian juga dapat dilihat pada gambar diatas.

Jalur Kritis

Jalur terpanjang melalui suatu jaringan PERT disebut sebagai jalur kritis. Ini merupakan jalur yang diikuti untuk memperoleh nilai TE kejadian terakhir.

SE(M) = TL(M) – TE(M) = SD(M) – TE(M)

SE dapat bernilai positif, negatif atau nol. Bila positif, berarti proyek selesai lebih dini daripada yang dijadwalkan, negatif bila lebih lambat, dan nol bila tepat waktu.

Tiap kejadian pada jalur kritis akan memiliki waktu longgar yang sama besarnya dengan waktu longgar pada kejadian terakhir. Dengan demikian, apabila kejadian terakhir diharapkan dapat diselesaikan tepat waktu, maka

TE(M) = SD(M) = TL(M) maka
SE(M) = TL(M) – TE(M) = 0

dan tiap kejadian pada jalur kritis akan memiliki waktu longgar yang bernilai nol. Hal ini yang biasa dipakai sebagai asumsi. Namun demikian, sangatlah mungkin bahwa harapan waktu penyelesaian tercepat dari suatu proyek akan lebih kecil atau lebih besar dari waktu yang sudah dijadwalkan. Jika waktu penyelesaian terjadwal proyek tak dapat dipenuhi [TE(M) > SD(M)], maka SE(M) akan bernilai negatif dan tiap kejadian pada jalur kritis akan memiliki waktu longgar yang negatif pula [-SE(M)]. Di lain pihak, apabila kondisi saat itu menunjukkan bahwa proyek dapat diselesaikan sebelum waktunya [TE(M) < SD(M)], maka SE(M) bernilai positif dan semua kejadian pada jalur kritis akan memiliki waktu longgar yang positif [+SE(M)].

Jika SE(M) bernilai nol atau negatif, sembarang aktivitas pada jalur kritis akan menunda waktu penyelesaian proyek. Sudah tentu, hal ini berasumsi bahwa semua aktivitas lainnya pada jalur kritis selesai seperti yang diharapkan.

Salah satu tujuan analisis PERT adalah untuk menunjukkan jalur kritis, sehingga sumberdaya dari luar sistem atau dari aktivitas pada jalur nonkritis dapat diberikan untuk aktivitas yang berada pada jalur kritis, bila diperlukan. Berkenaan dengan ini, apabila kerja sesungguhnya dimulai pada suatu proyek, jaringan PERT harus diperbaharui (*up-date*) sesuai dengan keperluan (mingguan, bulanan, dll.) dengan segala informasi terkini. Informasi mungkin dalam bentuk tanggal, sumberdaya tersedia, estimasi waktu aktivitas, waktu penyelesaian aktivitas aktual, dan lain sebagainya.

Jalur kritis pada gambar diatas dinotasikan dengan simbol --------- >

Karena TE(7) = TL(7), maka waktu longgar untuk kejadian-kejadian yang berada pada jalur kritis bernilai nol. Namun demikian, semua kejadian pada jalur 1 -> 4 -> 7 memiliki waktu longgar yang kosong, namun jalur bukan merupakan jalur kritis. Agar suatu aktivitas berada pada jalur kritis, maka aktivitas(I,J) harus memenuhi tiga kondisi:

TL(I) – TE(I) = SE(M)

TL(J) – TE(J) = SE(M)

TE(J) – TE(I) = TL(J) – TL(I) = ET(I,J)

dimana SE(M) merupakan waktu longgar kejadian terakhir. Semua aktivitas pada jalur 1 -> 3 -> 4 -> 6 -> 7 pada permasalahn dalam teladan memenuhi ketiga kondisi tersebut, sehingga jalur tersebut merupakan jalur kritis.

Peluang Penyelesaian Kejadian Sesuai Jadwal

Waktu penyelesaian sesuai jadwal hampir diartikan sebagai waktu penyelesaian kejadian terakhir. Rataan dan ragam waktu penyelesaian untuk tiap aktivitas untuk menentukan peluang penyelesaian kejadian sesuai jadwal. Asumsi yang diperlukan hanyalah waktu penyelesaian tiap kejadian memilihi sebaran normal dengan rataan TE dan ragam σ_{TE}^2, dimana σ_{TE}^2 merupakan jumlah ragam waktu kegiatan pada jalur yang digunakan untuk menghitung TE. Sebagai contoh

$$\begin{aligned} TE(4) &= ET(1,3) + ET(3,4) \\ &= 7,2 + 2,0 \\ &= 9,2 \end{aligned}$$

dengan demikian

$$\sigma_{TE(4)}^2 = \sigma_{(1,3)}^2 + \sigma_{(3,4)}^2 = \frac{25}{36} + \frac{1}{9} = \frac{29}{36}$$

Perlu dicatat bahwa $\sigma_{(1,3)}^2$ merupakan ragam waktu penyelesaian aktivitas (1,3) dan $\sigma_{(3,4)}^2$ merupakan ragam waktu penyelesaian aktivitas (3,4).

Untuk kejadian awal pada jalringan PERT atau kejadian dekat dengan awal proyek sebaran normal mungkin tidak terlalu valid; namun demikian, hal tersebut cukup valid untuk kejadian yang besar atau kejadian lebih lanjut dalam jaringan. Waktu penyelesaian kejadian lebih belakang merupakan jumlah beberapa waktu penyelesaian aktivitas, dimana setiap waktu memiliki rataan dan ragam tertentu. Sebagai akibatnya, dengan dalil limit pusat, waktu penyelesaian kejadian tersebut mendekati sebaran normal.

Diasumsikan

T = waktu penyelesaian kejadian

TE = rataan T

σ^2_{TE} = ragam T

SD = waktu penyelesaian sesuai jadwal dari suatu kejadian

maka

$$T \xrightarrow{D} N(TE, \sigma^2_{TE})$$

dibaca : T menyebar menurut sebaran normal dengan rataan TE dan ragam σ^2_{TE}.

Peluang suatu kejadian akan selesai sebelum atau tepat waktu sesuai jadwal adalah

$$P(T \le SD) = \int_{-\infty}^{SD} N(TE, \sigma^2_{TE})dt = \int_{-\infty}^{\frac{SD-TE}{\sigma_{TE}}} N(0,1)dt$$

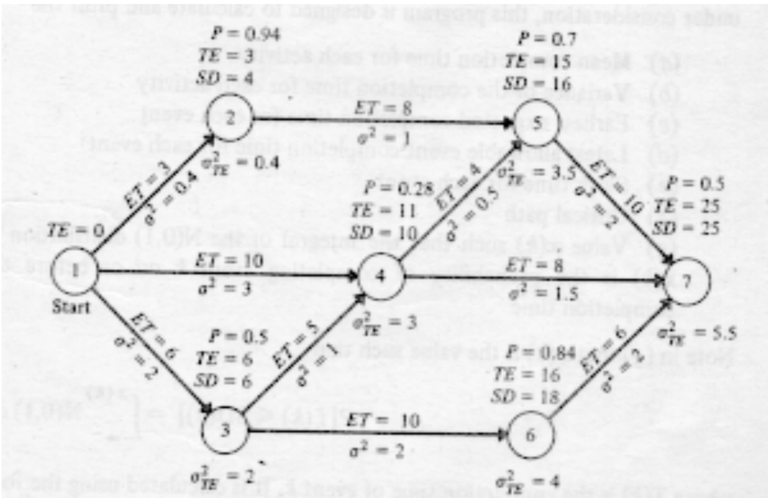
dan sebagai contoh

$$P[T \le SD(7)] = P(T \le 25) = \int_{-\infty}^{25} N(25;5{,}5)dt = \int_{-\infty}^{\frac{25-25}{\sqrt{5{,}5}}} N(0,1)dt = 0{,}5$$

Dari perhitungan terlihat bahwa terdapat kesempatan 50-50 untuk menyelesaikan pekerjaan sesuai dengan jadwal. Keseluruhan peluang penyelesaian kejadian sebelum atau sesuai dengan jadwal dapat ditabelkan sebagai berikut:

Kejadian	TE	SD	σ^2_{TE}	$x = \dfrac{SD - TE}{\sigma_{TE}}$	$P(T \leq SD)$
2	3	4	0,4	+1,58	0,94
3	6	6	2,0	+0,00	0,50
4	11	10	3,0	-0,58	0,28
5	15	16	3,5	+0,53	0,70
6	16	18	4,0	+1,00	0,84
7	25	25	5,5	+0,00	0,50

Kejadian yang memiliki peluang penyelesaian kejadian kecil atau rendah, menunjukkan bahwa kejadian tersebut harus "dipercepat" waktu penyelesaian aktivitas sebelumnya.



Sumber: Gillet, 1982

Esti masi	Aktivitas									
	1→2	1→3	1→4	2→5	3→4	3→6	4→5	4→7	5→7	6→7
a	2,20	1,51	5,00	6,00	2,00	5,51	1,76	4,00	5,00	2,51
m	2,45	6,12	9,90	7,50	5,00	10,21	4,06	8,16	10,38	5,62
b	6,00	10,00	15,39	12,00	8,00	14,00	6,00	11,35	13,49	11,00

