TUGAS PENGGANTI UJIAN TENGAH SEMESTER

RESUME BUKU

OPERATION RESEARCH

(MODEL-MODEL PENGAMBILAN KEPUTUSAN)

YULI SETYO BUDI

1534021360

PROGRAM STUDI MANAJEMEN

FAKULTAS EKONOMI

UNIVERSITAS KRISNADWIPAYANA

Judul Buku : **OPERATIONS RESERACH (Model-model Pengambilan Keputusan)**

Pengarang : **TjuTju Tarliyah Dimyati – Achmad Dimyati**

Penerbit : **Sinar Baru Algesindo**

OPERATION RESEARCH (PENELITIAN OPERASIONAL)

1. Gambaran Umum Penelitian Operasional

1.1. Sejarah Singkat Perkembangan Penelitian Operasional

Pada masa perang dunia II, angkatan perang Inggris membentuk suatu tim yang terdiri atas para ilmuwan untuk mempelajari persoalan-persoalan strategi dan taktik sehubungan dengan serangan-serangan yang dilancarkan musuh terhadap negaranya. Tujuan mereka adalah untuk menentukan pengunaan sumber-sumber kemiliteran yang terbatas seperti radar dan bomber, dengan cara paling efektif. Karena tim tersebut melakukan penelitian (*research*) terhadap operasi-operasi militer, maka muncullah nama *"Military Operations Research"* (Penelitian Operasional untuk masalah Kemiliteran), yang semenjak kelahirannya telah ditandai dengan digunakannya pengetahuan ilmiah dalam usaha menentukan penggunaan sumber-sumber yang terbatas .

Keberhasilan yang diperoleh angkatan perang Inggris ini kemuian mendorong angkatan perang Amerika Serikat untuk melakukan aktivitas serupa, dengan membentuk tim yang sama yang disebut tim *Operations Research*. Mereka berhasil dalam memecahkan persoalan-persoalan logistik suplai barang-barang keperluan perang, menentukan pola-pola dasar jaringan bagi operasi alat-alat elektronik. Setelah Perang Dunia II berakhir, *Operations Research* yang lahir di Inggris ini kemudian berkembang pesat di Amerika Serikat karena keberhasilan yang dicapai oleh tim *Operations Research* dalam bidang militer ini telah menarik perhatian orang-orang industri. Sedemikian pesat perkembangannya sehingga kini *Operations Research* telah digunakan dalam hampir seluruh kegiatan, baik di perguruan tinggi, konsultanrumah sakit, perencanaan kota, maupun pada kegiatan-kegiatan bisnis.

Sebagai suatu teknik pemecahan masalah, penelitian operasional harus dipandang sebagai suatu ilmu dan seni. Aspek ilmu terletak pada penggunan teknik-teknik dan algoritma-algoritma matematik untuk memecahkan persoalan yang dihadapi, sedangkan sebagai aspek seni adalah karena keberhasilan dari solusi model matematis ini sangat tergantung pada kreativitas dan kemampuan seseorang sebagai penganalisis dalam pengambilan keputusan (*the art of balancing*).

1.2. Komponen-komponen Utama Persoalan Keputusan

Munculnya persoalan-persoalan keputusan adalah kerena seorang pengambil keputusan sering dihadapkan beberapa pilihan tindakan yang harus dilakukan. Dalam menyelesaikan persoalan yang berkaitan dengan pengambilan keputusan ini harus diidentifikasi terlebih dahulu dua komponen utamanya, yaitu :

1. *objective* (tujuan);
2.variabel-variabel.

Tujuan (*objective*) adalah hasil akhir yang hendak dicapai dengan cara memilih suatu tindakan yang paling tepat untuk sistem yang dipelajari. Dalam bidang usaha biasa, tujuan diartikan sebagai “memaksimalkan profit” atau “meminimalkan biaya yang harus dikeluarkan”. Tetapi dalam bidang-bidang lain yang sifatnya non-profit (tidak mencari keuntungan), tujuan dapat berupa “pemberian kualitas layanan kepada para pelanggan”. Manakala tujuan telah didefinisikan, maka harus dilakukan pemilihan tindakan terbaik yang dapat mencapai tujuan tersebut. Dalam hal ini, kualitas pemilihan akan sangat tergantung pada apakah si pengambil keputusan mengetahui seluruh alternatif tindakan atau tidak.

Untuk dapat menentukan tindakan tindakan yang mungkin dilakukan itu haruslah diidentifikasi variabel-variabel sistem yang dapat dikendalikan oleh pengambil keputusan, yang keberhasilannya dalam mengidentifikasikan variabel-variabel ini pun akan sangat tergantung pada bias dan pelatihan si pengambil keputusan.

1.3. Model-model dalam Penelitian Operasional

Model adalah gambaran ideal dari suatu situasi (dunia) nyata sehingga sifatnya yang kompleks dapat disederhanakan. Ada beberapa jenis model yang biasas digunakan, diantaranya adalah :

1. Model-model ikonis/fisik

   Yaitu penggambaran fisik suatu sistem, baik dalam bentuk yang ideal maupun dalam skala berbeda.
2. Model-model analog/diagramatis

Model-model ini dapat menggambarkan situasi-situasi yang dinamis dan lebih banyak digunakan daripada model-model ikonis, karena sifatnya yang dapat dijadikan analogi bagi karakteristik sesuatu yang dipelajari.

3. Model-model simbolis/matematis

   Yaitu penggambaran dunia nyata melalui simbol-simbol matematis. Model matematis yang paling banyak digunakan dalam penelitian operasional adalah model matemati yang berupa persamaan atau ketidaksamaan.

4. Model-model simulasi

   Yaitu model-model yang meniru tingkah laku sistem dengan mempelajari interaksi komponen-komponennya. Karena tidak memerlukan fungsi-fungsi matematis secara eksplisit untuk merelasikan variabel-variabel sistem, maka model-model simulasi ini dapat dapat digunakan untuk memecahkan sistem kompleks yang tidak dapat diselesaikan secara matematis. Tetapi model-model ini tidak dapat memberikan solusi yang benar-benar optimal. Yang dapat diperoleh adalah jawaban suboptimal, yaitu jawaban optimal dari alternatif-alternatif yang di tes.

5. Model-model heuristik

   Kadang-kadang formulasi matematis bersifat sangat kompleks untuk dapat memberikan suatu solusi yang pasti. Atau mungkin solusi optimal yang diperoleh, tetapi memerlukan proses perhitungan yang sangat panjang dan tidak praktis. Untuk mengatasi kasus seperti ini dapat digunakan *metode heuristik*, yaitu suatu metode pencarian yang didasarkan atas intuisi atau aturan-aturan empiris uantuk memperoleh solusi yang lebih baik daripada solusi yang telah dicapai sebelumnya.

Dalam penelitian operasional, model yang paling banyak digunakan adalah model matematis/simbolis. Disamping itu, digunakan juga model model simulasi dan heuristik.

1.4. Metodologi Penelitian Operasional

Jika penelitian operasional akan digunakan untuk memecahkan suatu persoalan di suatu organisasi, maka harus dilakukan lima langkah sebagai berikut:

Langkah 1 : memformulasikan persoalan

Definisikan persoalan lengkap dengan spesifikasi tujuan organisasi dan bagian-bagian organisasi atau sistem yang bersangkutan. Hal ini mutlak harus dipelajari sebelum persoalannya dapat dipecahkan

Langkah 2 : mengobservasi sistem

Kumpulkan data untuk mengestimasi besaran parameter yang berpengaruh terhadap persoalan yang dihadapi. Estimasi ini digunakan untuk membangun dan mengevaluasi model matematis dari persoalannya.

Langkah 3 : memformulasikan model matematis dari persoalan yang dihadapi

Dalam memformulasikan persoalan ini biasanya digunakan model analitik, yaitu model matematis yang menghasilkan persamaan. Jika pada situasi yang sangat rumit tidak diperoleh model analitik, maka perlu dikembangkan suatu model simulasi.

Langkah 4 : mengevaluasi model dan menggunakannya untuk prediksi

Tentukan apakah model matematis yang dibangun pada langkah 3 telah menggambarkan keadaan nyata secara akurat. Jika belum, buatlah model yang baru

Langkah 5 : mengimplementasikan hasil studi

Pada langkah ini kita harus menerjemahkan hasil studi atau hasil perhitungan ke dalam bahasa sehari-hari yang mudah dimengerti

1.5. Paket Programa QSB+

Ada beberapa paket programa yang dapat dipakai untuk memecahkan persoalan OR (*Operations Research*) ini, antara lain : LP, LPV2, Simplex, QPTO, LP, LPROG, QSB, QSB+, dan Storm.

Paket programa QSB+ (*Quantitative System for Business Plus*) adalah suatu sistem yang menunjang proses mengambil keputusan (*decision support system*) yang sangat mudah untuk digunakan karena sifatnya yang interaktif. Penggunaan paket programa QSB+ memberikan beberapa keuntungan, antara lain:

1. Membantu dosen atau instruktur dalam menerangkan algoritma pemecahan persoalan-persoalana OR.
2. Membantu mahasiswa dalam mempelajari OR dengan cara yang lebih menarik dan menyenangkan.
3. Membantu praktisi dalam proses pengambilan keputusan.
4. Mudah digunakan, baik pada PC (*personal computer*) maupun pada *main frame*.
5. QSB+ dirancang sedemikian rupa sehingga dapat digunakan baik oleh orang yang tidak memiliki pengalaman dalam memecahkan persoalan bisnis secara kuantitatif dengan PC maupun oleh orang yang mengenal komputer dangan baik, tetapi tidak mampu membuat program komputer.
6. Informasi dan pesan yang dapat ditampilkan sangat mudah dimengerti.

7. Dapat memperlihatkan baik solusi akhir dari persoalan maupun langkah-langkah secara rinci dari proses pemecahan persoalan.
8. Menggunakan sistem menu sehingga pemakai dapat mengenal *option* (pilihan) yang tersedia untuk memecahkan persoalan.
9. Sistem menu memungkinkan pemakai untuk memasukkan persoalan baru, membaca persoalan yang telah ada, memodifikasi persoalan yang telah ada, atau memecahkan persoalan yang ada.
10. Memungkinkan pemakai untuk memasukkan data melalui *keyboard* atau membaca data dari disket jika data telah disimpan pada disket tersebut.
11. Format dirancang sedemikian rupa sehingga dapat disesuaikan dengan hampir semua format pada buku-buku teks yang konvensional sehingga pemakai yang telah mempelajari konsep dasar OR akan dapat menggunakan QSB+ dengan mudah.
12. Setiap programa memiliki kemampuan untuk memodifikasi persoalan yang telah ada.