Program Komputer

```
REM   ***************************************************************************************************
REM   *                                                                                                 *
REM   *           TEKNIK EVALUASI DAN REVIEW PREOGRAM  (PERT)                                           *
REM   *                              SIGIT NUGROHO                                                      *
REM   *                                                                                                 *
REM   ***************************************************************************************************
REM
REM   M adalah banyaknya EVENT maksimum
REM   N adalah banyaknya AKTIVITAS maksimum
REM
REM   Membaca Data Banyaknya EVENT dan AKTIVITAS maksimum yang juga
REM   mencakup dari dan ke, waktu Optimis, Realistis, dan Pesimistis
REM
      READ M
      READ N
      DIM C(N, 7), CPATH(N), SIGTE(N), SD(N)
      DIM TE(N), TL(N), SE(N), X(N), FLAG(N)
      FOR I = 1 TO N
            FOR J = 1 TO 5
                  READ C(I, J)
            NEXT J
            FLAG(I) = 0
      NEXT I
REM
REM   Membaca Data Jadwal Penyelesaian suatu Peristiwa dengan mengasumsi
REM   kan bahwa SD(1) = SD(M) = 0
REM
      FOR I = 1 TO M
            READ SD(I)
      NEXT I
REM
REM   Menghitung Rata-rata dan Ragam Waktu penyelesaian setiap peristiwa
REM   berdasarkan waktu optimis, realistis, dan pesimistis, serta mence
REM   tak I  J  AT(I,J)  MT(I,J)  BT(I,J)  ET(I,J)  SIG(I,J)
REM
      FOR I = 1 TO N
            C(I, 6) = (C(I, 3) + 4 * C(I, 4) + C(I, 5)) / 6
            C(I, 7) = (C(I, 5) * C(I, 5) - 2 * C(I, 5) * C(I, 3) + C(I, 3) * C(I, 3)) / 36
      NEXT I
      PRINT "Hasil Olahan PERT by SIGIT NUGROHO"
      PRINT "____________________________________"
      PRINT
      PRINT
      PRINT "I   J   AT(I,J) MT(I,J) BT(I,J) ET(I,J) SIG(I,J)"
      PRINT "___________________________________________________"
```

```
      PRINT
      FOR I = 1 TO N
            FOR J = 1 TO 7
                  PRINT USING "##.## "; C(I, J);
            NEXT J
      PRINT
      NEXT I
      PRINT
      PRINT
REM
REM   Menghitung rata-rata waktu penyelesaian tercepat untuk EVENT J
REM   atau TE(J)
REM
      TE(1) = 0
      SIGTE(1) = 0
      FOR J = 2 TO M
            TMAX = 0
            SIGMAX = 0
            FOR K = 1 TO N
                  IF C(K, 2) <> J THEN GOTO 10
                  TE(J) = TE(C(K, 1)) + C(K, 6)
                  SIGTE(J) = SIGTE(C(K, 1)) + C(K, 7)
                  IF TE(J) <= TMAX THEN GOTO 10
                  TMAX = TE(J)
                  SIGMAX = SIGTE(J)
10              NEXT K
            TE(J) = TMAX
            SIGTE(J) = SIGMAX
      NEXT J
REM
REM   Menghitung rata-rata waktu terlama yang masih diperbolehkan
REM   untuk EVENT J atau TL(J)
REM
      TL(M) = TE(M)
      SD(M) = TL(M)
      FOR F = 2 TO M
            I = M - F + 1
            TMIN = 100000
            FOR H = 1 TO N
                  K = N - H + 1
                  IF C(K, 1) <> I THEN GOTO 20
                  TL(I) = TL(C(K, 2)) - C(K, 6)
                  IF TL(I) >= TMIN THEN GOTO 20
                  TMIN = TL(I)
20              NEXT H
            TL(I) = TMIN
      NEXT F
REM
```

```
REM    Menghitung SLACK TIME setiap EVENT J atau beda waktu antara
REM    TL(J) - TE(J) serta mencetaknya
REM
       PRINT
       PRINT
       PRINT "  J    TE(J)   TL(J)   SE(J)"
       PRINT "______________________________"
       PRINT
       FOR J = 1 TO M
            SE(J) = TL(J) - TE(J)
            PRINT USING "###  ###.###  ###.###  ###.### "; J; TE(J); TL(J); SE(J)
       NEXT J
REM
REM    Menentukan Jalur Kritis
REM
       FOR K = 1 TO N
            I = C(K, 1)
            J = C(K, 2)
            IF ABS(TL(I) - TE(I) - SE(M)) > .0001 THEN GOTO 30
            IF ABS(TL(J) - TE(J) - SE(M)) > .0001 THEN GOTO 30
            IF ABS(TE(J) - TE(I) - TL(J) + TL(I)) > .0001 OR ABS(TL(J) - TL(I) - C(K, 6)) >
.0001 THEN GOTO 30
            FLAG(K) = 1
30     NEXT K
       CPATH(1) = 1
       KP = 1
       FOR K = 1 TO N
            I = C(K, 1)
            J = C(K, 2)
            IF FLAG(K) = 0 THEN GOTO 40
            IF I <> CPATH(KP) THEN GOTO 40
            KP = KP + 1
            CPATH(KP) = J
40     NEXT K
       PRINT
       PRINT
REM
REM    Mencetak Jalur Kritis
REM
       PRINT
       PRINT "Jalur Kritis"
       PRINT "____________"
       PRINT
       FOR I = 1 TO KP
           PRINT CPATH(I);
       NEXT I
REM
REM    Menghitung Titik Kritis Kurva Normal Baku N(0,1) untuk menen
```

```
REM     tukan peluang penyelesaian suatu EVENT dapat diselesaikan
REM     sebelum Jadwal Waktu Penyelesaian
REM
        PRINT
        PRINT ;
        FOR K = 2 TO M
           X(K) = (SD(K) - TE(K)) / (SIGTE(K)) ^ (.5)
        NEXT K
        PRINT ;
        PRINT
REM
REM     Mencetak K  TE(K)  SD(K)  SIGTE(K)  X(K)
REM
        PRINT
        PRINT " K     TE(K)   SD(K)   SIGTE(K)     X(K)"
        PRINT "________________________________________________"
        PRINT
        FOR K = 2 TO M
              PRINT USING "##  ###.###  ###.###  ###.###  ###.###"; K; TE(K); SD(K);
SIGTE(K); X(K)
              PRINT
        NEXT K
        END
REM
REM     Masukkan Data Disini
REM
        DATA 7, 10
        DATA 1, 2, 2.20, 2.45, 6.00
        DATA 1, 3, 1.51, 6.12, 10.00
        DATA 1, 4, 5.00, 9.90, 15.39
        DATA 2, 5, 6.00, 7.50, 12.00
        DATA 3, 4, 2.00, 5.00, 8.00
        DATA 3, 6, 5.51, 10.12, 14.00
        DATA 4, 5, 1.76, 4.06, 6.00
        DATA 4, 7, 4.00, 8.16, 11.35
        DATA 5, 7, 5.00, 10.38, 13.49
        DATA 6, 7, 2.51, 5.62, 11.00
        DATA 0, 4, 6, 10, 16, 18, 0
```

Teori Pengambilan Keputusan

Walaupun kita tahu akan berbagai definisi atau pengertian tentang keputusan, disini kita pakai definisi keputusan sebagai pemilihan diantara berbagai aksi atau tindakan yang memenuhi satu atau lebih tujuan.

Dalam tulisan ini akan disajikan dasar-dasar teori pengambilan keputusan secara kuantitatif. Untuk itu kiranya kita akan sajikan terlebih dahulu beberapa notasi atau istilah yang akan digunakan. Variabel dalam suatu keputusan yang dapat dikendalikan disebut dengan *decision alternative* atau pilihan keputusan yang disingkat dengan DA_i yang mungkin merupakan $DA_1, DA_2, \ldots, DA_m$. Variabel ini dikatakan dapat dikendalikan karena merupakan pilihan yang dibuat oleh pengambil keputusan. Dalam kebanyakan kasus pilihan keputusan yang layak dievaluasi, dan dipilih salah satu darinya. Keluaran dari sembarang DA_i tergantung pada interaksinya dengan variabel yang disebut dengan *state of nature* atau keadaan yang sebenarnya yang dinotasikan dengan S_j yang dapat juga berupa $S_1, S_2, \ldots, S_n$. Dari sisi pengambil keputusan, S_j ini merupakan peubah bebas, dimana pengambil keputusan tidak dapat mengendalikan keadaan sebenarnya mana yang akan terjadi, bila salah satu keputusan telah dipilih.

Matriks Payoff

Suatu cara yang baik untuk merangkum semua elemen dari permasalahan keputusan dapat disajikan dalam bentuk matriks *payoff*. Bentuk matriks menunjukkan hubungan antara variabel DA_i dengan S_j dalam array berdimensi dua yang mencakup m baris dan n kolom. Elemen atau anggota dari matriks menunjukkan perkiraan *payoff* p_{ij} sebagai hasil dari interaksi antara DA_i dengan S_j. Perhatikan matriks dibawah ini.

Pilihan Keputusan	Keadaan Sesungguhnya yang dapat terjadi			
	S_1	S_2	…	S_n
DA_1	p_{11}	p_{12}	…	p_{1n}
DA_2	p_{21}	p_{22}	…	p_{2n}
…	…	…	…	…
DA_m	p_{m1}	p_{m2}	…	p_{mn}

Pengambilan Keputusan pada Lingkungan yang pasti.

Dengan kondisi yang pasti, pengambil keputusan sudah mengetahui terlebih dahulu *state of nature* yang mana yang akan terjadi. Ini berarti bahwa setiap pilihan keputusan atau *decision alternative* hanya akan memiliki satu keluaran, dan *payoff* atau biaya dalam tiap kasus adalah tetap.

Perhatikan tabel berikut

Keputusan	Pendapatan
K1: Keluar Rp 1000,-	Rp 185000,-
K2: Keluar Rp 4000,-	Rp 530000,-
K3: Keluar Rp 7000,-	Rp 670000,-
K4: Keluar Rp 9000,-	Rp 420000,-

Seorang manajer advertensi harus memutuskan untuk mengeluarkan uang untuk membayar Rp. 1000,-, Rp. 4000,-, Rp. 7000,- atau Rp 9000,- pada rancangan paket untuk tiap pengiriman satuan produk tertentu. Besarnya keluaran tersebut menunjukkan adanya perbedaan rancangan paket. Angka disebelah kanannya menunjukkan besarnya pendapatan yang diperoleh sebagai akibat keputusan yang diambil. Manakah keputusan yang harus anda ambil, jika anda seorang manajer ?

Pengambilan keputusan beresiko mempunyai implikasi bahwa walaupun sembarang keadaan yang sebenarnya *(state of nature)* dapat terjadi, pengambil keputusan dapat mengestimasi peluang munculnya setiap keadaan tersebut. Hal ini berarti bahwa kemungkinan *payoff* pada kondisi tertentu dapat diboboti dengan peluang munculnya setiap keadaan. Dengan demikian kita dapat menggunakan konsep *Expected value* atau Nilai Harapan, untuk menentukan keputusan mana yang akan diambil.

Sebagai teladan dapat dilihat permasalahan berikut. Misalkan keputusan untuk mengiklankan atau tidak mengiklankan produk pertanian di televisi mengandung dua *state of nature* yaitu: volume penjualan rendah atau volume penjualan tinggi. Dengan melakukan iklan, apabila volume penjualan tinggi, maka pendapatan yang akan diperoleh adalah Rp. 27 juta dan bila volume penjualan rendah maka perusahaan akan mengalami kerugian Rp. 15 juta. Apabila tidak melakukan iklan di televisi, dan volume penjualan rendah, maka pendapatan dapat mencapai Rp 5 juta dan bila penjualan tinggi, pendapatan bahkan dapat mencapai Rp 12 juta. Apabila peluang untuk memperoleh volume penjualan rendah hanya 0.30 dan dengan demikian peluang memperoleh volume penjualan tinggi 0.70, keputusan manakah yang harus diambil ?

Matriks *payoff*

Keputusan	Volume Penjualan	
	Rendah	Tinggi
Iklan di TV	-15	27
Tidak Iklan di TV	5	12

Dengan menggunakan P(Rendah) = 0.30 dan P(Tinggi) = 0.70, maka kita dapat susun matriks nilai harapan sebagai berikut :

Matriks Nilai Harapan

Keputusan	Volume Penjualan	
	Rendah	Tinggi
Iklan di TV	-4.5	18.9
Tidak Iklan di TV	1.5	8.4

Nilai harapan untuk melakukan iklan di TV dengan demikian = -4.5+18.9 = 14.4 dan untuk tidak melakukan iklan di TV = 1.5 + 8.4 = 9.9. Dengan berdasarkan pada nilai harapan secara umum, maka pengambil keputusan akan memilih untuk mengiklankan produksnya melalui media TV.

Nilai harapan mengandung arti bahwa dalam waktu yang lama di masa mendatang, maka secara rata-rata bahwa pendapatan yang akan diperoleh adalah sebesar Rp 14.4 juta bila dilakukan iklan di TV untuk produk tersebut.

Teladan lain: Beragamnya permintaan akanikan "Baramundi" selama 20 hari Sabtu berturut-turut dapat disajikan dalam tabel berikut:

Kilogram terjual	Banyaknya hari Sabtu / frekuensi	Relatif Frekuensi
100	6	0.30
101	6	0.30
102	4	0.20
103	2	0.10
104	1	0.05
105	1	0.05
Total	**20**	**1.00**

Kios ikan tersebut membeli dari nelayan sebesar Rp 12000 per kilogram, dan menjualnya dengan harga eceran Rp 20000 per kilogram. Kios tersebut tutup pada hari Minggu dan buka lagi pada hari Senin. Dengan demikian, ikan yang tidak terjual pada hari Sabtu harus dibuang, karena membusuk dan tak dapat dijual lagi.

Keputusan untuk membeli berapa kilogram pada tiap hari Sabtu harus diambil agar secara rata-rata dimasa mendatang pendapatan akan maksimum ?

Matriks *payoff* (dalam ratus ribu rupiah) dengan demikian adalah sebagai berikut:

Beli dari Nelayan / Jual	Permintaan Pasar					
	S_1 : 100	**S_2 : 101**	**S_3 : 102**	**S_4 : 103**	**S_5 : 104**	**S_6 : 105**
DA_1 : 100	8.00	8.00	8.00	8.00	8.00	8.00
DA_2 : 101	7.88	8.08	8.08	8.08	8.08	8.08
DA_3 : 102	7.76	7.96	8.16	8.16	8.16	8.16
DA_4 : 103	7.64	7.84	8.04	8.24	8.24	8.24
DA_5 : 104	7.52	7.72	7.92	8.12	8.32	8.32
DA_6 : 105	7.40	7.60	7.80	8.00	8.20	8.40

Jika hanya 105 kg ikan dibeli, **DA_6**, dan hanya 103 kg terjual, **S_4**, maka nilai matriks payoff bersyaratnya adalah **P_{64}** adalah (20000 x 103) – (12000 x 105) = 800000.
Sedangkan matriks nilai harapannya (Rp ratus ribu) adalah

Beli dari Nelayan / Jual	Permintaan Pasar					
	S_1 : 100	**S_2 : 101**	**S_3 : 102**	**S_4 : 103**	**S_5 : 104**	**S_6 : 105**
DA_1 : 100	2.400	2.400	1.600	0.800	0.400	0.400
DA_2 : 101	2.364	2.424	1.616	0.808	0.404	0.404
DA_3 : 102	2.328	2.388	1.632	0.816	0.408	0.408
DA_4 : 103	2.292	2.352	1.608	0.824	0.412	0.412
DA_5 : 104	2.256	2.316	1.584	0.812	0.416	0.416
DA_6 : 105	2.220	2.280	1.560	0.800	0.410	0.420

Nilai harapan untuk keputusanmembeli ikan dari nelayan (untuk dijual lagi) untuk 100, 101, 102, 103, 104, dan 105 kg berturut-turut adalah sebagai berikut: 8.000; 8.020; 7.980; 7.900; 7.800; dan 7.690. Dengan demikian untuk jangka panjang, sebaiknya keputusan yang diambil adalah membeli ikan sebanyak 101 kg untuk penjualan hari Sabtu.

Pengambilan Keputusan pada Lingkungan yang tak pasti.

Pengambil keputusan kadang menemui atau menghadapi situasi dimana tak ada landasan untuk menduga peluang dari berbagai keadaan yang sesungguhnya. Karenanya, pengambilan keputusan dalam hal ini dilakukan pada lingkungan yang tak pasti. Sialnya, kebanyakan keputusan penting biasanya harus dibuat pada kondisi-kondisi seperti ini. Misalnya pertanyaan apakah perusahaan akan mengenalkan produk barunya atau tidak. Beberapa teknik telah dikembangkan dengan landasan yang konsisten untuk kondisi lingkungan yang tak pasti.

Kriteria Maximax

Kriteria maximax ini merupakan aturan keputusan yang sering digunakan oleh kelompok optimis. Leonid Hurwicz beralasan bahwa tak ada basis untuk berasumsi bahwa keadaan sesungguhnya tidaklah beragam dibandingkan dengan pengambil keputusan. Akhirnya, orang akan memperoleh keberuntungan dan menang sesekali. Kriteria maximax memungkinkan pengambil keputusan yang optimis untuk memberikan nilai yang besar dengan memaksimumkan payoff.

> Pilih pilihan keputusan dengan payoff tertinggi dan asumsikan bahwa keadaan sesungguhnya yang diperlukan untuk menghasilkan payoff ini akan terjadi.
> Dalam bahasa matematiknya $\max_i \max_j p_{ij} \quad \forall_{i,j}$

Untuk ilustrasi, Bengkel Bobo tidak memiliki mesin diagnostik guna mengetahui kerusakan mesin. Bobo sedang mempertimbangkan untuk membeli mesin diagnostik seharga Rp 12 juta tersebut. Jika permintaan tune-up tinggi frekuensinya, maka membeli mesin ini merupakan investasi yang baik baginya, karena bengkel akan dapat melayani lebih banyak mobil. Jika sebaliknya, sebaiknya ia tak perlu beli mesin tersebut. Payoff untuk kedua keputusan tersebut (Beli atau Tidak Beli) dapat dilihat pada tabel berikut. Tanpa menggunakan mesin, bila permintaan tune up rendah, perkiraan keuntungannya sebesar Rp 6 juta dan bila tinggi maka keuntungannya dapat mencapai Rp 8 juta.

Pilihan Keputusan	Tune-up	
	S1 : Rendah	**S2 : Tinggi**
DA_1 : Beli	-2	15
DA_2 : Tidak Beli	6	8

Jika Bobo menggunakan kriteria maximax, ia memilih payoff tertinggi, yaitu Rp 15 juta. Dengan demikian ia akan memilih untuk membeli mesin dan asumsi bahwa permintaan tune-up tinggi, sedangkan untuk tidak membeli mesin, payoff tertingginya hanya Rp 8 juta.

Dalam satu hal, aturan keputusan ini tidak realistik bahkan untuk para optimis, karena setiap orang tidak mungkin sama derajat optimisnya. Oleh karenanya, Hurwicz mengembangkan *koeffisien optimisme*, yang memungkinkan pengambil keputusan untuk memberikan bobot untuk hal yang paling mungkin dan paling tidak mungkin disukai. Misalkan koeffisien optimismenya 0.70 maka kita akan dapatkan hal berikut

Pilihan Keputusan	**Tune-up**		**Koeffisien Optimisme**		**Total Terboboti**
	Rendah	**Tinggi**	**0.30**	**0.70**	
Beli	-2	15	-0.6	10.5	9.9
Tidak Beli	6	8	1.8	5.6	7.4

Bila dilihat total nilai yang terboboti, maka kesimpulan untuk membeli mesin juga harus dipilih, bila koeffisien optimismenya 0.70.

Kriteria Maximin

Tak ada alasan tertentu untuk berpendapat bahwa pengambil keputusan perlu seseorang yang optimistik. Abraham Wald berpendapat bahwa mereka harus mengambil dari yang berpandangan paling pesimistik dan memperlakukannya sebagai lawan. Dalam memformulasikan kriteria maksimisasi payoff minimum, Wald beralasan bahwa pengambil keputusan harus mengikuti asumsi bahwa keadaan sesungguhnya berlawanan dengannya dan harus bertindak sejalan :

> Pilih keputusan yang memiliki nilai kemungkinan terbesar dari keluaran yang paling tidak dikehendaki.
> Atau dalam bahasa matematiknya $\max_i \min_j p_{ij} \quad \forall_{i,j}$

Kriteria ini jelas merupakan aturan keputusan paling konservatif. Dengan menggunakan kasus yang sama seperti diatas, maka payoff keluaran yang paling tidak dikehendaki untuk keputusan Beli adalah –Rp 2 juta, sedangkan payoff keluaran yang paling tidak diinginkan untuk Tidak Beli adalah Rp 6 juta. Dari kedua ini, maksimumnya adalah Rp 6 juta, yaitu apabila kita memutuskan untuk Tidak Beli mesin diagnostik.

Kriteria Minimax Regret

Leonard Savage memformulasikan kriteria ini. Kriteria ini juga merupakan kriteria keputusan orang-orang pesimis. Premis dalam kasus ini adalah setelah pilihan keputusan telah dipilih dan keadaan sesungguhnya terjadi, pengambil keputusan menerima payoff sesuai dengan pilihan yang dilakukannya. Jika kenyataannya bukan merupakan hal yang paling dikehendaki untuk keadaan sesungguhnya yang benar-benar terjadi, pengambil keputusan akan mengalami penyesalan (regret) untuk tidak membuat pilihan yang paling diinginkannya. Dengan dasar ini Savage mengembangkan aturan keputusan berikut

> Pilih keputusan (*Decision alternative, DA*) dimana terdapat perbedaan minimum antara payoff yang diterma dan payoff yang seharusnya dapat diterima jika keadaan sebenarnya yang terjadi telah diketahui terlebih dahulu.
>
> Atau dalam bahasa matematiknya $\min_i \max_j p_{ij}^* \quad \forall_{i,j}$

Dengan menggunakan data yang sama, kita dapat peroleh matrix regretnya $\left[p_{ij}^*\right]$ seperti berikut (dalam Rp juta)

Keputusan	Tune-up		Maksimum Regret
	Rendah	Tinggi	
Beli	8	0	8
Tidak Beli	0	7	7

Jika keadaan sesungguhnya adalah terjadi rendahnya frekuensi tune-up mobil, dan bila keputusan yang diambil adalah Tidak beli mesin diagnostik, maka nilai regretnya adalah Rp 6 juta (payoff tertinggi pada tune-up rendah) – Rp 6 juta (payoff bila keputusan yang diambil adalah Tidak Beli mesin pada kondisi Tune-up rendah) = 0. Sedangkan jika keadaan sesungguhnya adalah rendahnya frekuensi tune-up mobil, dan bila keputusan yang diambil adalah Beli mesin diagnostik, maka nilai regretnya adalah Rp 6 juta (payoff tertinggi pada tune-up rendah) – (-Rp 6 juta) (payoff bila keputusan yang diambil adalah Tidak Beli mesin pada kondisi Tune-up rendah) = Rp 8 juta.

Kriteria Laplace

Tiga aturan diatas (kriteria Maximax, Maximin, dan Minimax Regret) telah mengabaikan adanya peluang. Banyak pembuat keputusan tidak merasa nyaman dengan cara pengabaian peluang ini. Kriteria Laplace ini dapat dituliskan sebagai berikut

Jika peluang akan keadaan sesungguhnya tak diketahui, asumsikan bahwa mereka memiliki kesempatan yang sama untuk muncul atau terjadi.

Kriteria ini juga dikenal dengan prinsip alasan tak cukup dan postulat Bayes. Dengan menggunakan konsep nilai harapan, kriteria Laplace ini memilih keputusan yang memilihi nilai harapan terbesar.

E(Beli) = -Rp 2 juta x 0.50 + Rp 15 juta x 0.50 = Rp 6.5 juta

E(Tak Beli) = Rp 6 juta x 0.50 + Rp 8 juta x 0.50 = Rp 7 juta

Karenanya, diputuskan untuk tidak membeli mesin diagnostik, karena nilai harapannya lebih besar dari nilai harapan bila diputuskan untuk membeli mesin.

Prosedur Bayes

Misalkan

$A = \{a_1, a_2, ..., a_m\}$ merupakan himpunan semua kemungkinan aksi atau kegiatan yang mungkin diambil oleh pembuat keputusan.

$\theta = \{\theta_1, \theta_2, ..., \theta_n\}$ merupakan himpunan semua kondisi sesungguhnya yang dapat terjadi.

$l(a_i, \theta_j)$ = loss atau kerugian dengan memutuskan untuk melakukan aksi a_i apabila keadaan atau kondisi sesungguhnya adalah θ_j.

Dengan menggunakan notasi ini, maka kriteria minimax adalah memilih A yang memenuhi $\min_A \left\{ \max_\theta \; l(A, \theta) \right\}$. Untuk prosedur keputusan Bayes digunakan $\min_A \left\{ E_\theta \left[\; l(A, \theta) \right] \right\}$.

Prosedur Bayes ini dapat dibagi menjadi dua, yaitu tanpa data dan dengan data. Untuk prosedur Bayes tanpa data,

$\underset{\theta}{E}\left[l(a_i,\theta)\right] = \sum_{j=1}^{n} l(a_j,\theta).P(\theta=\theta_j)$ sedangkan yang dengan data

$\underset{\theta}{E}\left[l(a_i,\theta)\right] = \sum_{j=1}^{n} l(a_j,\theta).h_\theta(\theta_j \mid X=x)$ untuk i = 1,2,…,m.