2. Programa Linier

2.1. Pengertian Umum

Programa linier adalah suatu cara untuk menyelesaikan persoalan pengalokasian sumber-sumber yang terbatas di antara beberapa aktivitas yang bersaing, dengan cara yang terbaik yang mungkin dilakukan. Persoalan pengalokasian ini akan muncul manakala seseorang harus memilih tingkat aktivitas-aktivitas tertentu yang bersaing dalam hal penggunaan sumber daya langka yang diperlukan untuk melaksanakan aktivitas-aktivitas tersebut. Beberapa contoh dari situasi dari uraian di atas antara lain adalah persoalan pengalokasian fasilitas produksi, persoalan pengalokasian sumber daya nasional untuk kebutuhan domestik, penjadwalan produksi, solusi permainan, dan pemilihan pola pengiriman. Satu hal yang menjadi ciri situasi di atas adalah adanya keharusan untuk mengalokasikan sumber terhadap aktivitas.

Programa linier ini menggunakan model matematis untuk menjelaskan persoalan yang dihadapinya. Sifat “linier” di sini memberi arti bahwa seluruh fungsi matematis dalam model ini merupakan fungsi linier, sedangakan kata “programa” merupakan sinonim utnuk perencanaan. Dengan demikian programa linier adalah perencanaan aktivitas-aktivitas untuk memperoleh suatu hasil yang optimal, yaitu suatu hasil yang mencapai tujuan terbaik diantar seluruh alternatif yang fisibel.

Contoh kasus :

PT Indah Gelas adalah perusahaan yang memproduksi kaca berkualitas yang digunakan untuk jendela dan pintu kaca. Perusahaan ini memiliki tiga buah pabrik yaitu, pabrik 1 yang membuat bingkai aluminium, pabrik 2 yang membuat bingkai kayu, dan pabrik 3 yang digunakan untuk memproduksi kaca dan merakit produk keseluruhan. Saat ini perusahaan mendapat pesanan berupa dua macam produk baru yang potensial, yaitu pintu kaca setinggi 8 kaki dengan bingkai aluminium (produk 1), dan jendela berukuran 4 x 6 kaki dengan bingkai produk kayu (produk 2).. karena perusahaan sedang mengalami penurunan pendapatan akibat resesi dunia, maka pimpinan perusahaan merasa perlu untuk memperbaiki/mengubah lintasan produksinya dengan cara menghentikan pembuatan beberapa produk yang tidak menguntungkan sehingga kapasitas produksi dapat digunakan untuk membuat salah satu atau kedua p[roduk baru yang potensial tersebut. Kepala bagian pemasaran telah menyimpulkan bahwa peusahaan harus dapat menjual kedua produk itu sebanyak-banyaknya, yaitu sejumlah yang dapat dibuat dengan kapasitas yang ada. Akan tetapi, karena kedua produk tersebut akan bersaing untuk menggunakan kapasitas produksi yang sama di pabrik 3, maka persoalannya adalah: berapa banyakkah masing-masing produk dapat dibuat sehingga diperoleh keuntungan terbaik?

Untuk menyelesaikan persoalan tersebut, terlebih dahulu harus dicari data mengenai:

1. Persentase kapasitas produksi masing-masing pabrik yang dapat digunakan untuk kedua macam produk tersebut
2. Persentase kapasitas yang diperlukan masing-masing produk untuk setiap unit yang diproduksi per menit.
3. Keuntungan per unit untuk masing-masing produk.

Produk / Pabrik	Kapasitas yang digunakan perunit Ukuran produksi		Kapasitas yang dapat digunakan
	1	2	
1	1	0	4
2	0	2	12
3	3	2	18
Keuntungan per unit	$ 3	$ 5	

Karena kapasitas yang telah digunakan oleh suatu produk di pabrik 3 menyebabkan produk lain tidak dapat menggunakannya, maka persoalan tesrbut dikenal sebagai persoalan programa linier dengan tipe “campuran produk” atau *product mix*.

Untuk memformulasikan model matematis dari persoalan tersebut, kita tentukan x_1 dan x_2 sebagai jumlah unit dari produk 1 dan produk 2 yang diproduksi per menit. Dengan demikian x_1 dan x_2 menjadi variabel keputusan dari model ini, dan tujuannya adalah memilih harga-harga x_1 dan x_2 sehingga diperoleh nilai maksimum dari:

$$z = 3x_1 + 5x_2$$

berdasarkan pembatas yang ada, yaitu kapasitas pabrik yang dapat digunakan.

Tabel tersebut memberikan implikasi bahwa setiap unit produk 1 yang diproduksi per menit akan menggunakan 1 persen dari kapasitas pabrik 1, padahal kapasitas yang digunakan hanya 4 persen. Pembatas ini dinyatakan secara matematis dengan ketidaksamaan $x_1 \leq 4$. Dengan cara yang sama, pabrik 2 memiliki pembatas $2x_2 \leq 12$. Persentase kapasitas pabrik 3 digunakan dengan cara memilih x_1 dan x_2 sebagai produk-produk baru tersebut sehingga ukuran produksinya adalah $3x_1 + 2x_2$. Karena itu secara matematis pembatas dari pabrik 3 ini adalah $3x_1 + 2x_2 \leq 18$. Karena ukuran produksi ini tidak mungkin berharga negatif, maka variabel-variabel keputusan ini harus dibatasi sehingga berharga nonnegatif dengan $x_1 \geq 0$ dan $x_2 \geq 0$.

Sebagai kesimpulan, persoalan di atas dapat dinyatakan secara matematis yaitu sebagai berikut:

$$\begin{aligned} z &= 3x_1 + 5x_2 \\ x_1 & \qquad\quad \leq 4 \\ & \qquad 2x_2 \leq 12 \\ 3x_1 & + 2x_2 \leq 18 \\ \\ x_1 & \geq 0,\ x_2 \geq 0 \end{aligned}$$

Dari ilustrasi tersebut ditarik kesimpulan mengenai pengertian persoalan programa linier sebagai berikut: Persoalan programa linier adalah suatu persoalan optimasi dimana kita melakukan hal-hal berikut ini :

1. Kita berusaha memaksimalkan atau meminimalkan suatu fungsi linier dari variabel-variabel keputusan yang disebut fungsi tujuan.
2. Harga/besaran dari variable-variabel keputusan itu harus memenuhi stu set pembatas. Setiap pembatas harus merupakan persamaan linier atau ketidaksamaan linier.
3. Suatu pembatas tanda dikaitkan dengan setiap variabel. Untuk setiap variabel x_i, pembatasan tanda akan menunjukkan apakah x_i harus nonnegatif ($x_i \geq 0$) atau x_i tidak terbatas tanda.

2.2. Model Programa Linier

Pada contoh persoalan PT Indah Gelas terdapat tiga buah sumber terbatas (yaitu kapasitas produksi pada ketiga pabrik) yang harus dialokasikan di antara dua aktivitas yang bersaing (yaitu dua macam produk baru yang dipesan). Sekarang, bagaimana jika ada sejumlah (katakanlah m buah) sumber yang terbatas yang harus dialokasikan di antara sejumlah (katakanlah n buah) aktivitas yang sedang bersaing?

Istilah lebih umum dari model programa linier ini adalah sebagai berikut:

a. Fungsi yang dimaksimalkan, yaitu $c_1 x_1 + c_2 x_2 + \dots + c_n x_n$, disebut sebagai fungsi tujuan.
b. Pembatas-pembatas atau konstrain.
c. Sebanyak m buah konstrain pertama sering disebut sebagai konstrain fungsional atau pembatas teknologis.
d. Pembatas $x_j \geq 0$ disebut sebagai konstrain nonnegatif.
e. Variabel x_j adalah variabel keputusan.
f. Konstanta-konstanta a_{ij}, b_i, c_j adalah parameter-parameter model.

Selain model programa linier dengan bentuk seperti yang telah diformulasikan tersebut, ada pula model programa linier dengan bentuk yang agak lain, seperti:

1. Fungsi tujuan bukan memaksimalkan, melainkan meminimalkan.

   Contoh : meminimalkan $z = c_1 x_1 + c_2 x_2 + \dots + c_n x_n$

2. Beberapa konstrain fungsionalnya memiliki ketidaksamaan dalam bentuk lebih besar atau sama dengan.

   Contoh : $a_{i1} x_1 + a_{i2} x_2 + \dots + a_{in} x_n \geq b_i$

   untuk beberapa harga i

3. Beberapa konstrain fungsionalnya memiliki bentuk persamaan

   Contoh : $a_{i1} x_1 + a_{i2} x_2 + \ldots + a_{in} x_n = b_i$

   untuk beberapa harga i

4. Menghilangkan konstrain nonnegatif untuk beberapa variabel keputusan.

   Contoh : x_j tidak terbatas dalam tanda, untuk beberapa harga j

2.3. Asumsi dalam Model Programa Linier

Dalam menggunakan model programa linier, diperlukan beberpa asumsi sebagai berikut :

1. Asumsi sebanding (*proportionality*)
   a. Kontribusi setiap variabel keputusan terhadap fungsi tujuan adalah sebanding dengan nilai variabel keputusan
   b. Kontribusi suatu variabel keputusan terhadap ruas kiridari setiap pembatas juga sebanding dengan nilai variabel keputusan itu
2. Asumsi penambahan (*additivity*)
   a. Kontribusi setiap variabel keputusan terhadap fungsi tujuan bersifat tidak tergantung pada nilai dari avariabel keputusan yang lain.
   b. Kontribusi suatu variabel keputusan terhadap ruas kiri dari setiap pembatas bersifat tidak tergantung pada nilai dari variabel keputusan yang lain.
3. Asumsi pembagian (*divisibility*)

   Dalam persoalan programa linier, variabel keputusan boleh diasumsikan berupa bilangan pecahan.
4. Asumsi kepastian (*certainty*)
5. Setiap parameter, yaitu koefisien fungsi tujuan, ruas kanan, dan koefisien teknologis diasumsikan dapat diketahui secara pasti.