Program Komputer

```
REM ******************************************************************
REM *      PROGRAM TEORI KEPUTUSAN PADA LINGKUNGAN TAK TENTU      *
REM *              MAXIMAX, MAXIMIN, MINIMAX REGRET              *
REM *                                                            *
REM *               Ir. SIGIT NUGROHO, M.Sc., Ph.D.              *
REM ******************************************************************
CLS
INPUT "BANYAKNYA KEPUTUSAN ="; BK
INPUT "BANYAKNYA KEADAAN SESUNGGUHNYA ="; KS
DIM L(BK, KS), M(BK, KS), N(BK, KS), Q(BK, KS), O(BK, KS)
DIM A(BK), B(BK), R(BK), S(KS), PO(KS), Y(BK, KS), Z(BK, KS), YY(BK), ZZ(BK)
DIM K$(BK), SN$(KS), L$(BK), REG$(BK), MAXO$(BK), LAPL$(BK)
FOR I = 1 TO BK
    PRINT "KEPUTUSAN KE "; I; : INPUT K$(I)
    L$(I) = K$(I)
    REG$(I) = K$(I)
    MAXO$(I) = K$(I)
    LAPL$(I) = K$(I)
NEXT I
FOR J = 1 TO KS
    PRINT "KEADAAN SESUNGGUHNYA KE "; J; : INPUT SN$(J)
NEXT J
FOR I = 1 TO BK
    FOR J = 1 TO KS
        INPUT L(I, J)
        M(I, J) = L(I, J)
        N(I, J) = L(I, J)
        O(I, J) = L(I, J)
        Y(I, J) = L(I, J)
        Z(I, J) = L(I, J)
    NEXT J
NEXT I
KOEF = 1
FOR J = 1 TO KS - 1
    PRINT "KOEFFISIEN OPTIMISME KE "; J; : INPUT PO(J)
    KOEF = KOEF - PO(J)
NEXT J
PO(KS) = KOEF
FOR I = 1 TO BK
    FOR J = 1 TO KS - 1
        FOR K = J + 1 TO KS
```

```
                        IF M(I, J) > M(I, K) THEN TEMP = M(I, J): M(I, K) = M(I, J): M(I, K) = TEMP
                    NEXT K
               NEXT J
               A(I) = M(I, KS)
               B(I) = M(I, 1)
          NEXT I
          FOR J = 1 TO KS
               FOR I = 1 TO BK - 1
                    FOR K = I + 1 TO BK
                        IF N(I, J) > N(K, J) THEN TEMP = N(I, J): N(I, J) = N(K, J): N(K, J) = TEMP
                    NEXT K
               NEXT I
               S(J) = N(BK, J)
          NEXT J
          FOR I = 1 TO BK - 1
               FOR K = I + 1 TO BK
                    IF A(I) > A(K) THEN TEMP = A(I): A(I) = A(K): A(K) = TEMP: TEMPO$ = K$(I): K$(I) =
K$(K): K$(K) = TEMPO$
               NEXT K
          NEXT I
          MAXIMAX = A(BK)
          FOR I = 1 TO BK - 1
               FOR K = I + 1 TO BK
                    IF B(I) > B(K) THEN TEMP = B(I): B(I) = B(K): B(K) = TEMP: TEMPU$ = L$(I): L$(I) =
L$(K): L$(K) = TEMPU$
               NEXT K
          NEXT I
          MAXIMIN = B(BK)
          FOR I = 1 TO BK
               FOR J = 1 TO KS
                    Q(I, J) = S(J) - O(I, J)
               NEXT J
          NEXT I
          FOR I = 1 TO BK
               FOR J = 1 TO KS - 1
                    FOR K = J + 1 TO KS
                        IF Q(I, J) > Q(I, K) THEN TEMP = Q(I, J): Q(I, K) = Q(I, J): Q(I, K) = TEMP
                    NEXT K
               NEXT J
               R(I) = Q(I, KS)
          NEXT I
          FOR I = 1 TO BK - 1
               FOR K = I + 1 TO BK
                    IF R(I) > R(K) THEN TEMP = R(I): R(I) = R(K): R(K) = TEMP: TEMPA$ = REG$(I):
REG$(I) = REG$(K): REG$(K) = TEMPA$
               NEXT K
          NEXT I
          MINREG = R(1)
          PRINT
          FOR I = 1 TO BK
               FOR J = 1 TO KS
                    PRINT L(I, J);
               NEXT J: :
          NEXT I
```

```
    FOR I = 1 TO BK
        FOR J = 1 TO KS
            Y(I, J) = PO(J) * L(I, J)
        NEXT J
    NEXT I
    FOR I = 1 TO BK
        FOR J = 1 TO KS - 1
            FOR K = J + 1 TO KS
                IF Y(I, J) > Y(I, K) THEN TEMP = Y(I, J): Y(I, K) = Y(I, J): Y(I, K) = TEMP
            NEXT K
        NEXT J
        YY(I) = Y(I, KS)
    NEXT I
    FOR I = 1 TO BK - 1
        FOR K = I + 1 TO BK
            IF YY(I) > YY(K) THEN TEMP = YY(I): YY(I) = YY(K): YY(K) = TEMP: TEPO$ =
MAXO$(I): MAXO$(I) = MAXO$(K): MAXO$(K) = TEPO$
        NEXT K
    NEXT I
    MAXOPT = YY(BK)
    FOR I = 1 TO BK
        FOR J = 1 TO KS
            Z(I, J) = L(I, J) / KS
        NEXT J
    NEXT I
    FOR I = 1 TO BK
        FOR J = 1 TO KS - 1
            FOR K = J + 1 TO KS
                IF Z(I, J) > Z(I, K) THEN TEMP = Z(I, J): Z(I, K) = Z(I, J): Z(I, K) = TEMP
            NEXT K
        NEXT J
        ZZ(I) = Z(I, KS)
    NEXT I
    FOR I = 1 TO BK - 1
        FOR K = I + 1 TO BK
            IF ZZ(I) > ZZ(K) THEN TEMP = ZZ(I): ZZ(I) = ZZ(K): ZZ(K) = TEMP: TEPO$ =
LAPL$(I): LAPL$(I) = LAPL$(K): LAPL$(K) = TEPO$
        NEXT K
    NEXT I
    LAPLACE = ZZ(BK)

    PRINT
    PRINT "MAXIMAX            = "; MAXIMAX; : PRINT K$(BK)
    PRINT "MAXIMIN            = "; MAXIMIN; : PRINT L$(BK)
    PRINT "MINIMAX REGRET     = "; MINREG; : PRINT REG$(1)
    PRINT "MAXIMAX KOEF OPT   = "; MAXOPT; : PRINT MAXO$(BK)
    PRINT "LAPLACE            = "; LAPLACE; : PRINT LAPL$(BK)
    END
```

```
REM ******************************************************************
REM *              PROGRAM TEORI KEPUTUSAN BERESIKO              *
REM *                                          *
REM *              Ir. SIGIT NUGROHO, M.Sc. Ph.D.           *
REM ******************************************************************
CLS
INPUT "BANYAKNYA KEGIATAN (ACTION) ="; KEG
DIM PAYOFF(KEG, KEG), SS(KEG), DA(KEG), RF(KEG), NDA$(KEG)
DIM F(KEG), NSN$(KEG)
FOR I = 1 TO KEG
     PRINT "NAMA KEGIATAN KE "; I; "ADALAH "; : INPUT NDA$(I)
NEXT I
S = 0
FOR I = 1 TO KEG
     PRINT NDA$(I); : INPUT DA(I)
     PRINT "FREKUENSI = "; : INPUT F(I)
     S = S + F(I)
NEXT I
INPUT "HARGA BELI = "; BELI
INPUT "HARGA JUAL = "; JUAL
FOR I = 1 TO KEG
     RF(I) = F(I) / S
NEXT I
FOR I = 1 TO KEG
     FOR J = 1 TO KEG
          IF DA(J) > DA(I) THEN PAYOFF(I, J) = (JUAL - BELI) * DA(I) ELSE PAYOFF(I, J) =
JUAL * DA(J) - BELI * DA(I)
          PRINT PAYOFF(I, J); :
     NEXT J:
NEXT I
FOR I = 1 TO KEG
     SS(I) = 0
     FOR J = 1 TO KEG
          PAYOFF(I, J) = RF(J) * PAYOFF(I, J)
          SS(I) = SS(I) + PAYOFF(I, J)
     NEXT J
     PRINT SS(I)
NEXT I
FOR I = 1 TO KEG - 1
     FOR K = I + 1 TO KEG
          IF SS(I) > SS(K) THEN TEMP1 = SS(I): SS(I) = SS(K): SS(K) = TEMP1: TEMP$ =
NDA$(I): NDA$(I) = NDA$(K): NDA$(K) = TEMP$
     NEXT K
NEXT I
PRINT
PRINT
PRINT "MAKSIMUM NILAI HARAPAN = "; INT(SS(KEG))
PRINT "KEPUTUSAN JANGKA PANJANG YANG DIAMBIL = "; NDA$(KEG)
END
```

```
REM ****************************************************************
REM *                    TEORI KEPUTUSAN BAYES                   *
REM *       TANPA DATA (NFLAG=0) DAN DENGAN DATA (NFLAG=1)        *
REM *                                                             *
REM *             Q(X|THETA) PELUANG BERSYARAT X|THETA            *
REM *                                                             *
REM *                Ir. SIGIT NUGROHO, M.Sc., Ph.D.              *
REM ****************************************************************
CLS
DIM A(40, 50), P(50), EL(40), HTHETA(50), ELPOST(40)
DIM THETA(50)
REM NFLAG=0 TANPA DATA
REM NFLAG=1 DENGAN DATA
INPUT "NFLAG ="; NFLAG
INPUT "BANYAKNYA KEPUTUSAN ="; M
INPUT "BANYAKNYA KONDISI SESUNGGUHNYA ="; N
REM MEMBACA LOSS MATRIX
FOR I = 1 TO M
    FOR J = 1 TO N
        PRINT "LOSS MATRIKS KEPUTUSAN KE "; I; " PADA KONDISI "; J; : INPUT A(I, J)
    NEXT J
NEXT I
REM MEMBACA PRIOR DISTRIBUSI
FOR J = 1 TO N
    PRINT "PELUANG PRIOR KONDISI SESUNGGUHNYA KE "; J; : INPUT P(J)
NEXT J
IF NFLAG = 0 GOTO 99
FOR J = 1 TO N
    PRINT "KONDISI SESUNGGUHNYA KE "; J; : INPUT THETA(J)
NEXT J
INPUT "NILAI X CONTOH ="; XSTAR
99  PRINT "MATRIKS KERUGIAN (LOSS MATRIX)"
FOR I = 1 TO M
    FOR J = 1 TO N
        PRINT A(I, J); :
    NEXT J:
NEXT I
PRINT "SEBARAN PELUANG PRIOR KEADAAN SESUNGGUHNYA"
FOR J = 1 TO N
    PRINT P(J)
NEXT J
SMALL = 10000000
FOR I = 1 TO M
    EL(I) = 0
    FOR J = 1 TO N
        EL(I) = EL(I) + P(J) * A(I, J)
    NEXT J
    IF EL(I) >= SMALL GOTO 30
    SMALL = EL(I)
    INDEX = I
30  NEXT I
IF NFLAG = 0 GOTO 101
GX = 0
FOR J = 1 TO N
```

```
        TJ = THETA(J)
        X = XSTAR
        T = TJ
        Q = EXP(-((X - T) / SQR(10)) ^ 2 / 2) / SQR(20 * 3.1416)
        GX = GX + Q * P(J)
    NEXT J
    FOR J = 1 TO N
        TJ = THETA(J)
        X = XSTAR
        T = TJ
        Q = EXP(-((X - T) / SQR(10)) ^ 2 / 2) / SQR(20 * 3.1416)
        HTHETA(J) = Q * P(J) / GX
    NEXT J
    PRINT
    PRINT "NILAI CONTOH X ADALAH "; XSTAR
    CLS
    FOR J = 1 TO N
        PRINT "KONDISI SESUNGGUHNYA KE "; J; " ADALAH"; THETA(J)
        PRINT "DENGAN PELUANG POSTERIOR "; HTHETA(J)
    NEXT J
    SMPOST = 10000000
    FOR I = 1 TO M
        ELPOST(I) = 0
        FOR J = 1 TO N
            ELPOST(I) = ELPOST(I) + HTHETA(J) * A(I, J)
        NEXT J
        IF ELPOST(I) >= SMPOST GOTO 170
        SMPOST = ELPOST(I)
        INPOST = I
170   NEXT I
101   PRINT
    FOR I = 1 TO M
        PRINT "NILAI HARAPAN KEPUTUSAN KE "; I; " ADALAH "; EL(I):
    NEXT I
    PRINT "DENGAN MENGGUNAKAN SEBARAN PRIOR KONDISI SESUNGGUHNYA,"
    PRINT "NILAI HARAPAN KERUGIAN MINIMUMNYA ADALAH "; SMALL
    PRINT "PILIH KEPUTUSAN KE "; INDEX
    PRINT
    IF NFLAG = 0 GOTO 200
    FOR I = 1 TO N
        PRINT "NILAI HARAPAN KEPUTUSAN KE "; I; " ADALAH "; ELPOST(I):
    NEXT I
    PRINT "DENGAN MENGGUNAKAN SEBARAN POSTERIOR KONDISI SESUNGGUHNYA,"
    PRINT "NILAI HARAPAN KERUGIAN MINIMUMNYA ADALAH "; SMPOST
    PRINT "PILIH KEPUTUSAN KE "; INPOST
200   END
```