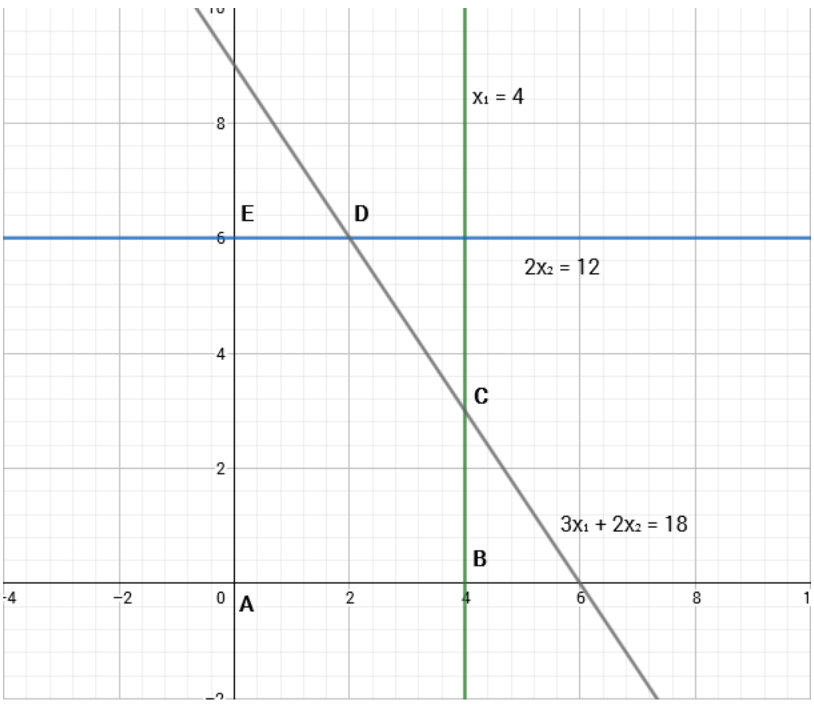
3. Teknik Pemecahan Model Programa Linier

3.1. Solusi Grafis

Untuk mencari solusi suatu persoalan programa linier dengan cara grafis, berikut ini dikemukakan dua buah contoh yaitu persoalan maksimasi dan minimasi.

1. Solusi grafis untuk persoalan maksimasi

   Perhatikan kembali contoh soal PT Indah Gelas. Pada prosedur grafis ini kita harus membuat grafik berdimensi 2 dengan x_1, dan x_2 sebagai sumbu-sumbunya. Langkah pertama adalah mengidentifikasi harga-harga (x_1, x_2) yang memenuhi pembatas-pembatas yang ada dengan cara menggambarkan garis-garis yang harus membatasi daerah harga-harga yang diperbolehkan. Ingat bahwa pembatas nonnegatif $x_1 \geq 0$ dan $x_2 \geq 0$ akan menyebabkan (x_1, x_2) harus berada pada sisi positif dari sumbu-sumbunya (pada kuadran I). Setelah itu, perhatikan bahwa pembatas $x1 \leq 4$ berarti bahwa (x_1, x_2) tidak boleh berada di sebelah kanan garis $x_1 = 4$. Demikian pula dalam menggambarkan pembatas $2x_2 \leq 12$ berarti bahwa (x_1, x_2) tidak boleh berada di atas garis $2x_2 = 12$ atau ($x_2 = 6$). Untuk menggambarkan pembatas terakhir $3x_1 + 2x_2 \leq 18$ adalah dengan cara menentukan titik (x_1, x_2) yang memenuhi garis $3x_1 + 2x_2 = 18$. Perlu diingat bahwa titik-titik yang memenuhi pembatas $3x_1 + 2x_2 \leq 18$ adalah titik-titik di bawah garis $3x_1 + 2x_2 = 18$. Setelah seluruh pembatas digambarkan maka akan diperoleh daerah berlakunya harga-harga (x_1, x_2) seperti pada grafik berikut.



Arah panah pada setiap garis pembatas menunjukkan arah berlakunya harga (x_1, x_2) pada masing-masing pembatas. Karena kita harus mendapatkan harga (x_1, x_2) yang memenuhi seluruh pembatas yang ada, maka akhirnya kita hanya perlu memperhatikan bidang ABCDEF, yaitu suatu bidang yang dibatasi oleh garis-garis pembatas yang memenuhi syarat (fisibel) sehingga bidang ABCDE ini disebut sebagai daerah fisibel. Langkah terakhir yang harus kita lakukan adalah menentukan suatu titik pada daerah fisibel yang dapat memaksimumkan harga $z = 3x_1 + 5x_2$. Caranya adalah dengan menggambarkan sebuah garis z yang telah mempunyai koefisien arah :

$$Tg\ \alpha = \frac{x2}{x1} = -\frac{3}{5}$$ (berharga negatif karena α berada pada kuadran II)

α adalah sudut antara garis z dengan sumbu x1. Setelah itu, buatlah garis lain yang sejajar dengan garis z sedemikian sehingga garis tersebut dapat melalui titik sudut terjauh pada bidang ABCDE. Titik sudut terjauh itu dinamakan titik optimal karena ia akan memberikan harga (x_1, x_2) yang memaksimalkan $z = 3x_1 + 5x_2$.

Harga (x_1, x_2) pada titik optimal diperoleh dengan cara menentukan titik potong garis ED (pembatas ke-2) dengan garis CD (pembatas ke-3) sebagai berikut :

$$\begin{array}{rcrcl} & & 2x_2 & = & 12 \\ \underline{3x_1} & \underline{+} & \underline{2x_2} & \underline{=} & \underline{18} \; - \\ -\,3x_1 & & & = & -6 \end{array}$$

sehingga diperoleh harga-harga $x_1 = 2$ dan $x_2 = 6$.

Dengan demikian, solusi optimal dari persoalan PT Indah Gelas ini adalah bahwa perusahaan harus membuat produk 1 sebanyak 2 unit per menit, dan produk 2 sebanyak 6 unit per menit, dengan keuntungan yang dapat diperoleh sebesar z = 3(2) + 5(6) atau sebesar Rp 36 per menit.

2. Solusi grafis untuk persoalan minimasi

Contoh soal :

PT Auto Indah memproduksi dua jenis mobil, yaitu mobil sedan dan truk. Untuk meraih konsumen berpenghasilan tinggi, perusahaan ini memutuskan untuk melakukan promosi dalam dua macam acara TV, yaitu pada acara hiburan dan acara olahraga. Promosi pada acara hiburan akan disaksikan oleh 7 juta pemirsa wanita dan 2 juta pemirsa pria. Promosi pada acara olahraga akan disaksikan oleh 2 juta pemirsa wanita dan 12 juta pemirsa pria. Biaya promosi pada acara hiburan adalah 5 juta rupiah/menit, sedangkan pada acara olahraga biayanya adalah 10 juta rupiah/menit. Jika perusahaan menginginkan promosinya disaksikan sedikitnya 28 juta pemirsa wanit dan sedikitnya oleh 24 juta pemirsa pria, bagaimanakah strategi promosi itu sebaiknya?

Jawaban :

Variabel keputusan :

x_1 = lamanya promosi dalam acara hiburan

x_2 = lamanya promosi dalam acara olahraga

formulasi persoalan :

minimalkan $z = 5x_1 + 10x_2$

berdasarkan

$$7x_1 + 2x_2 \geq 28$$

$$2x_1 + 12x_2 \geq 24$$

$$x_1 \geq 0, \; x_2 \geq 0$$

3.2. Bentuk Standar Model Programa Linier

Telah diterangkan pada bab sebelumnya bahwa model programa linier ini dapat memiliki pembatas-pembatas yang bertanda ≤, =, maupun ≥. Demikian juga variabel-variabelnya yang dapat berupa variabel nonnegatif, dapat pulavariabel-variabel yang tidak terbatas dalam tanda (*unrestricted in sign*).

Di dalam menyelesaikan persoalan programa linier dengan menggunakan metode simpleks, bentuk dasar yag digunakan haruslah bentuk standar, yaitu bentuk formulasi yang memiliki sifat-sifat sebagai berikut :

- Seluruh pembatas harus berbentuk persamaan (bertanda =) dengan ruas kanan yang nonnegatif
- Seluruh variabel harus merupakan variabel nonnegatif
- Fungsi tujuannya dapat berupa maksimasi atau minimasi

Untuk mengubah suatu bentuk formulasi yang belum standar ke dalam bentuk standar ini dapat dilakukan dengan cara sebagai berikut :

1. Pembatas (*constraint*)
   a. Pembatas yang bertanda ≤ atau ≥ dapat dijadikan suatu persamaan (bertanda =) dengan menambahkan atau mengurangi dengan suatu variabel *slack* pada rusa kiri pembatas itu.

      Contoh 1:

      $$x_1 + 2x_2 \leq 6$$

      kita tambahkan *slack* $S_1 \geq 0$ pada ruas kiri sehingga diperoleh persamaan:

      $$x_1 + 2x_2 + S_1 = 6, \; S_1 \geq 0$$

      Jika pembatas di atas menyatakan batas penggunaan suatu sumber, maka S_1 akan menyatakan banyaknya sumber yang tidak terpakai.

      Contoh 2:

      $$3x_1 + 2x_2 - 3x_3 \geq 5$$

karena ruas kirinya tidak lebih kecil dari ruas kanan, maka harus dikurangkan variabel $S_2 \geq 0$ pada ruas kiri sehingga diperoleh persamaan :

$$3x_1 + 2x_2 - 3x_3 - S_2 = 5,\ S_2 \geq 0$$

b. Ruas kanan dari suatu persamaan dapat dijadikan bidang nonnegatif dengan cara mengalikan kedua ruas dengan -1.

   Contoh:

   $2x_1 - 3x_2 - 7x_3 = -5$, secara matematis adalah sama dengan $-2x_1 + 3x_2 + 7x_3 = 5$

c. Arah ketidaksamaan dapat berubah apabila kedua ruas dikalikan dengan -1.

   Contoh:

   $2 < 4$ adalah sama dengan $-2 > -4$

   $2x_1 - x_2 \leq -5$ adalah sama dengan $-2x_1 + x_2 \geq 5$

d. Pembatas dengan ketidaksamaan yang ruas kirinya berada dalam tanda mutlak dapat diubah menjadi dua ketidaksamaan.

   Contoh 1:

   untuk $b \geq 0$, $|a_1x_1 + a_2x_2| \leq b$ adalah sama dengan $a_1x_1 + a_2x_2 \leq b$ dan $a_1x_1 + a_2x_2 \geq -b$.

   Contoh 2:

   untuk $q \geq 0$, $|p_1x_1 + p_2x_2| \geq q$ adalah sama dengan $p_1x_1 + p_2x_2 \geq q$ atau $p_1x_1 + p_2x_2 \leq -q$.

2. Variabel

   Suatu variabel y1 yang tidak terbatas dalam tanda dapat dinyatakan sebagai dua variabel nonnegatif dengan menggunakan subtitusi :

   $$y_i = y_i' - y_i'' \text{ dimana } y_i' \text{ dan } y_i'' \geq 0$$

   subtitusi seperti ini harus dilakukan pada seluruh pembatas dan fungsi tujuannya.

3. Fungsi tujuan

   Walaupun model standar programa linier ini dapat berupa maksimasi atau minimasi, kadang-kadang diperlukan perubahan dari satu bentuk ke bentuk lainnya. Dalam hal ini, maksimasi dari suatu fungsi adalah sama dengan minimasi dari negatif fungsi yang sama.

   Contoh:

   Maksimalkan $z = 5x_1 + 2x_2 + 3x_3$

   secara matematis adalah sama dengan :

   minimalkan $(-z) = -5x_1 - 2x_2 - 3x_3$

3.3. Metode Simpleks

Metode ini merupakan prosedur aljabar yang bersifat iteratif, yang bergerak selangkah demi selangkah, dimulai dari suatu titik ekstrem pada daerah fisibel (ruang solusi) menuju ke titik ekstrem yang optimal.