Teori Antrian

Teori Antrian berkenaan dengan seluruh aspek dari situasi dimana pelanggan harus antri untuk mendapatkan suatu layanan. Situasi antrian yang umum diantaranya: Mahasiswa antri untuk mengisi KRS, Pesawat yang akan mendarat atau tinggal landas, Mesin yang akan diperbaiki, Pasien yang ingin periksa dokter, Orang yang mengantri beli bensin di pom bensin, dan nasabah yang akan melakukan transaksi di Bank.

Antrian merupakan bagian dari kehidupan manusia sehari-hari. Antrian terbentuk bilamana banyaknya yang akan dilayani melebihi kapasitas layanan yang tersedia. Dalam banyak hal, penambahan jumlah layanan dapat dipenuhi untuk mengurangi antrian atau menghindari antrian yang terus membesar; namun demikian, biaya penambahan layanan dapat menyebabkan keuntungan berada di bawah taraf yang dapat diterima. Dipihak lain, antrian yang terlalu panjang dapat mengakibatkan kehilangan penjualan ataupun pelanggan. Karenanya, permasalahan muncul karena: terlalu banyak permintaan (pelanggan terlalu lama menunggu) dan terlalu sedikit permintaan (terlalu banyak waktu luang atau mengganggur).

Masalah yang dihadapi pihak manajemen adalah bagaimana menyeimbangkan biaya yang kerkenaan dengan waktu tunggu terhadap biaya yang berkaitan dengan pencegahan atau penghindaran waktu tunggu guna memaksimumkan keuntungan. Analisis sistem antrian dapat menjawab permasalahan ini dengan kondisi yang agak umum. Namun demikian, sebelum melihat bagaimana masalah antrian dapat diselesaikan, kita akan tinjau kerangka kerja sistem antrian.

Sistem antrian mencakup pelanggan (mahasiswa, pesawat, mesin, dan lain sebagainya) yang datang dengan laju konstan atau bervariasi untuk mendapatkan layanan pada suatu fasilitas layanan. Jika pelanggan yang datang dapat memasuki fasilitas layanan, mereka dapat langsung dilayani. Jika pelanggan harus menunggu dilayani, mereka berpartisipasi atau membentuk antrian, dan akan berada dalam antrian hingga mereka dapat giliran untuk dilayani. Mereka akan dilayani dengan laju layanan yang konstan atau bervariasi dan akhirnya meninggalkan sistem. Sistem antrian mencakup baik antrian dan fasilitas layanannya.

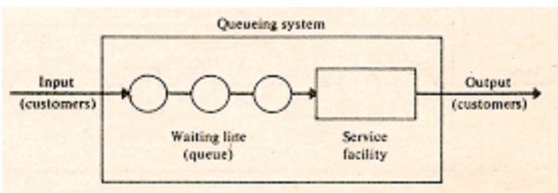
Terdapat beberapa tipe sistem antrian, akan tetapi semua itu dapat diklasifikasikan kedalam ciri-ciri berikut:

a. *Proses input atau kedatangan*. Proses ini mencakup banyaknya kedatangan per satuan waktu, jumlah antrian yang dapat dibuat, maksimum panjang antrian, dan maksimum jumlah pelanggan potensial (yang menghendaki layanan).
b. *Proses layanan*. Proses ini mencakup sebaran waktu untuk melayani seorang pelanggan, banyaknya layanan yang tersedia, dan pengaturan layanan (paralel atau seri).
c. *Disiplin antrian*. Ini merupakan bentuk dimana pelanggan membentuk antrian : yang datang duluan dilayani duluan atau FIFO (First In First Out), yang datang terakhir dilayani duluan atau LIFO (Last In First Out), pemilihan secara acak, pemilihan berdasarkan prioritas, dan lain sebagainya.

NOTASI dan ASUMSI

Dalam bagian ini, kita asumsikan bahwa:

a. Layanan mengikuti aturan siapa datang dahulu, akan dilayani dahulu pula (FIFO)
b. Datangnya pelanggan benar-benar secara acak namun dengan laju tertentu.
c. Sistem antrian berada dalam kondisi *steady-state*.



Sumber: Gillet, 1982

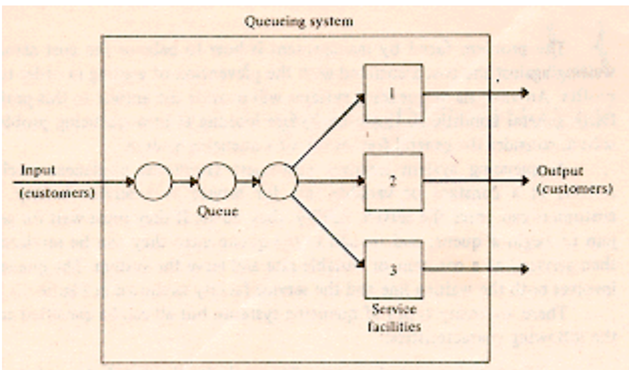
Ketiga asumsi ini valid dalam kebanyakan sistem antrian yang realistik dan akan digunakan untuk menggambarkan penggunaan teori antrian. Asumsi (a) menerangkan bahwa pelanggan yang datang duluan akan dilayani terlebih dulu tak peduli apakah pelanggan akan berada dalam antrian atau tidak. Asumsi (b) menerangkan bahwa kedatangan memiliki kesempatan yang sama kapan saja dan tidak tergantung oleh waktu yang telah berlalu sejak kedatangan terakhir. Ini setara dengan mengatakan bahwa banyaknya kedatangan per satuan waktu merupakan peubah acak (*random variable*) yang menyebar menurut sebaran Poisson. Bila X = banyaknya kedatangan per satuan waktu, maka

$$f(x) = P(X = x) = \frac{e^{-\lambda}\lambda^{x}}{x!} \qquad x = 0,1,2,... \qquad \lambda > 0$$

dan

$$E(X) = \lambda$$

dimana λ merupakan banyaknya kedatangan per satuan waktu. Hasil menarik lainnya dari (b) adalah bahwa waktu diantara dua kedatangan berturutan, T (juga sering disebut dengan *interarrival time*) memiliki sebaran eksponensial dengan parameter yang sama, λ.



Sumber: Gillet, 1982

Dengan demikian apabila T = waktu diantara dua kedatangan, maka

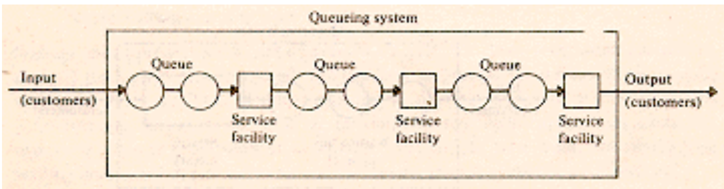
$$g(t) = \lambda e^{-\lambda t} \qquad \lambda > 0 \qquad t > 0$$

dan

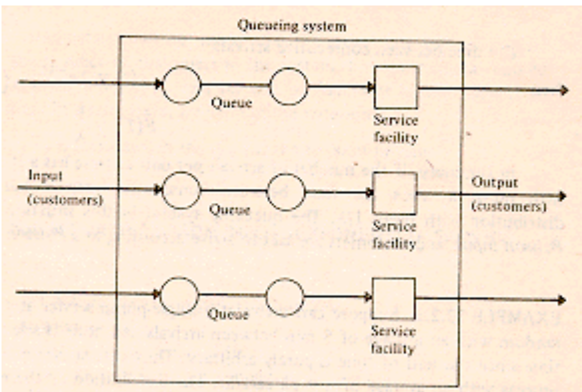
$$E(T) = \frac{1}{\lambda}$$

Jadi dapat disimpulkan, jika banyaknya kedatangan per satuan waktu memiliki sebaran Poisson dengan rata-rata λ, maka waktu diantara dua kedatangan memiliki sebaran eksponensial dengan rata-rata 1/λ. Sistem antrian seperti ini dikatakan memiliki *input Poisson*, dan pelanggan dikatakan datang mengikuti *Proses Poisson.*

Asumsi (c) berarti bahwa sistem antrian telah beroperasi cukup lama bebas dari keadaan awal sistem dan tidak tergantung dari waktu. Yang dimaksudkan disini adalah sistem telah berada pada suatu keadaan seimbang berdasarkan waktu. Sebaran jumlah kedatangan per satuan waktu dan sebaran waktu layanan tidak berubah berdasarkan waktu.



Sumber: Gillet, 1982



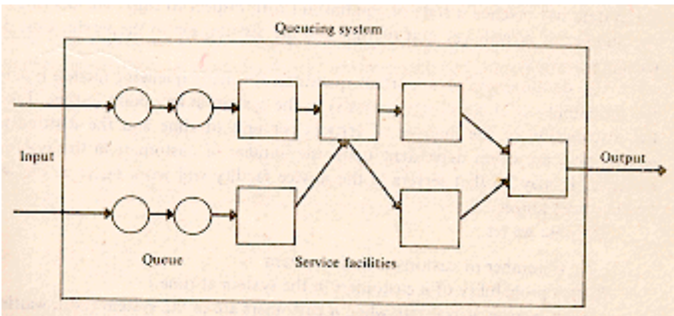
Sumber: Gillet, 1982

Misalkan

S_n	=	Banyaknya pelanggan yang berada di dalam sistem
$P_n(t)$	=	Peluang n pelanggan berada dalam sistem pada waktu t
λ_n	=	Rataan laju kedatangan apabila n pelanggan berada di dalam sistem (sedang menunggu atau dilayani)
μ_n	=	Rataan laju layanan apabila n pelanggan berada di dalam sistem

Sistem yang berada dalam kondisi *steady-state* tidak berimplikasi bahwa laju kedatangan dan laju layanan bebas akan banyaknya pelanggan dalam sistem. Untuk antrian terhingga, sumber yang terbatas, dan model layanan-ganda, λ_n dan μ_n merupakan fungsi dari jumlah pelanggan dalam sistem. Dalam kondisi *steady-state*, kita akan gunakan notasi

$P_n =$ peluang n pelanggan berada dalam sistem kapan saja.