Untuk dapat lebih memahami uraian selanjutnya, berikut ini diberikan pengertian dari beberapa terminologi dasar yang banyak digunakan dalam membicarakan metode simpleks. Untuk itu perhatikan kembali model programa linier berikut ini :

Maks. Atau min. : $z = c_1x_1 + c_2x_2 + \ldots + c_nx_n$

Berdasarkan :

$$
\begin{array}{l}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\
. \\
. \\
. \\
a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m \\
x_i \geq 0 \ (i = 1, 2, \ldots, n)
\end{array}
$$

Jika didefinisikan :

$$
A = \begin{array}{l} a_{11}\ a_{12} \ldots a_{1n} \\ a_{11}\ a_{12} \ldots a_{1n} \\ . \\ . \\ a_{m1}\ a_{m2} \ldots a_{mn} \end{array} \; ; x = \begin{array}{c} x_1 \\ x_2 \\ \\ \\ x_n \end{array} \; ; b = \begin{array}{c} b_1 \\ b_1 \\ \\ \\ b_m \end{array}
$$

maka pembatas dari model tersebut dapat dituliskan ke dalam bentuk sistem persamaan $AX = b$

Perhatikan suatu sistem $AX = b$ dari m persamaan linier dalam n variabel ($n > m$)

Definisi :

1. Solusi basis. Solusi basis untuk $AX = b$ adalah solusi dimana terdapat sebanyak-banyaknya m variabel berharga bukan nol. Untuk mendapatkan solusi basis dari $AX = b$ maka sebanyak $(n - m)$ variabel harus dinolkan. Variabel-variabel yang dinolkan

disebut variabel non basis (NBV). Selanjutnya, dapatkan harga dari n – (n – m) = m variabel lainnya yang memenuhi AX = b, yang disebut variabel basis (BV).

2. Solusi basis fisibel. Jika seluruh variabel pada suatu solusi basis berharga nonnegatif, maka solusi itu disebut solusi basis fisibel (BFS).
3. Solusi fisibel titik ekstrem. Yang dimaksud dengan solusi fisibel titik ekstrem atau titik sudut adalah solusi fisibel yang tidak terletak pada suatu segmen garis yang menghubungkan dua solusi fisibel lainnya.

Ada tiga sifat titik pokok ekstrem ini, yaitu

Sifat 1a : jika hanya ada satu solusi optimal maka pasti ada satu titik ekstrem

Sifat 1b : jika solusi optimalnya banyak maka paling sedikit ada dua titik ekstrem yang berdekatan.

Sifat 2 : hanya ada sejumlah terbatas titik ekstrem pada setiap persoalan

Sifat 3 : jika suatu titik ekstrem memberikan harga z yang lebih baik dari yang lainnya maka pasti solusi itu merupakan solusi optimal.

Sifat 3 ini menjadi dasar dari metode simpleks yang prosedurnya meliputi tiga langkah sebagai berikut:

1. Langkah inisialisasi : mulai dari suatu titik ektrem (0,0).
2. Langkah iterasi : bergerak menuju titik ekstrem berdekatan yang lebih baik. Langkah ini diulangi sebanyak yang diperlukan.
3. Aturan penghentian : memberhentikan langkah ke-2 apabila telah sampai pada titik ekstrem yang terbaik (titik optimal)

Sebagai ilustrasi, kita lihat kembali persoalan PT Indah Gelas. Algoritma simpleks dimulai dari titik A (0, 0) yang biasa di sebut sebagai solusi awal (*starting solution*). Kemudian bergerak ke titik sudut yang berdekatan, bisa ke B atau ke E. Dalam hal ini, pemilihan (B atau E) akan tergantung pada koefisien fungsi tujuan. Karena koefisien x_2 lebih besar daripada x_1, dan fungsi tujuannya maksimasi, maka solusi akan bergerak searah dengan peningkatan x_2 hingga mencapai titik ekstrem E. Pada titik B proses yang sama diulangi untuk menguji apakah masih ada titik ekstrem lain yang dapat memperbaiki nilai fungsi tujuan. Karena titik ekstrem D (2,6) memberikan nilai fungsi tujuan yang lebih baik daripada titik E (0,6) dan titik C (4,3), maka iterasi berhenti, dengan titik D (2,6) sebagai titik optimal.

Dengan demikian ada dua aturan yang berlaku dalam memilih titik ekstrem yang berikut setelah mencapai suatu titik ekstrem tertentu, yaitu:

- Titik ekstrem yang berikutnya ini harus merupakan titik ekstrem yang berdekatan dengan titik ektrem yang sudah dicapai. Sebagai contoh, dri titik A tidak bisa bergerak langsung ke titik D atau C karena mereka tidak berdekatan
- Solusi ini tidak akan pernah kembali ke titik ekstrem yang telah dicapai sebelumnya. Misalnya, dari titik E tidak akan kembali lagi ke titik A.

Sebagai ringkasan dari ide metode simpleks ini adalah bahwa metode ini selalu dimulai pada suatu titik sudut fisibel, dan selalu bergerak melalui titik sudut fisibel berdekatan, menguji masing-masing titik mengenai optimalitasnya sebelum bergerak pada titik lainnya. Pada persoalan PT Indah Gelas diperlukan iterasi untuk mencapai solusi optima, yaitu A, E, dan D. Untuk mengekspresikan ide ini dalam konteks metode simpleks, diperlukan suatu korespondensi antara metode grafis dan metode simpleks mengenairuang solusi dan titik-titik sudut (titik-titik ekstrem) sebagai berikut:

Definisi geometris (metode grafis)	Definisi aljabar (metode simpleks)
Ruang Solusi	Pembatas-pembatas dalam bentuk standar
Titik-titik sudut/ekstrem	Solusi-solusi basis dari bentuk standar

Maka, sebagai ilustrasi dari representasi ruang solusi secara aljabar ini, kita lihat lagi persoalan PT Indah Gelas. Bentuk standar model persoalan ini adalah:

Maksimalkan : $z = 3x_1 + 5x_2 + OS_1 + OS_2 + OS_3$

Berdasarkan pembatas:

$$x_1 + S_1 = 4$$
$$2x_2 + S_2 = 12$$
$$3x_2 + 2x_2 + S_3 = 18$$
$$X_1, x_2, S_1, S_2, S_3 \geq 0$$

Setiap titik pada ruang solusi diatas dapat direpresentasikan sebagai variabel-variabel x1, x2, S1, S2, dan S3 dari bentuk standarnya. Jika S1 = 0 maka garis nya adalah x1 = 4 yang merepresentasikan sudut BC. Jika S1 > 0 maka titik-titik fisibel bergerak ke arah produk 1 dari ruang solusi tersebut.

Dari gambar tersebut kita juga dapat mengidentifikasi titik-titik ekstrem secara aljabar sebagai berikut:

Titik ekstrem	Variabel-variabel nol	Variabel-variabel bukan nol
A	x1, x2	S1, S2, S3
B	x2, S1	x1, S2, S3
C	S1, S3	x1, x2, S2
D	S3, S2	x1, x2, S1
E	S2, x1	x2, S1, S3

Dari uraian diatas ada dua hal yang dapat disimpulkan, yaitu:

- Karena bentuk standar persoalan ini memiliki 3 persamaan pembatas dengan 5 titik, maka setiap titik ekstrem pasti memiliki sebanyak 2 (= 5 - 3) variabel berharga nol.
- Titik-titik ekstrem yang berdekatan, berebda hanya pada 1 variabel.

Kesimpulan pertama menunjukkan bahwa kita dapat mengidentifikasi titik-titik ekstrem suatu ruang solusi secara aljabar, dengan cara mengenolkan sebanyak (n - m) variabel. Banyaknya persamaan pembatas fungsional adalah m, sedangkan banyknya variabel adalah (m ≤ n) adalah n.

Secara matematis solusi yang diperoleh dari pengenolan (n - m) variabel itu kemudian disebut sebagai solusi basis (*basic solution*). Jika suatu solusi basis dapat memenuhi pembatas-pembatas nonnegatif, maka solusi ini disebut sebagai solusi basis fisibel (*feasible basic solution*). Variabel-variabel yang dinolkan disebut sebagai variabel-variabel basis (*basic variables*). Jumlah iterasi maksimum dalam metode simpleks adalah sama dengan jumlah maksimum solusi basis dalam bentuk standar. Dengan demikian jumla iterasi simpleks ini tidak akan lebih dari:

$$C_m^n = n\,!\,/\,[\,(n - m)\,!\;m\,!\,]$$

Dari kesimpulan yang kedua, titik ekstrem yang berdekatan hanya berbeda pada satu variabel, kita dapat menetapkan titik ekstrem berikutnya dengan mengganti variabel nonbasis (variabel yang telah dinolkan) yang telah dicapai dengan variabel basis yang telah dicapai. Sebagai contoh pada persoalan PT Indah Gelas, misalkan bahwa kita sedang berada di titik A dan akan bergerak ke titik E. Untuk dapat mencapai titik E ini kita naikkan harga variabel nonbasis x2 dari nilai semula (yaitu 0) hingga mencapai titik E. Pada titik E, variabel S2 (yang sebelumnya merupakan variabel basis di titik A) secara otomatis menjadi nol, artinya menjadi variabel nonbasis.

3.5. Kasus Khusus dalam Penggunaan Algoritma Simpleks

1. Degenerasi

   Kasus ini terjadi apabila satu atau lebih variabel basis berharga nol (b = 0) sehingga iterasi yang dilakukan selanjutnya bisa menjadi suatu *loop* yang akan kembali pada bentuk sebelumnya. Kejadian ini disebut cycling atau circling.

2. Solusi Optimal Banyak

   Seperti telah dibahas dalam subbab sebelumnya, suatu persoalan dapat memiliki lebih dari satu solusi optimal. Kasus ini terjadi apabila fungsi tujuan paralel dengan fungsi pembatas, di mana paling sedikit salah satu dari variabel nonbasis (pada persamaan z pada iterasi terakhir) mempunyai koefisien berharga nol. Akibatnya, walaupun variabel tersebut dinaikkan harganya (dijadikan variabel basis), harga z tetap tidak berubah. Karena itu, solusi-solusi optimal yang lain biasanya dapat diidentifikasi dengan cara menunjukkan iterasi-iterasi tambahan pada metode simpleksnya, di mana variabel-variabel nonbasis yang berkoefisien nol itu selalu dipilih untuk menjadi *entering variable*.