Sumber: Gillet, 1982

Model Antrian dengan Input Poisson dan Layanan Eksponensial

Asumsi yang diperlukan untuk model antrian dengan kedatangan pelanggan menyebar menurut sebaran Poisson dan waktu layanan menyebar eksponensial adalah:

a. Kedatangan dalam sistem muncul secara acak.
b. Kedatangan membentuk antrian tunggal.
c. Disiplin yang digunakan : FIFO.
d. Keluarnya pelanggan dari sistem (setelah dilayani) juga secara acak.
e. Peluang datangnya pelanggan dalam interval waktu dari t ke t+Δt, untuk Δt yang cukup kecil, apabila sistem berada dalam keadaan S_n pada waktu t adalah $\lambda_n\Delta t$.
f. Peluang keluarnya pelanggan dalam interval waktu dari t ke t+Δt, untuk Δt yang cukup kecil, apabila sistem berada dalam keadaan S_n pada waktu t adalah $\mu_n\Delta t$.
g. Peluang lebih dari satu kedatangan dan/atau keluarnya pelanggan ke/dari sistem dalam interval waktu dari t ke t+Δt, untuk Δt yang cukup kecil, apabila sistem berada dalam keadaan S_n pada waktu t adalah adalah o(t) Δt, dimana o(t) suatu nilai yang dapat diabaikan (*negligible*).

Dengan asumsi dasar diatas kita dapat menurunkan satu gugus persamaan differensial, dimana bila diselesaikan, akan menghasilkan $P_n(t)$. Dalam *steady-state* persamaan differensial tersebut menjadi persamaan perbedaan yang dapat dituliskan sebagai berikut:

$$\lambda_{n-1}P_{n-1} + \mu_{n+1}P_{n+1} - (\lambda_n + \mu_n)P_n = 0 \quad n > 0$$

$$-\lambda_0 P_0 + \mu_1 P_1 = 0 \quad n = 0$$

Dengan menggunakan kedua persamaan diatas, kita akan dapat mencari P_1, P_2, …, P_n sebagai fungsi dari P_0, yang dapat dituliskan sebagai berikut

$$P_n = \frac{\lambda_0\lambda_1\cdots\lambda_{n-1}}{\mu_1\mu_2\cdots\mu_n}P_0 = \left(\frac{\prod_{i=0}^{n-1}\lambda_i}{\prod_{i=1}^{n}\mu_i}\right)P_0$$

Dengan menggunakan fakta bahwa total peluang untuk semua kejadian adalah 1, maka kita dapat menentukan P_0. Dari persamaan

$$\sum_{n=0}^{\infty} P_n = P_0 + \sum_{n=1}^{\infty} \left(\frac{\prod_{i=0}^{n-1} \lambda_i}{\prod_{i=1}^{n} \mu_i} \right) P_0 = 1 \,,$$

maka $P_0 = \dfrac{1}{1 + \sum_{n=1}^{\infty} \left(\frac{\prod_{i=0}^{n-1} \lambda_i}{\prod_{i=1}^{n} \mu_i} \right)}$

Model Antrian Takhingga-Sumber Takhingga-Layanan Tunggal

Masih berkaitan dengan input Pisson dan layanan Eksponensial, dimana antrian diasumsikan dapat berukuran besar sekali, demikian juga pelanggan potensial (sumber) juga sangat besar serta hanya terdapat satu layanan saja dalam sistem. Asumsi lain yang perlu ditambahkan adalah

a. Rata-rata laju kedatangan konstan, $\lambda_n = \lambda$ untuk semua n.
b. Rata-rata laju layanan konstan, $\mu_n = \mu$ untuk semua n.
c. Rata-rata laju kedatangan lebih kecil dari rata-rata laju layanan, $\lambda < \mu$.

Dengan asumsi ini, kita dapat menentukan P_0, P_1, P_2, … sebagai berikut

$$\sum_{n=0}^{\infty} P_n = \sum_{n=0}^{\infty} \left(\frac{\lambda}{\mu} \right)^n P_0 = \frac{P_0}{1 - \frac{\lambda}{\mu}} = 1$$

Dengan demikian kita peroleh

$P_0 = 1 - \dfrac{\lambda}{\mu}$ dan

$$P_n = \left(\frac{\lambda}{\mu}\right)^n \left(1 - \frac{\lambda}{\mu}\right) \text{ untuk n = 1, 2, 3, …}$$

Untuk model ini, misalkan X = banyaknya kedatangan per satuan waktu, maka

$$f(x) = \frac{e^{-\lambda}\lambda^x}{x!} \qquad \lambda > 0 \quad x = 0,1,2,3,... \quad \text{dan E(X) = } \lambda.$$

Parameter λ merupakan rata-rata laju kedatangan per satuan waktu.

Misalkan juga T = waktu untuk melayani pelanggan, maka

$$g(t) = \mu e^{-\mu t} \qquad \mu > 0 \quad t > 0 \quad \text{dan E(T) = } 1/\mu.$$

Telah kita bicarakan sebelumnya, apabila banyaknya kedatangan pelanggan per satuan waktu menyebar menurut sebaran Poisson dengan parameter λ, maka waktu antar kedatangan berurutan menyebar menurut sebaran eksponensial dengan parameter yang sama yaitu λ. Sebagaimana hal ini, jika waktu layanan menyebar menurut sebaran eksponensial dengan parameter μ, maka banyaknya pelanggan yang dilayani per satuan waktu juga menyebar menurut sebaran Poisson dengan parameter μ.

Secara umum λ disebut sebagai laju kedatangan dan μ disebut sebagai laju layanan.

Kita tertarik dalam pemodelan dari sistem ini dan mempelajari ciri penting darinya untuk menentukan jika hasilnya dapat digunakan untuk memodifikasi sistem antrian. Modifikasi dapat dilakukan, misalnya dengan menambah jumlah layanan, menyediakan ruang yang lebih untuk tempat antrian, atau membeli alat baru untuk menaikkan waktu layanan. Kebanyakan keputusan berdasarkan dari beberapa kuantiti berikut:

$$L = \frac{\lambda}{\mu - \lambda}$$ = Rata-rata banyaknya pelanggan berada dalam sistem

$$L_Q = \frac{\lambda^2}{\mu(\mu - \lambda)}$$ = Rata-rata banyaknya pelanggan dalam antrian

$L_W = \dfrac{\mu}{\mu - \lambda}$ = Rata-rata banyaknya pelanggan dalam antrian yang tak kosong

$W = \dfrac{1}{\mu - \lambda}$ = Rata-rata waktu yang diperlukan pelanggan berada dalam sistem

$W_Q = \dfrac{\lambda}{\mu(\mu - \lambda)}$ = Rata-rata waktu yang diperlukan pelanggan berada dalam antrian

$P(n > k) = \left(\dfrac{\lambda}{\mu}\right)^{k+1}$ = Peluang terdapat lebih dari k pelanggan berada di dalam sistem

$P(T > t) = e^{-\mu(1-\lambda/\mu)t}$ = Peluang waktu seorang pelanggan berada dalam sistem sedikitnya t satuan waktu

Dari informasi tersebut diperoleh hubungan bahwa $L = \lambda W$, $L_Q = \lambda W_Q$, $W = W_Q + \dfrac{1}{\mu}$ dan $L = L_Q + \dfrac{\lambda}{\mu}$.

Dengan pengecualian untuk model antrian terbatas dan sumber terbatas, hubungan ini berlaku untuk semua model.

TELADAN

Pelanggan yang datang ke sebuah pangkas rambut layanan tunggal, mengikuti proses Poisson dengan rataan waktu antar kedatangan berurutan 20 menit. Pelanggan memerlukan rata-rata 15 menit untuk di pangkas rambutnya.

a. Berapa peluang seorang pelanggan tidak harus menunggu untuk dilayani ?

   Dari informasi diatas berarti λ = 3 pelanggan/jam dan μ = 4 pelanggan/jam. Selanjutnya dengan menggunakan formula yang ada kita peroleh P_0 = peluang tak ada pelanggan di dalam sistem, sehingga seseorang dapat langsung dilayani tanpa harus menunggu = 1-3/4 = ¼ = 0,25

b. Berapakah rata-rata jumlah pelanggan yang datang ke pangkas rambut tersebut ?
   L = 3/(4-3) = 3 pelanggan

c. Berapa rata-rata waktu seorang pelanggan akan berada dalam pangkas rambut tersebut ?

   W = 1/(4-3) = 1 jam

LATIHAN

1. Penggemar bola basket datang membeli tiket untuk sebuah pertandingan dengan laju kedatangan 1 orang per menit. Diperlukan rata-rata waktu 20 detik untuk membeli tiket.
   a. Jika seorang penggemar bola basket datang dua menit sebelum pertandingan dimulai dan jika ia memerlukan waktu 1 ½ menit untuk mencapai tempat duduk sesuai dengan nomornya, dapatkah ia berharap bahwa ia dapat melihat tip-off setelah ia duduk pada tempatnya ? Tip-off adalah saat mulainya pertandingan. Asumsikan bahwa pertandingan berlangsung tepat waktu.
   b. Berapa peluang bahwa ia dapat duduk pada tempatnya sebelum pertandingan dimulai ?
   c. Kapan seorang penggemar harus datang, agar ia yakin 99% akan dapat tempat duduk sebelum pertandingan dimulai.
2. Pasien yang datang ke rumah sakit untuk layanan darurat memiliki laju kedatangan seorang per jam. Saat ini hanya ada satu layanan kasus darurat yang baru dapat dilayani. Pasien memerlukan rata-rata waktu 20 menit untuk penanganan masalah darurat ini. Berapa waktu layanan yang diperlukan agar rata-rata waktu dalam sistm tidak lebih dari 25 menit.
3. Seorang sekretaris menerima rata-rata 8 pekerjaan per jam. Kebanyakan pekerjaan yang dapat diselesaikan dalam waktu singkat, sedang sisanya sangat panjang. Namun demikian, dapat diasumsikan bahwa waktu yang diperlukan untuk

menyelesaikan pekerjaan tersebut menyebar menurut sebaran eksponensial dengan parameter 6 menit.

a. Berapa lama sebuah pekerjaan akan berada dalam sistem ?
b. Berapa peluang sebuah pekerjaan akan dapat diselesaikan kurang dari 15 menit ?
c. Berapa peluang lebih dari 5 pekerjaan berada dalam sistem ?

Model Antian Takhingga-Sumber Takhingga-Layanan Ganda

Kebanyakan sistem antrian yang realistik memiliki lebih dari satu layanan.