3. Solusi tak Terbatas

   Kasus ini terjadi apabilaruang solusi tidak terbatas sehingga nilai fungsi tujuan dapat meningkat (untuk maksimasi) atau menurun (minimasi) secara tidak terbatas. Apabila persoalannya dapat diselesaikan secara grafis (berdimensi dua), maka kasus ini akan mudah terdeteksi. Akan tetapi, jika persoalan yang dihadapi berdimensi tiga atau lebih, maka untuk mendeteksi apakah solusinya terbatas atau tidak, maka dilakukan dengan cara:

   a. Perhatikan koefisien-koefisien pembatas dari variabel nonbasis pad suatu iterasi. Jika koefisien-koefisien tersebut berharga negatif atau nol, berarti solusinya tak terbatas.
   b. Jika koefisien fungsi tujuan variabel tersebut berharga negatif (untuk maksimasi) atau positif (untuk minimasi), maka nilai fungsi tujuannya tidak terbatas.

3.6. Menyelesaikan Persoalan lP dengan Pembatas Bertanda ≥ dan atau =

Dalam pembicaraan mengenai metode simpleks, kita telah menggunakan variabel *slack* sebagai solusi basis awal, sedemikian sehingga masing-masing merupakan ruas kanan yang berharga positif pada masing-masing persamaan.

Sekarang perhatikan untuk kasus yang persamaan pembatasnya tidak menggunakan tanda (≤), tetapi bertanda (=) atau (≥). Untuk kasus yang persamaan pembatasnya bertanda (=), daerah fisibelnya hanya berupa segmen garis sehingga kita tidak dapat memperoleh solusi fisibel basis awal karena tidak ada variabel *slack* yang dapat digunakan sebagai variabel basis awalnya. Sebagai contoh, apabila persamaan pembatas ketiga dari persoalan PT Indah Gelas diubah menjadi $3x_1 + 2x_2 \leq 18$ menjadi $3x_1 + 2x_2 = 18$, maka daerah fisibelnya hanya berupa segmen garis yang menghubungkan titik (2, 6) dengan (4, 3). Demikian juga untuk kasus dengan persamaan pembatas bertanda (≥), kita tidak akan memiliki solusi fisibel basis awal karena ruas kanan berharga negatif. Contoh: $3x_1 + 2x_2 \geq 18$, adalah sama dengan $-3x_1 - 2x_2 \leq -18$. Dengan menambah variabel *slack* menjadi $-3x_1 - 2x_2 + S_3 = -18$, S_3 tidak bisa menjadi variabel basis awal karena harganya negatif.

Untuk menyelesaikan kedua jenis kasus tersebut, kita memerlukan adanya variabel *dummy* (variabel palsu) yang disebut *variabel artifisial*, sehingga variabel basis awal bisa tetap ada. Sebagai ilustrasi kita lihat contoh berikut:

Contoh 1:

Maksimalkan: $z = 3x_1 + 5x_2$

Berdasarkan pembatas :

$$
\begin{array}{rcrcr}
x1 & & & \leq & 4 \\
 & & 2x2 & \leq & 12 \\
3x1 & + & 2x2 & = & 18
\end{array}
$$

$x1, x2 \geq 0$

Bentuk di atas kita ubah menjadi:

$$
\begin{array}{rrcrcrcrcrcr}
z & - & 3x1 & - & 5x2 & & & & & & = & 0 \\
 & & x1 & & & + & S1 & & & & = & 4 \\
 & & & & 2x2 & & & + & S2 & & = & 12 \\
 & & 3x1 & + & 2x2 & & & & & + R3 & = & 18
\end{array}
$$

$x1, x2, S1, S2, R3 \geq 0$

Pengaruh variabel artifisial (R) ini adalah untuk memperluas daerah fisibel. Pada kasus di atas, daerah fisibel berkembang dari semula berupa segmen garis yang menghubungkan titik-titik (2,6) dan (4, 3) menjadi bidang ABCDE.

Contoh 2:

Maksimalkan : $z = 3x1 + 5x2$

Berdasarkan pembatas :

$$
\begin{array}{rcrcr}
x1 & & & \geq & 4 \\
 & & 2x2 & \geq & 12 \\
3x1 & + & 2x2 & = & 18
\end{array}
$$

$x1, x2 \geq 0$

Bentuk diatas kita ubah menjadi:

$$
\begin{array}{lcccccccccccccl}
z & - & 3x1 & - & 5x2 & & & & & & & & & = & 0 \\
 & & x1 & & & - & S1 & & & + & R1 & & & = & 4 \\
 & & & & 2x2 & & & - & S2 & & & + & R2 & = & 12 \\
 & & 3x1 & + & 2x2 & & & & & & & + & R3 & = & 18
\end{array}
$$

$x1, x2, S1, R1, R2, R3 \geq 0$

Pada akhirnya, iterasi-iterasi metode simpleks akan secara otomatis menjadikan variabel artifisial ini tidak muncul lagi (berharga nol), yaitu apabila persoalan semula telah terselesaikan. Dengan kata lain, kita gunakan variabel artifisial ini hanya untuk memulai solusi, dan harus menghilangkannya (menjadikannya berharga nol) pada akhir solusi. Jika tidak demikian, solusi yang diperoleh akan tidak fisibel.

4. Teori Dualitas dan Analisis Kepekaan

4.1. Teori Dualitas

Teori dualitas merupakan salah satu konsep programa linier yang pentingdan menarik ditinjau dari segi teori dan praktiknya. Ide dasar yang melatarbelakangi teori ini adalah bahwa setiap persoalan programa linier memiliki suatu programa linier lain yang saling berkaitan yang disebut “dual”, sedemikian sehingga solusi pada persoalan semula (yang disebut “primal”) juga memberi solusi pada dualnya.

Pendefinisian dual ini akan tergantuang pada jenis pembatas, tanda-tanda variabel, dan bentuk optimasi dari persoalan primalnya. Akan tetapi, karena setiap persoalan programa linier harus dibuat dalam bentuk standar terlebih dahulu sebelum modelnya dipecahkan, maka pendefinisian di bawah ini akan secara otomatis meliputi ketiga hal diatas.

Bentuk umum masalah primal-dual adalah sebagai berikut

Primal :

Maksimalkan : $z = c1x1 + c2x2 + ... + c_n x_n$

Berdasarkan pembatas :

$a11x1 + a12x2 + ... + a1_n x_n \quad <= b1$

$a21x1 + a22x2 + ... + a2_n x_n \quad <= b2$

.

.

.

$a_m 1x1 + a_m 2x2 + ... a_{mn} X_n \quad <= b_m$

$x1.x2, ..., x_n \quad <= 0$

Dual :

Minimalkan : $w = b_1 y_1 + b_2 y_2 + ... + b_m y_m$

Berdasarkan pembatas:

$a11y1 + a21y2 + ... + a_m 1y_m \quad >= c1$

$a12y1 + a22y2 + ... + a_m 2y_m \quad >= c2$

.

.

$a1_n y1 + a2_n y2 + ... + a_{mn} y_{mn} \quad >= c_n$

$y1, y2, ... , y_m \quad >= 0$

Jika dibandingkan kedua persoalan diatas, ternyata terdapat korespondensi antara primal dengan dual sebagai berikut:

1. Koefisien fungsi tujuan primal menjadi konstanta ruas kanan bagi dual, sedangkan konstanta ruas kanan primal menjadi koefisien fungsi tujuan bagi dual.
2. Untuk setiap pembatas primal ada satu variabel dual, dan untuk setiap variabel primal ada satu pembatas dual.
3. Tanda ketidaksamaan pada pembatas akan tergantung pada fungsi tujuannya.
4. Fungsi tujuan berubah bentuk (maksimasi menjadi minimasi dan sebaliknya).
5. Setiap kolom pada primal berkorespondensi dengan baris (pembatas) pada dual.
6. Setiap baris (pembatas) pada primal berkorespondensi denagn kolom pada dual.
7. Dual dari dual adalah primal.

4.2. Hubungan Primal-Dual

Untuk menjelaskan hubungan antara primal dengan dual, perhatikan ilustrasi berikut ini:

Primal

Masksimalkan: $z = 5x_1 + 12x_2 + 4x_3$

Berdasarkan pembatas :

$$\begin{array}{rcrcrcr} x_1 & + & 2x_2 & + & x_3 & \leq & 10 \\ 2x_1 & - & x_2 & + & 3x_3 & = & 8 \end{array}$$

$x_1, x_2, x_3 \geq 0$

Bentuk standar:

Maksimalkan : $z = 5x_1 + 12x_2 + 4x_3 + OS_1 – MR_2$

berdasarkan pembatas:

$$\begin{array}{rcrcrcrcrcr} x_1 & + & 2x_2 & + & x_3 & + & S_1 & & & = & 10 \\ 2x_1 & - & x_2 & + & 3x_3 & & & + & R_2 & = & 8 \end{array}$$

$x_1, x_2, x_3, S_1, R_2 \geq 0$

Dual dari persoalan diatas adalah:

Minimalkan: $w = 10y_1 + 8(y_2' – y_2'')$

Berdasarkan pembatas:

$$\begin{array}{rcrcr} y_1 & + & 2(y_2' – y_2'') & \geq & 5 \\ 2y_1 & - & (y_2' – y_2'') & \geq & 12 \\ y_1 & + & 3(y_2' – y_2'') & \geq & 4 \end{array}$$

$y_1 \geq 0$, tidak terbatas dalam tanda

Bentuk standar:

Minimalkan: $W = 10y_1 + 8(y_2' - y_2'') – 0(S_1 + S_2 + S_3) + M(R_1 + R_2 + R_3)$

Berdasarkan pembatas:

$$\begin{array}{rcrcrcrcrcrcrcrcr} y_1 & + & 2(y_2'-y_2'') & - & S_1 & & & & & + & R_1 & & & & & = & 5 \\ 2y_1 & - & (y_2'-y_2'') & & & - & S_2 & & & & & + & R_2 & & & = & 12 \\ y_1 & + & 3(y_2'-y_2'') & & & & & - & S_3 & & & & & + & R_3 & = & 4 \end{array}$$

$y_1, y_2', y_2'' \geq 0$

4.3. Sifat-sifat Primal Dual yang Penting

Sifat-sifat primal-dual ini penting untuk dipahami terutama pada saat membicarakan masalah anailis kepekaan. Dengan menggunakan sifat-sifat ini kita dapat menentukan nilai variabel-variabel tertentu dengan cara yang sangat efisien. Ada empat sifat yang perlu diketahui, yaitu:

1. Menentukan koefisien fungsi tujuan variabel-variabel basis awal
2. Menentukan koefisien fungsi tujuan variabel-variabel nonbasis awal
3. Menentukan nilai ruas kanan (solusi) dari variabel-variabel basis
4. Menentukan koefisien pembatas

4.4. Metode Dual Simpleks

Apabila pada suatu iterasi kita mendapat persoalan program liner yang sudah optimal (berdasarkan kondisi optimalitas), tetapi belum fisibel (ada pembatas nonnegatif yang tidak terpenuhi), maka persoalan tersebut harus diselesaikan dengan menggunakan metode dual simpleks. Syarat digunakan metode ini adalah bahwa seluruh pembatas harus merupakan ketidaksamaan yang bertanda (≤), sedangkan fungsi tujuan bisa berupa maksimasi atau minimasi.