Kita asumsikan, dalam bagian ini, terdapat

a. Sejumlah s layanan.
b. Tiap pemberi layanan memberikan pelayanan dengan rata-rata laju yang sama dan konstan μ.
c. Rata-rata laju kedatangan juga konstan, $\lambda_n = \lambda$ untuk semua n
d. $\lambda < s\mu$

Dengan asumsi-asumsi tersebut, kita dapatkan

$$P_n = \begin{cases} \dfrac{1}{n!}\left(\dfrac{\lambda}{\mu}\right)^n P_0 & n = 0,1,2,\ldots,s-1 \\ \dfrac{1}{s!s^{n-s}}\left(\dfrac{\lambda}{\mu}\right)^n P_0 & n = s, s+1,\ldots. \end{cases}$$

yang menunjukkan besarnya peluang terdapat n pelanggan di dalam sistem pada suatu saat, dimana

$$P_0 = \frac{1}{\left\{\left[\sum_{n=0}^{s-1}\dfrac{1}{n!}\left(\dfrac{\lambda}{\mu}\right)^n\right] + \dfrac{1}{s!\left(1-\dfrac{\lambda}{\mu s}\right)}\left(\dfrac{\lambda}{\mu}\right)^s\right\}}.$$

Peluang pelanggan yang baru datang harus menunggu untuk dilayani sama dengan peluang sedikitnya ada s pelanggan di dalam sistem dapat dinyatakan sebagai berikut

$$P(n \geq s) = \sum_{n=s}^{\infty} P_n = \frac{\left(\frac{\lambda}{\mu}\right)^s P_0}{s!\left(1 - \frac{\lambda}{\mu s}\right)}$$

Properti lain adalah

$$L_Q = \frac{\left(\frac{\lambda}{\mu}\right)^{s+1} P_0}{s \cdot s!\left(1 - \frac{\lambda}{\mu s}\right)^2} \qquad L = L_Q + \frac{\lambda}{\mu} \qquad W = \frac{L}{\lambda}$$

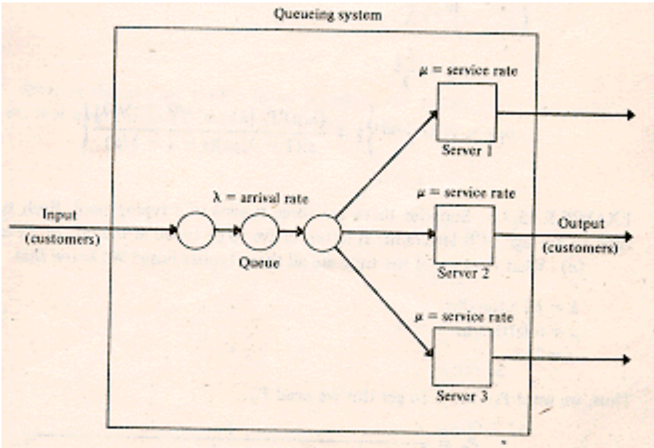
$$W_Q = \frac{L_Q}{\lambda}$$

serta

$$P(T > t) = e^{-\mu t}\left\{1 + \frac{\left(\frac{\lambda}{\mu}\right)^s P_0\left[1 - e^{-\mu t\left(s-1-\frac{\lambda}{\mu}\right)}\right]}{s!\left(1 - \frac{\lambda}{\mu s}\right)\left(s - 1 - \frac{\lambda}{\mu}\right)}\right\}$$

TELADAN

Misalkan terdapat tiga orang pengetik dalam sebuah rental komputer. Setiap pengetik dapat mengetik dengan rata-rata 6 surat per jam. Jika surat-surat yang masuk ke rental tersebut per jamnya sebanyak 15 surat.



Sumber: Gillet, 1982

a. Berapa fraksi waktu dimana semua pengetik sibuk ?

Kita peroleh nilai dari informasi yang ada bahwa λ = 15 surat per jam, μ = 6 surat per jam serta s = 3. Dengan demikian kita ingin mencari P(n≥3). Untuk mencari nilai ini, kita perlu mencari terlebih dulu nilai P_0.

$$P_0 = \frac{1}{\left[1 + \frac{\lambda}{\mu} + \frac{1}{2}\left(\frac{\lambda}{\mu}\right)^2 + \frac{1}{6}\left(\frac{\lambda}{\mu}\right)^3 \frac{1}{\left(1 - \frac{\lambda}{3\mu}\right)}\right]} = 0{,}044944$$

Sehingga P(n≥3) = 0,70225

b. Berapa rata-rata banyaknya surat yang menunggu diketik ?

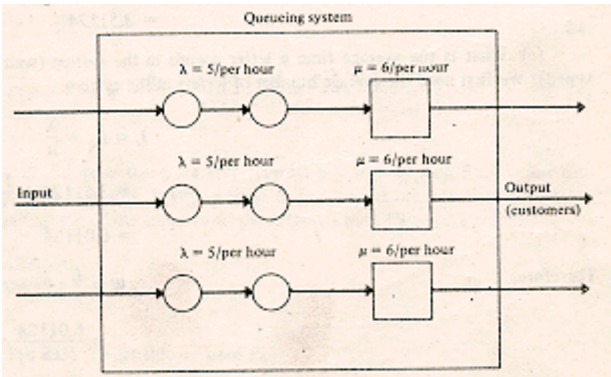
L_Q = 3,51124

c. Berapa rata-rata waktu yang diperlukan sebuah surat berada di dalam sistem ?

$$W = \frac{L}{\lambda} = \frac{1}{\lambda}\left(L_Q + \frac{\lambda}{\mu}\right) = 0{,}40075 \text{ jam} \approx 24 \text{ menit}$$

d. Berapa peluang sebuah surat akan menunggu lebih dari 20 menit berada dalam antrian dan sedang diketik oleh pengetik tersebut ?

$$P\left(T > \frac{20}{60}\right) = 0{,}46198$$



Sumber: Gillet, 1982

e. Misalkan setiap pengetik secara terpisah menerima antrian surat yang harus diketik dengan laju 5 surat per jam, masih dengan asumsi bahwa tiap pengetik dapat mengetik dengan rata-rata 6 surat per jam. Berapa rata-rata waktu yang diperlukan sebuah surat berada di dalam sistem ?

$$W = \frac{1}{\mu - \lambda} = 1 \text{ jam.}$$

Model Antrian Terhingga-Sumber Takhingga-Laynan Tunggal

Apabila ruang tunggu tempat dimana pelanggan harus menanti untuk dilayani terbatas, maka model berikut dapat digunakan. Jika pelanggan datang dan antrian telah penuh, maka pelanggan tersebut tidak bergabung dalam antrian jadi tidak masuk ke sistem. Situasi ini sering terjadi, misalnya : tempat yang terbatas untuk mobil yang datang pada sebuah layanan tunggal *drive-in bank* atau *do-it-yourself car wash* dan tempat duduk yang terbatas dalam sebuah layanan tunggal pangkas rambut.

Misalkan M adalah jumlah maksimum pelanggan yang dapat berada dalam suatu sistem pada suatu saat. Dalam kasus ini nilai λ tidak harus lebih kecil dari μ karena antrian tidak akan menjadi besar sekali.

Persamaan perbedaan pada bagian-bagian terdahulu masih valid dalam kasus ini, dengan nilai $\mu_n = \mu$ untuk n = 1,2,3,….. dan $\lambda_n = \lambda$ untuk n = 0,1,2,3,…,M-1 dan $\lambda = 0$ untuk n = M, M+1, M+2, ….

Dengan demikian, kita peroleh

$$P_0 = \frac{1}{\left[1+\sum_{n=1}^{M}\left(\frac{\lambda}{\mu}\right)^n\right]} = \frac{1}{\left[\frac{1+\left(\frac{\lambda}{\mu}\right)^{M+1}}{1-\frac{\lambda}{\mu}}\right]} = \begin{cases} \dfrac{1-\frac{\lambda}{\mu}}{1-\left(\frac{\lambda}{\mu}\right)^{M+1}} & untuk\ \lambda \neq \mu \\ \dfrac{1}{M+1} & untuk \quad \lambda = \mu \end{cases}$$

Selanjutnya, juga dapat diperoleh formula-formula

$$P_n = \begin{cases} \left(\frac{\lambda}{\mu}\right)^n P_0 \quad untuk \quad \lambda \neq \mu \quad n = 0,1,2,...,M \\ P_0 \qquad\qquad untuk \quad \lambda = \mu \end{cases}$$

$$L = \begin{cases} \dfrac{\lambda}{\mu-\lambda} - \dfrac{(M+1)\left(\frac{\lambda}{\mu}\right)^{M+1}}{1-\left(\frac{\lambda}{\mu}\right)^{M+1}} & untuk \quad \lambda \neq \mu \\ \dfrac{M}{2} & untuk \quad \lambda = \mu \end{cases}$$

$$L_Q = L - (1 - P_0)$$

Laju kedatangan pelanggan besarnya λ selagi masih ada tempat yang kosong dalam antrian; namun demikian, apabila sistem sudah penuh laju ini menjadi nol. Sebagai konsekuensinya, akan lebih bermanfaat apabila kita gunakan laju kedatangan efektif secara menyeluruh, λeff.

Dari $L_Q = L - \dfrac{\lambda_{eff}}{\mu}$ diperoleh $\lambda_{eff} = \mu\left(L - L_Q\right) = \mu\left(1 - P_0\right)$.

Kemudian, $W = \dfrac{L}{\lambda_{eff}}$ dan $W_Q = \dfrac{L_Q}{\lambda_{eff}}$

TELADAN

Sebuah pangkas rambut layanan tunggal memiliki 6 kursi yang disediakan untuk pelanggan menunggu giliran pangkas rambut. Diasumsikan bahwa pelanggan yang datang apabila semua kursi penuh, terus pergi tanpa masuk dalam sistem. Laju datangnya pelanggan adalah 3 orang per jam dan diperlukan rata-rata sekitar 15 menit untuk melayani pelanggan.

a. Berapakah peluang seorang pelanggan dapat dilayani tanpa harus menunggu ?

   Dengan M = 7, λ = 3 per jam dan μ = 4 per jam, maka
   $P_0 = 0{,}2778$

b. Berapakah rata-rata banyaknya pelanggan menunggu dalam antrian ?

   L = 2,11 sehingga
   $L_Q = 2{,}11-(1-0{,}2778) = 1{,}39$

c. Berapakah nilai laju kedatangan efektifnya ?

   $\lambda_{eff} = 4(1-0{,}2778) = 2{,}89$ per jam

d. Berapa lama seorang pelanggan berharap akan berada dalam ruang pangkas rambut ?

   W = 2,11/2,89 = 0,73 jam atau 43,8 menit

e. Berapakah fraksi pelanggan potensial yang mengurungkan niatnya masuk ke ruang pangkas rambut ?

   P(7 dalam sistem) = $P_7 \approx 0{,}037$ atau 3,7%.

Model Antrian Terhingga-Sumber Takhingga-Layanan Ganda

Model ini mendeskripsikan situasi dimana ruang tunggu terbatas dan pemberi layanan lebih dari satu.

Beberapa notasi yang akan digunakan dalam model ini dapat disarikan sebagai berikut:

s = Banyaknya layanan. Asumsikan s < M.

M =Banyaknya pelanggan maksimum yang dapat berada dalam sistem

$$\lambda_n = \begin{cases} \lambda & untuk \quad n = 0,1,2,\ldots,M-1 \\ 0 & untuk \quad n = M, M+1,\ldots. \end{cases}$$

$$\mu_n = \begin{cases} n\mu & untuk \quad n = 0,1,2,\ldots,s \\ s\mu & untuk \quad n = s+1, s+2,\ldots \end{cases}$$

$$P_0 = \frac{1}{\sum_{n=0}^{s}\left(\frac{1}{n!}\right)\left(\frac{\lambda}{\mu}\right)^n + \frac{1}{s!}\left(\frac{\lambda}{\mu}\right)^s \sum_{n=s+1}^{M}\left(\frac{\lambda}{\mu s}\right)^{n-s}}$$

$$P_n = \begin{cases} \frac{1}{n!}\left(\frac{\lambda}{\mu}\right)^n P_0 & untuk \quad n \le s \\ \frac{1}{s!\,s^{n-s}}\left(\frac{\lambda}{\mu}\right)^n P_0 & untuk \quad s < n \le M \\ 0 & untuk \quad n > M \end{cases}$$

$$L_Q = \frac{P_0\left(\frac{\lambda}{\mu}\right)^s\left(\frac{\lambda}{\mu s}\right)}{s!\left(1-\frac{\lambda}{\mu s}\right)^2}\left[1-\left(\frac{\lambda}{\mu s}\right)^{M-s} - (M-s)\left(\frac{\lambda}{\mu s}\right)^{M-s}\left(1-\frac{\lambda}{\mu s}\right)\right]$$

$$L = L_Q + s - \sum_{n=0}^{s-1}(s-n)P_n$$

$$\lambda_{eff} = \mu\left[s - \sum_{n=0}^{s-1}(s-n)P_n \right]$$

$$W_Q = \frac{L_Q}{\lambda_{eff}}$$

$$W = \frac{L}{\lambda_{eff}}$$

LATIHAN

Sebuah pangkas rambut dengan dua layanan memiliki 5 kursi untuk para pelanggan yang menunggu layanan pangkas rambut. Pelanggan akan langsung pergi tanpa harus masuk tempat pangkas rambut apabila semua kursi tunggu tersebut penuh. Rata-rata laju datangnya pelanggan 3,7634 per jam dan rata-rata pemangkas rambut hanya perlu waktu 15 menit untuk memangkas rambut pelanggan.

a. Berapakah peluang seorang pelanggan yang datang akan langsung dilayani ?
b. Berapakah rata-rata banyaknya pelanggan harus menunggu untuk dilayani ?
c. Berapa lama seorang pelanggan berharap akan berada di dalam sistem ?
d. Berapakah peluang seorang pelanggan potensial terpaksa tak dapat masuk tempat pangkas rambut ?