Pada dasarnya metode dual simpleks ini menggunakan tabel yang sama seperti metode simpleks pada primal, tetapi *leaving* dan *entering variable*-nya ditentukan sebagai berikut:

1. *Leaving variable* (kondisi optimalitas)

   Yang menjadi *leaving variable* pada dual simpleks adalah variabel basis yang memiliki harga negatif terbesar. Jika semua variabel basis telah berharga positif atau nol, berarti keadaan fisibel telah tercapai
2. *Entering variable* (kondisi optimalitas)
   a. Tentukan perbandingan (rasio) antara koefisien persamaan z dengan koefisien persamaan *leaving variable*. Abaikan penyebut yang positif atau nol. Jika semua penyebut berharga positif atau nol, berartipersoalan yang bersangkutan tidak memiliki solusi fisibel.
   b. Untuk persoalan minimasi, *entering variable* adalah variabel dengan rasio terkecil, sedangkan untuk persoalan maksimasi, *entering variable* adalah variabel dengan rasio absolut terkecil.

4.5. Beberapa Perumusan Penting

Formulasi suatu persoalan LP dapat pula kita nyatakan dalam bentuk sebagai berikut:

Maks/Min: $z = c_{BV}\, x_{BV} + c_{NBV}\, x_{NBV}$

Berdasarkan:

$$Bx_{BV} + Nx_{NBV} = b$$

di mana:

c_{BV} yaitu koefisien fungsi tujuan untuk variabel-variabel basis, adalah vektor baris berukuran 1 x m.

c_{NBV} yaitu koefisien fungsi tujuan untuk variabel-variabel non-basis, adalah vektor baris berukuran 1 x (n - m)

B merupakan matriks berukuran m x m, yaitu matriks dari koefisien pembatas teknologis, khusus untuk variabel-variabel basis.

a_j adalah kolom pada pembatas untuk variabel x_j.

b adalah vektor kolom m x 1, yaitu ruas kanan pembatas

N adalam matriks m x (n -m), yaitu matriks yang kolom-kolomnya adalah kolom variabel non-basis.

4.6. Analisis Kepekaan

Analisis kepekaan adalah analisis yang dilakukan untuk mengetahui akibat/pengaruh dari perubahan yang terjadi pada parameter-parameter programa linier terhadap solusi optimal yang telah dicapai.

4.7. Shadow Prices

Untuk mendefinisikan kegunaan konsep *shadow price* ini, misalkan kita memiliki persoalan maksimasi dengan m pembatas dimana b_i adalah ruas kanan dari pembatas ke-i

Definisi:

> Shadow price pembatas ke-I dari suatu persoalan maksimasi adalah besaran yang menyatakan peningkatan nilai z-optimal sebagai akibat dinaikkanya harga b_i sebesar 1 unit, yaitu dari b_i menjadi (b_i + 1)

Dengan menggunakan teorema dual, *shadow price* dari pembatas ke-I dapat ditentukan dengan mudah

4.8. Dualitas dan Analis Kepekaan

Teorema dualitas mengatakan: jika set dari variabel basis (BV) fisibel, maka BV akan optimal jika dan hanya jika solusi dual yang bersangkutan (yaitu $C_{BV}B^{-1}$) adalah dual fisibel.

Hal tersebut di atas akan digunakan sebagai cara lain untuk menyelesaikan analisis kepekaan, untuk kasus yang berkaitan dengan:

a. Perubahan koefisien fungsi tujuan pada variabel nonbasis
b. Perubahan kolom variabel nonbasis
c. Penambahan aktivitas baru

Kita tahu bahwa perubahan-perubahan ini tidak akan mengganggu fisibilitas dari variabel basis (BV). Berdasarkan teorema diatas, kita juga tahu bahwa ketiga jenis perubahan ini akan menyebabkan solusi basis saat ini tetap optimal jika dan hanya jika solusi dual saat ini (yaitu $C_{BV}B^{-1}$) yaitu tetap dua fisibel.

5. Tipe-tipe Khusus Persoalan Programa Linier

5.1. Persoalan Transportasi

Persoalan transportasi membahas masalah pendistribusian suatu komoditas atau prosuk dari sejumlah sumber (*supply*) kepada sejumlah tujuan (*destination, demmand*), dengan tujuan meminimalkan ongkos pengangkutan yang terjadi.

Ciri-ciri khusus persoalan transportasi ini adalah:

1. Terdapat sebuah sumber dan sejumlah tujuan tertentu.
2. Kuantitas komoditas atau barang yang didistribusikan dari setiap sumber dan yang diminta oleh setiap tujuan, besarnya tertentu.
3. Komoditas yang dikirim atau diangkut dari suatu sumber ke suatu tujuan, besarnya sesuai dengan permintaan dan atau kapasitas sumber.
4. Ongkos pengangkutan komodita sdari suatu sumber ke suatu tujuan, besarnya tertentu.

5.2. Model Transshipment

Model *transshipment* adalah model transportasi yang memungkinkan untuk dilakukannya pengiriman barang (komoditas) cara tidak langsung, di mana barang dari suatu sumber dapat berada pada sumber lain atau tujuan lain sebelum mencapai tujuan akhirnya. Jadi, pada model *transshipment* ini suatu sumber sekaligus dapat berperan sebagai tujuan dan sebaliknya, suatu tujuan dapat juga berperan sebagai sumber.

Dalam model ini, setiap sumber maupun tujuan dipandang sebagai titik potensial bagi *demmand* maupun *supply*. Oleh karenaitu untuk menjamin bahwa tiap titik potensial tersebut mampu menampung total barang disamping jumlah barang yang telah ada pada titik-titik tersebut, maka perlu ditambahkan kepada titik-titik itu kuantitas *supply* dan *demmand*-nya masing-masing sebesar B.

$$B \geq \sum_{i=1}^{m} a_i = \sum_{j=1}^{n}$$

Dengan demikian, apabila ada persoalan transportasi sebagai berikut:

	T1	T2	T3	
S1	10	20	30	100
S2	20	50	40	200
	100	100	100	

Maka persoalan *transshipment*-nya adalah:

	S1	S2	T1	T2	T3	
S1			10	20	30	100+B
S2			20	50	40	100+B
T1						B
T2						B
T3						B
	B	B	100+B	100+B	100+B	

Model diatas baru lengkap apabila ongkos per unit pengangkut untuk baris-baris dan kolom-kolom yang lainnya telah ditetapkan. Dalam hal ini perlu diingat bahwa ongkos per unit pada elemen-elemen diagonal adalah nol.

Asumsikan bahwa seluruh ongkos per unitnya telah ditentukan, maka model *transshipment* selengkapnya adalah:

	S1	S2	T1	T2	T3	
S1	0	80	10	20	30	400
S2	10	0	20	50	40	500
T1	20	30	0	40	10	300
T2	40	20	10	0	20	300
T3	60	70	80	20	0	300
	300	300	400	400	400	

Selanjutnya persoalan diatas diselesaikan dengan menggunakan teknik transportasi seperti biasa, sehingga diperoleh solusi optimal sebagai berikut :

	S1	S2	T1	T2	T3	
S1	300			100		400
S2		300	200			500
T1			200		100	300
T2				300		300
T3					300	300
	300	300	400	400		

Elemen-elemen diagonal dari tabel di atas kita abaikan, karena secara fisik tidak mempunyai arti apa-apa. Solusi optimal diatas menyatakan bahwa sumber 2 (S2) mengirimkan seluruh *supply*-nya pada tujuan 1 (T1), dimana 100 unit disimpan untuk memenuhi *demmand* pada tujuan 1 tersebut, dan sisanya, yaitu sebanyak 100 unit, kemudian dikirimkan kepada tujuan 3 (T3) untuk memenuhi *demmand* pada tujuan 3. Adapun *demmand* pada tujuan 2 (T2) dipenuhi langsung dari sumber 1 (S1).

5.3. Model Penugasan (*Assignment Model*)

Model penugasan merupakan kasus khusus dari model transportasi, dimana sejumlah m sumber ditugaskan kepada sejumlah n tujuan (satu sumber untuk satu tujuan) sedemikian hingga didapat ongkos total yang minimal.

Biasanya yang dimaksud dengan sumberadalah pekerjaan (atau pekerja), sedangkan yang dimaksud dengan tujuan adalah mesin-mesin. Jadi, dalam hal ini, ada m pekerjaan yang ditugaskan pada n mesin, dimana apabila pekerjaan i (i = 1, 2, ..., m) ditugaskan kepada mesin (j = 1, 2, ..., n) akan muncul ongkos penugasa c_{ij}. Karena satu pekerjaan ditugaskan hanya pada satu mesin, maka *supply* yang dapat digunakan pada setiap sumber adalah 1 (atau $a_i = 1$, untuk seluruh i). Demikian pula halnya dengan mesin-mesin; karena satu mesin hanya dapat menerima satu pekerjaan, maka demmand dari setiap tujuan adalah 1 (atau $b_j = 1$, untuk seluruh j). Jika ada suatu pekerjaan yang tidak dapat ditugaskan pada mesin tertentu, maka c_{ij} yang berkorespondensi dengannya dinyatakan sebagai M, yang merupakan ongkos yang sangat tinggi.

Sebelum model ini dapat dipecahkan dengan teknik transportasi, terlebih dahulu persoalannya harus diseimbangkan dengan menambahkan pekerjaan-pekerjaan atau mesin-mesin khayalan, bergantung pada apakah $m < n$ atau $m > n$. Dengan demikian diasumsikan bahwa $m = n$.

6. Analisis Jaringan

Pada bab sebelumnya kita telah membatasi perhatian pada persoalan-persoalan transportasi atau distribusi yang berkaitan dengan masalah pengiriman komoditas dari suatu sumber ke suatu tujuan dengan ongkos transportasi minimum. Ternyata model transportasi ini dapat juga direpresentasikan dan diselesaikan sebagai suatu jaringan.

Persoalan jaringan dapat dibagi menjadi tiga macam, yaitu:

1. Persoalan rute terpendek (*shortest route*);
2. Persoalan minimasi jaringan atau rentang pohon minimal;
3. Persoalan aliran maksimal.

6.1. Terminologi Jaringan

Berdasarkan terminologi teori grafis, maka suatu grafik akan terdiri atas satu set titik-titik yang dihubungkan, yang disebut *node*. *Node* tertentu dihubungkan oleh garis yang disebut busur.

Beberapa terminologi tambahan dari jaringan ini adalah:

- Siklus, yaitu lintasan yang menghubungkan suatu node dengan *node* itu sendiri.
- Pohon, yaitu grafik yang mempunyai lintasan yang menghubungkan pasangan-pasangan *node*, dimana siklus tidak terjadi.
- Busur maju *i*, yaitu busur yang meninggalkan *node i*.
- Busur mundur *i*, yaitu busur yang menuju *node i*.
- Kapasitas aliran suatu busur dengan arah tertentu, yaitu batas atas aliran (atau jumlah aliran total) yang fisibel pada busur tersebut.
- Sumber suatu jaringan, yaitu *node* yang menjadi awal bagi busur-busurnya, dimana aliran bergerak meninggalkannya.
- Tujuan suatu jaringan, yaitu *node* yang dituju oleh busur-busurnya, dan aliran masuk ke *node* tersebut.

7. Perencanaan dan Pengendalian Proyek Dengan Pert-CPM

PERT-*type system* ini dirancang untuk membantu dalam perencanaan dan pengendalian sehingga tidak langsung terlibat dalam optimasi. Tujuan sistem ini adalah:

1. Untuk menentukan probabilitas tercapainya batas waktu proyek.
2. Untuk menetapkan kegiatan mana (dari suatu proyek) yang merupakan *bottlenecks* (menentukan waktu penyelesaian seluruh proyek) sehingga dapat diketahui pada kegiatan mana kita harus bekerja keras agar jadwal dapat terpenuhi.
3. Untuk mengevaluasi akibat dari perubahan-perubahan program, PERT-*type system* ini juga dapat mengevaluasi akibat dari terjadinya penyimpangan pada jadwal proyek.

7.2. Simbol-simbol yang digunakan

Dalam menggambarkan suatu *network* digunakan tiga buah simbol sebagai berikut:

1. Anak panah (*arrow*)/ $\longrightarrow$

   Anak panah ini menyatakan sebuah kegiatan atau aktivitas. Kegiatan di sini didefinisikan sebagai hal yang memerlukan durasi/jangka waktu tertentu dalam pemakaian sejumlah *resources* (sumber tenaga, perlatan, biaya, material). Panjang atau miringnya anak panah ini tidak memiliki arti. Kepala anak panah menjadi pedoman arah tiap kegiatan, yang menunjukkan bahwa suatu kegiatan dimulai dari permulaan dan berjalan maju sampai akhir dengan arah dari kiri ke kanan.

2. Lingkaran kecil (*node*)/ ○

   Lingkaran kecil ini menyatakan sebuah kejadian atau peristiwa atau *event*. Kejadian/*event* di sini didefinisikan sebagai ujung atau pertemuan dari satu atau beberapa kegiatan.

3. Anak panah terputus-putus/ - - - -►

   Anak panah terputus-putus ini menyatakan kegiatan semu atau *dummy*. *Dummy* di sini berguna untuk membatasi mulainya kegiatan. Seperti halnya kegiatan biasa, panjang dan kemiringan *dummy* juga tidak memiliki arti. Bedanya dengan kegiatan biasa adalah bahwa *dummy* tidak memiliki durasi/jangka waktu tertentu karena tidak memakai atau menghabiskan sejumlah *resources*.

Dalam pelaksanaannya, simbol-simbol ini digunakan dengan mengikuti aturan-aturan sebagai berikut :

1. Diantara dua *event* yang sama, hanya boleh digambarkan satu anak panah.
2. Nama suatu aktivitas dinyatakan dengan huruf atau dengan nomor *event*.
3. Anak panah harus mengalir dari *event* bernomor rendah ke *event* bernomor tinggi.
4. Diagram hanya memiliki sebuah *initial event* dan sebuah *terminal event*.

7.2. Penentuan Waktu

Setelah *network* suatu proyek dapat digambarkan, langkah berikutnya adalah mengestimasi waktu yang diperlukan untuk masing-masing aktivitas, dan menganalisis seluruh diagram *network* untuk menentukan waktu terjadinya masing-masing kegiatan.

Dalam mengetimasi dan menganalisis waktu ini, akan kita dapatkan satu atau beberapa lintasan tertentu dari kegiatan-kegiatan pada *network* tersebut yang menentukan jangka waktu penyelesaian seluruh proyek. Lintasan ini disebut *lintasan kritis* (*critical path*). Disamping lintasan krisis ini terdapat lintasan –lintasan lain yang memiliki jangka waktu yang lebih pendek daripada daripada lintasan krisis. Dengan demikian, maka lintasan yang tidak kritis ini memiliki waktu utnuk bisa terlambat, yang dinamakan *float*.

Float memberikan sejumlah kelonggaran waktu dan elastisitas pad asebuah *network*, dan ini dipakai pada waktu penggunaan *network* dalam praktik , atau digunakan pda waktu mengerjakan penentuan jumlah material, peralatan, dan tenaga kerja. *Float* ini terbagi atas dua jenis, yaitu *float total* dan *free float*.

7.3. Perhitungan Maju

Ada tiga langkah yang dilakukan pada perhitungan maju, yaitu:

1. Saat tercepat terjadinya *initial event* ditentukan pada hari ke nol sehingga untuk initial event berlaku TE = O. Asumsi ini tidak benar untuk proyek yang berhubungan dengan proyek-proyek lain.
2. Kalau *initial event* terjadi pada hari yang ke-nol maka :

   $ES_{(i,\ j)} \quad = \quad TE_{(j)} \quad = \quad 0$

   $EF_{(i,\ j)} \quad = \quad TS_{(i,\ j)} \quad + \quad t_{(i,\ j)}$

   $\quad = \quad TE_{(i)} \quad + \quad t_{(i,\ j)}$
3. Event yang menggabungkan beberapa aktivitas (*merge event*).

Sebuah *event* hanya dapat terjadi jika aktivitas-aktivitas yang mendahuluinya telah diselesaikan. Maka saat paling cepat terjadinya sebuah event sama dengan nilai terbesar dari saat tercepat untuk menyelesaikan aktivitas yang berakhir pada *event* tersebut.

7.4. Perhitungan Mundur

Seperti halnya pada perhitungan maju, pada perhitungan mundur ini pun terdapat tiga langkah, yaitu:

1. Pada terminal *event* berlaku TL = TE
2. Saat paling lambat untuk memulai suatu aktivitas sama dengan saat paling lambat saat paling lambat untuk menyelesaikan aktivitas itu dikurangi dengan *duration* aktivitas tersebut.
3. Event yang mengeluarkan beberapa aktivitas (*burst event*).

Setiap aktivitas hanya dapat dimulai apabila *event* yang mendahuluainya telah terjadi. Oleh karena itu, saat paling lambat terjadinya sebuah *event* sama dengan nilai terkecil dari saat-saat paling lambat untuk memulai aktivitas-aktivitas yang berpangkal pada *event* tersebut.

7.5. Perhitungan Kelonggaran Waktu

Setelah perhitungan maju dan pehitungan mundur selesai dilakukan, maka berikutnya harus dilakukan perhitungan kelonggaran waktu (*float/slack*) dari aktivitas (i,j) yang terdiri atas *total float* dan *free float*.

Total float adalah jumlah waktu di mana waktu penyelesaian suatu aktivitas dapat diundur tanpa mempengaruhi saat paling cepat dari penyelesaian proyek secara keseluruhan. Karena itu *total float* dapat dihitung dengan cara mencari selisih antara saat paling lambat dimulainya aktivitas dengan saat paling cepat dimulainya aktivitas (LS – LE), atau bisa juga dengan mencari selisih antara saat paling lambat diselesaikannya aktivitas dengan saat paling cepat diselesaikannya aktivitas (LF – EF). Dalam hal ini cukup dipilih salah satu saja.

Yang dimaksud *free float* adalahjumlah waktu dimana penyelesaian suatu aktivitas dapat diukur tanpa mempengaruhi saat paling cepat dari dimulainya aktivitas yang lain atau saat paling cepat terjadinya *event* lain pada *network*. *Free float* aktivitas (i, j) dihitung dengan cara mencari selisih antara saat tercepat terjadinya *event* diujung aktivitas dengan saat tercepat diselesaikannya aktivitas (i, j) tersebut.

Suatu aktivitas yang tidak mempunyai kelonggaran (*float*) disebut aktivitas kritis. Aktivitas-aktivitas kritis ini akan membentuk lintasan kritis yang biasanya dimulai dari *initial event* sampai ke *terminal event*. Walaupun demikian, lintasan kritis ini bisa juga bersifat parsial. Bila waktu merupakan faktor yang sangat menentukan bagi keberhasilan suatu proyek, maka lintasan kritis inilah yang perlu dikendalikan.

7.6. Pembuatan Peta Waktu dan Pengaturan Sumber

Sebagai langkah terakhir dari perhitungan *network* ini adalah membuat peta waktu yang merupakan jadwal pelaksanaan proyek. Peta ini harus dibuat dengan memperhatikan batasan-batasan dari sumber yang dapat digunakan, karena tidak mungkin beberapa aktivitas dapat diselesaikan sekaligus mengingat terbatasnya tenaga kerja dan peralatan.

Pada saat pembuatan peta waktu ini, kita dapat memanfaatkan *total float* aktivitas-aktivitas yang tidak kritis untuk digunakan dalam pengaturan sumber yang diperlukan. Caranya adalah dengan menggeser-geser aktivitas yang tidak kritis ini (ke depan atau ke belakang) dalam batas waktu maksimal yang dapat digunakannya. Untuk membuat peta ini kita dapat melihat langsung diagram *network*-nya.

Peranan *total float* dan *free float* dalam penjadwalan aktivitas-aktivitas yang tidak kritis ini mengikuti dua aturan umum sebagai berikut:

1. Jika *total float* sama dengan *free float*, maka aktivitas-aktivitas yang tidak kritis dapat dijadwalkan dimana saja, diantara ES dan LF-nya masing-masing.
2. Jika *free float* lebih kecil dari *total float*, maka saat dimulainya aktivitas yang tidak kritis ini dapat diundur relatif terhadap saat tercepat dimulainya saktivitas tersebut. Lamanya pengunduran waktu ini tidak boleh lebih besar dari besarnya *free float* sehingga penjadwalan dari aktivitas-aktivitas yang berikutnya tidak terganggu.

Tujuan dari pengaturan jadwal pelaksanaan aktivitas-aktivitas dalam suatu proyek ini adalah agar dapat memperkecil sumber maksimal yang diperlukan.

7.7. Perkiraan waktu penyelesaian suatu aktivitas

Sampai saat ini kita telah menggunakan data waktu penyelesaian suatu aktivitas (*duration*) sebagai dasar perhitungan waktu dan penjadwalan proyek. Tetapi, bagaimanakah cara memperkirakan waktu yang diperlukan oleh aktivitas-aktivitas tersebut?