Model Antrian Sumber Terbatas dan Layanan Ganda

Dalam model ini, banyaknya pelanggan yang menghendaki layanan terbatas jumlahnya. Dengan demikian, laju kedatangan efektifnya merupakan fungsi dari laju layanan tiap pemberi layanan, banyaknya layanan, banyaknya pelanggan potensial, dan sebaran "waktu hingga layanan diperlukan" untuk tiap pelanggan. Kita asumsikan, tiap pelanggan datang untuk mendapatkan layanan menurut proses Poisson dengan laju λ per jam. Hal ini tidak seperti keseluruhan pelanggan potensial yang datang dengan laju yang sama seperti telah dijelaskan terdahulu. Sehingga secara umum, seluruh informasi yang terdahulu tidak valid lagi. Dikatakan secara umum, karena ada beberapa model dengan sumber terbatas yang akan diperlakukan seperti pada model sumber takterbatas atau takhingga, jika laju kedatangan total pelanggan potensial tidak terlalu dipengaruhi oleh fakta bahwa hanya ada pelanggan potensial yang terbatas. Setiap sistem antrian memiliki sumber terbatas atau terhingga, tetapi sumber tersebut secara umum cukup besar untuk berasumsi bahwa pelanggan datang dari sumber takhingga. Seberapa besar populasi potensial pelanggan dapat dijustifikasi dengan menggunakan model sumber takhingga ? Tak ada satupun jawaban yang dapat menjawab pertanyaan ini, karena merupakan fungsi dari laju kedatangan dan laju layanan.

Beberapa situasi yang dapat digunakan untuk menggambarkan permasalahan diatas:
Terdapat s teknisi yang ditugaskan untuk menangani N mesin yang secara periodik rusak.
Sebanyak N mahasiswa yang diberikan akses sendiri ke sebanyak s komputer.

Untuk kasus pertama mesin bergabung dalam sistem antrian apabila mengalami kerusakan. Mesin tersebut tak dapat digunakan dan tidak akan rusak lagi hingga diperbaiki. Dalam kasus kedua, mahasiswa sebagai pelanggan potensial.

Bila diasumsikan layanan dari tiap pemberi layanan menyebar menurut sebaran eksponensial, dan juga N = banyaknya pelanggan potensial, s = banyaknya layanan atau pemberi layanan, λ = laju kedatangan untuk tiap pelanggan, serta μ = laju layanan untuk tiap pemberi layanan, maka akan kita dapatkan statistik-statistik

$$P_0 = \frac{1}{\sum_{n=0}^{s-1} \binom{N}{n} \left(\frac{\lambda}{\mu}\right)^n + \sum_{n=s}^{N} \frac{n!}{s!s^{n-s}} \binom{N}{n} \left(\frac{\lambda}{\mu}\right)^n}$$

$$P_n = \begin{cases} P_0 \binom{N}{n} \left(\frac{\lambda}{\mu}\right)^n & untuk \quad 0 \le n \le s \\ P_0 \frac{n!}{s!s^{n-s}} \binom{N}{n} \left(\frac{\lambda}{\mu}\right)^n & untuk \quad s < n \le N \end{cases}$$

$$L_Q = \sum_{n=s+1}^{N} (n-s) P_n$$

$$\lambda_{eff} = \mu \left[s - \sum_{n=0}^{s} (s-n) P_n \right]$$

$$L = L_Q + \frac{\lambda_{eff}}{\mu}$$

$$\lambda_{eff} = \mu \left[s - \sum_{n=0}^{s} (s-n) P_n \right] = \lambda (N - L)$$

$$W_Q = \frac{L_Q}{\lambda_{eff}}$$

$$W = \frac{L}{\lambda_{eff}}$$

TELADAN

Misalkan dua teknisi memiliki tanggungjawab masing-masing untuk menjaga lima mesin yang sensitif untuk tetap bekerja atau berfungsi. Tiap mesin memiliki laju kerusakan sekali dalam satu jam. Sebagai tambahan, kedua teknisi dapat memperbaiki mesin dengan laju yang sama yaitu 4 per jam.

a. Dapatkan rata-rata banyaknya mesin yang menunggu diperbaiki.

   $P_0 = 0,314932$
   $L_Q = 0,118$

b. Dapatkan rata-rata banyaknya mesin yang tak dapat berfungsi.

   $L = 1,094$

c. Dapatkan laju kerusakan efektif dari kelima mesin.

   $\lambda_{eff} = 3,904$

Daftar Pustaka

Anderson, D.R., D.J. Sweeney and T.A. Williams. 2004. *Quantitative Methods for Business*. 9th ed. Thomson-Brooks/Cole. Singapore.

Bradley, S.P., A.C. Hax, and T.L. Magnanti. 1977. *Applied Mathematical Programming*. Addison-Wesley Publishing Company. Reading. USA.

Eppen, G.D. and F.J. Gould. 1979. *Quantitative Concepts for Management. Decision Making without Algorithms.* Prentice-Hall, Inc. Englewood Cliffs, New Jersey. USA.

Gillet, B.E. 1982. *Introduction to Operations Research. A Computer-oriented Algorithmic Approach.* Tata McGraw-Hill Publishing Company Ltd. New-Delhi. India.

Lee, S.M., L.J. Moore, and B.W. Taylor III. 1990. *Management Science.* 3rd ed. Allyn and Bacon. Boston. USA.

Taha, H.A. 2003. *Operations Research. An Introduction*. 7th ed. International Edition. Pearson Education International. Prentice Hall. Singapore.

Taylor III, B.W. 2004. *Introduction to Management Science*. 8th ed. Pearson Education International. Prentice Hall. Singapore.

Wagner, H.M. 1980. *Principles of Operations Research with Applications to Managerial Decisions.* 2nd ed. Prentice-Hall of India. New Delhi. India.

SIGIT NUGROHO, Ph.D. (University of Kentucky-USA, 1994) dilahirkan di Surakarta pada tanggal 30 Nopember 1960. Ia menyelesaikan Pendidikan Dasar dan Menengahnya di Yogyakarta. Setelah tamat **SMA Negeri III 'Padmanaba' Yogyakarta**, ia meneruskan studinya di Institut Pertanian Bogor pada tahun 1980 melalui jalur Proyek Perintis II. Lulus sebagai **Sarjana Statistika (Ir.)** tahun 1984 dari Jurusan Statistika – **Fakultas Matematika dan Ilmu Pengetahuan Alam – Institut Pertanian Bogor (FMIPA-IPB)**. Sejak awal 1986 ia bekerja sebagai staf pengajar pada Fakultas Pertanian Universitas Bengkulu (Faperta UNIB), yang selanjutnya pada tahun 2000 pindah ke Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Bengkulu. Sampai buku ini ditulis, jabatan akademiknya adalah **Lektor Kepala** dalam bidang Statistika.

Pada tahun 1987 ia melanjutkan studinya di Department of Statistics, University of Kentucky, U.S.A. dan meraih gelar **Master of Science (M.Sc.**) dalam bidang Statistika pada tahun 1989. Setelah dua tahun kembali ke Universitas Bengkulu mengasuh mata kuliah Matematika I dan II, Metode Statistika I dan II, serta Rancangan Percobaan di Faperta UNIB ia kembali meneruskan studinya pada tahun 1991 ke jenjang yang lebih tinggi di tempat yang sama (Department of Statistics, University of Kentucky, U.S.A). Dibawah bimbingan **Zakkula Govindarajulu, Ph.D.** (Minnesota, 1961), ia menyelesaikan disertasinya yang berjudul "*On the Locally Most Powerful Rank Test of the Two-way Experiment*" dan dinyatakan lulus pada tanggal 15 April 1994 dihadapan tim penguji yang terdiri dari: William S. Griffith, Ph.D., Henry Howard, Ph.D., William S. Rayens, Ph.D., Mokhtar Ali, Ph.D., Mai Zhou, Ph.D. dan mendapatkan gelar **Doctor of Philosophy (Ph.D.)** dalam bidang Statistika.

Pada tahun 1988 penulis mengikuti Kursus "*Analysis of Messy Data*" di Washington, D.C. yang diberikan langsung oleh penulis buku tentang analisis tersebut, yaitu: George M. Milliken, Ph.D. dan Dallas T. Johnson, Ph.D.

Selain sebagai staf pengajar (**Lektor Kepala** dalam bidang Statistika) Universitas Bengkulu, ia juga sebagai dosen tamu pada program doktor di Jurusan Statistika IPB (2003) dan beberapa pendidikan tinggi lainnya. Sebagai tambahan, ia juga sebagai konsultan *Data Analysis*. Pada tahun 2003-2006 penulis juga menjadi **Senior Instruktur** pada Divisi Pendidikan dan Pelatihan **PT. Bank Rakyat Indonesia (Persero) Tbk.** Sampai dengan tahun 1997 penulis juga menjadi anggota *American Statistical Association.* Berbagai kegiatan seminar dalam bidang statistika telah diikutinya baik lokal, nasional, regional, ataupun internasional.

Beberapa Publikasi Jurnal yang berhubungan dengan bidang ilmunya:

1. Uji Nonparametrik Perlakuan Acak dalam Rancangan Acak Kelompok Lengkap. *Forum Statistika dan Komputasi IPB* (1997) **2** (1), 10-14 ISSN 0853-8115.

2. Tests for Random Effects in Two-way Experiment with One Observation per Cell. ***Indian Journal of Mathematics*** vol **41** No 1 January 1999. B.N. Prasad Birth Centenary Commemoration Volume. (*with* Dr. Z. Govindarajulu, Univ of Kentucky)
3. Nonparametric tests for random effects in the balanced incomplete block design. ***Statistics & Probability Letters*** **56**, 431-437 2002. (*with* Dr. Z. Govindarajulu, Univ of Kentucky)
4. Some Notes on Nonparametric Test of Random Treatment Effects in One-way and Two-way Experiments. ***Journal of Quantitative Methods*** nol **3** no 2. 2007 (*with* Dr. Z. Govindarajulu, Univ of Kentucky)

SIGIT NUGROHO, Ph.D.
Website: http://www.stasignug.cjb.net/
Email : snsw1960@telkom.net dan sinugsta@yahoo.com

Buku ini digunakan sebagai pegangan bagi mereka yang akan mempelajari metode kuantitatif atau riset operasi. Terminologi lain riset operasi adalah manajemen sains.

Materi yang disajikan meliputi :
- Masalah Investasi
- Masalah *Stage Coach*
- Penjadwalan Produksi
- Penggantian Alat
- Masalah Penugasan
- Model Inventori Deterministik
- Masalah Pengurutan
- Masalah Transportasi
- PERT/CPM
- Teori Pengambilan Keputusan Kuantitatif

Hal lain yang perlu dicatat adalah, adanya program untuk beberapa materi yang ditulis dalam bahasa BASIC.

Mudah-mudahan buku ini dapat memberikan tambahan pengetahuan bagi yang mempelajarinya.

ISBN 978-979-9431-32-5
9 789799 431325
Metode Kuantitatif

UNIB Press
Jalan WR Supratman - Bengkulu
38371