Ada dua cara yang biasa digunakan untuk memperkirakan waktu penyelesaian suatu aktivitas ini, yaitu:

1. *Single duration estimate* atau perkiraan waktu (*duration*) tunggal untuk setiap aktivitas. Cara ini dapat dilakukan pabila *duration* dapat diketahui dengan akurat dan tidak terlalu berfluktuasi. Pendekatan CPM (*Critical Path Method*) menggunakan cara ini karena CPM beranggapan bahwa setiap fluktuasi dapat diatasi dengan fungsi kontrol.
2. *Triple duration estimate*, yaitu cara perkiraan waktu yang didasarkan atas tiga jenis *duration* sebagai berikut:

   T_o = *optimistic duration,* yaitu waktu yang diperlukan untuk menyelesaikan suatu aktivitas jika tidak terjadi kesalahan pada pelaksanaan aktivitas itu (segala sesuatunya berjalan baik sekali).

   T_m = *most likely duration*, yaitu waktu yang paling sering terjadi bila aktivitas dilakukan berulang-ulang (dalam kondisi normal).

T_p = *pesimistic duration*, yaitu waktu yang diperlukan bila terjadi kesalahan pada pelaksanaan aktivitas yang bersangkutan.

Cara ini merupakan dasar perhitungan untuk PERT yang memiliki asumsi dasar bahwa jika suatu aktivitas dilakukan berkali-kali, maka *actual times* (waktu yang nyata untuk menyelesaikan aktivitas itu) akan membentuk distribusi frekuensi Beta, di mana *optimistic* dan *pesimistic duration* merupakan buntut (*tail*), sedangkan *most likely duration* adalah mode dari distribusi Beta tersebut.

7.8. Penentuan Ongkos dalam Penjadwalan Proyek

Dalam penjadwalan proyek, aspel ongkos diperhitungkan dengan membuat hubungan ongkos dengan *duration* untuk setiap aktivitas pada proyek itu. Yang dimaksud dengan ongkos di sini adalah ongkos langsung saja, tidak termasuk ongkos-ongkos administrasi, supervisi dan lain-lain.

Ada suatu batas yang dinamakan *crash time* (batas waktu percepatan) yang menyatakan bahwa pengurangan waktu berikutnya (yang melampaui batas) tidak akan efektif lagi.

Setelah hubungan ongkos dengan waktu ini ditentukan, selesaikanlah aktivitas-aktivitas proyek dalam duration normalnya. Kemudian tentukan lintasan-lintasan kritis dan ongkos langsungnya. Langkah berikutnya adalah mempertimbangkan pengurangan *duration*. Karena pengurangan waktu ini hanya akan efektif jika *duration* aktivitas-aktivitas kritis dikurangi, maka yang perlu diperhatikan adalah aktivitas-aktivitas kritis itu saja. Agar diperoleh pengurangan *duration* dengan ongkos sekecil mungkin, maka kita harus menekan sebanyak mungkin aktivitas-aktivitas kritis yang memiliki kemiringan garis ongkos – waktu terkecil.

Banyaknya aktivitas yang dapat ditekan ini dibatasi oleh *crash time* masing-masing. Namun batasan-batasan lain juga harus diperhitungkan sebelum menetapkan jumlah aktivitas yang pasti dapat dipersingkat. Sebagai hasil penekanan suatu aktivitas ini adalah jadwal baru yang mungkin mempunyai lintasan kritis yang baru pula. Ongkos jadwal baru ini tentunya lebih besar dari jadwal sebelumnya. Dari jadwal baru ini kita pilih *aktivitas-aktivitas kritis dengan kemiringan terkecil* untuk dipercepat pelaksanaannya. Prosedur ini diulangi sehingga seluruh aktivitas kritis berada pada *crash time* masing-masing.

8. Programa Bilangan Bulat

8.1. Pendahuluan

Programa bilangan bulat (*Integer Programming*) adalah bentuk lain dari programa linier di mana asumsi divisibilitasnya melemah atau hilang sama sekali. Bentuk ini muncul karena dalam kenyataannya tidak semua variabel keputusan dapat berupa bilangan pecahan. Misalnya, jika variabel keputusan yang dihadapi berkaitan dengan jumlah mesin yang diperlukan pada suatu horizon perencanaan, maka jawaban 10/3 mesin sangat tidak realistis dalam konteks keputusan yang nyata. Dalam hal ini harus ditentukan apakah akan menggunakan 3 atau 4 mesin.

Asumsi divisibilitas melemah, artinya sebagian dari nilai variabel keputusan harus berupa bilangan bulat (integer) dan sebagian lainnya boleh berupa bilangan pecahan. Persoalan IP di mana hanya sebagian dari variabel keputusannya yang harus integer disebut sebagai IP campuran.

8.2. Programa Linier Relaksasi

Bentuk programa linier yang diperoleh dengan mengabaikan pembatas integer disebut sebagai programa linier relaksasi.

Contoh:

Maksimalkan: $z = 3x_1 + 2x_2$

berdasarkan:

$x_1 + x_2 \leq 6$

$x_1, x_2 \geq 0$

dan

Maksimalkan: $z = x1 – x2$

Berdasarkan:

$x_1 + 2x_2 \leq 2$

$2x_1 - x_2 \leq 1$

$x_1, x_2 \geq 0$

Setiap persoalan programa bilangan bulat dapat dipandang sebagai persoalan programa linier relaksasi dengan beberapa pembatas tambahan, yaitu pembatas yang menyatakan variabel-variabel mana yang harus berharga integer atau harus berharga nol atau satu. Hal ini berarti bahwa daerah fisibel untuk setiap persoalan programa bilangan bulat akan berada dalam daerah fisibel untuk persoalan programa linier relaksasi yang bersangkutan.

8.3. Memformulasikan Persoalan Programa Bilangan Bulat

Pada bagian ini akan kita lihat bagaimana persoalan-persoalan praktis dapat diformulasikan sebagai persoalan programa bilangan bulat.

Contoh soal:

CV Kayu Indah yang memproduksi meja dan kursi, menjual produknya dengan keuntungan Rp 8.000/unit meja dan RP 5.000/unit kursi. Mengingat perusahaan ini baru dalam tahap permulaan, saat ini perusahaan hanya mampu menyediakan 45 m^3 kayu setiap harinya, sedangkan jam kerja yang tersedia tidak lebih dari 6 jam orang per hari. Jika satu unit meja membutuhkan 1 jam-orang dan 9 m^3 kayu, sedangkan satu unit kursi membutuhkan 1 jam-orang dan 5 m^3 kayu, bagaimanakah formulasi persoalan tersebut agar diperoleh keuntungan yang maksimal?

Jawaban:

Misalkan x_1 = jumlah unit meja yang diproduksi per hari

x_2 = jumlah unit kursi yang diproduksi per hari

Kita tahu bahwa x_1 dan x_2 harus berharga integer. Total keuntungan yang dapat diperoleh (dalam ribuan rupiah) adalah $8x_1 + 5x_2$. Pembatas yang dihadapi berkaitan dengan jam kerja dan kayu yang tersedia setiap harinya. Dengan demikian maka formulasinya adalah :

Maksimalkan: $z = 8x_1 + 5x_2$

berdasarkan:

$$x_1 + x_2 \leq 6$$

$$9x_1 + 5x_2 \leq 45$$

$$x_1, x_2 \geq 0;\ x_1, x_2 \text{ integer}$$

8.4. Menyelesaikan Persoalan IP Murni dengan Teknik Branch-and-Bound

Dalam prakteknya, hampir seluruh persoalan bilangan bulat (IP) diselesaikan dengan menggunakan teknik *branch-and-bound*. Teknik ini mencari solusi optimal dari seuatu persoalan IP dengan mengenumerasi titik-titik dalam daerah fisibel dari suatu subpesroalan. Sebelum menjelaskan bagaimana cara kerja teknik ini, baik murni maupun campuran, dapat diperoleh bentuk persoalan programa linier relaksasi. Jika solusi dari LP relaksasi ini memiliki seluruh variabel yang berharga integer, maka sulusi optimal dari persoalan IP relaksasi itu juga solusi optimal dari persoalan IP. Perhatikan persoalan berikut ini :

Maksimalkan: $z = 3x_1 + 2x_2$

Berdasarkan: $2x_1 + x_2 \leq 6$

$x_1, x_2 \geq 0$; x_1, x_2 integer.

Solusi optimal dari LP relaksasi persoalan IP tersebut adalah x1 = 0, x2 = 6, dan z = 12.

Karena seluruh variabel pada solusi di atas telah berharga integer, maka berdasarkan pernyataan di atas, solusi ini seharusnya juga merupakan solusi dari persoalan IP semula. Sekarang perhatikan bahwa daerah fisibel untuk persoalan IP adalah subset dari titik-titik pada daerah fisibel untuk persoalan LP relaksasi. Dengan demikian nilai z optimal untuk persoalan IP tidak mungkin lebih besar daripada nilai z optimal untuk persoalan LP relaksasi. Hal ini berarti bahwa nilai z optimal untuk persoalan IP harus ≤ 12. Tetapi titik x_1 = 0 dan x_2 = 6 adalah fisibel untuk persoalan IP dan memberikan z =12. Jadi, x_1 = 0, x_2 = 6, dan z = 12 pasti optimal untuk persoalan IP.

8.5. Menyelisaikan Persoalan IP Campuran dengan Teknik Branch-and-Bound

Untuk menyelesaikan persoalan IP campuran dengan teknik *branch-and-bound*, kita modifikasi uraian pada subbab sebelumnya, yaitu dengan hanya melakukan pencabangan pada variabel-variabel yang harus berharga integer. Sebagai contoh perhatikan formulasi berikut ini:

Maksimalkan: $z = 2x_1 + x_2$

Berdasarkan:

5x1 + 2x2 ≤ 8

x1 + x2 ≤ 3

x1, x2 ≥ 0; x1 integer

Seperti biasa, dimulai dengan menyelesaian persoalan LP relaksasi dari persoalan IP di atas. Solusi optimal dari LP relaksaso ini adalah z = 11/3, x1 = 2/3, x2 7/3. Karena x2 dbolehkan berharga pecahan, maka pencabangan tidak akan dilakukan terhadap variabel ini. Dari pencabangan terhadap x1 diperoleh subpersoalan 2 dan 3. Selanjutnya di pilih subpersoalan 2 untuk diselesaikan. Solusi optimalnya adalah z = 3, x1 = 0, x2 = 3, yang merupakan calon solusi. Untuk menyelesaikan subpersoalan 3, solusinya adalah z = 7/2, x1 = 1, x2 = 3/2 yang juga merupakan calon solusi. Karena nilai z dari subpersoalan 3 lebih besar daripada nilai z pada subpersoalan 2, maka subpersoalan 2 dapat diabaikan dan calon solusi dari subpersoalan3, yaitu z = 7/2, x1 = 1, x2 = 3/2, adalah solusi optimal untuk persoalan IP campuran di atas.

8.6. Menyelesaikan Persoalan IP dengan Teknik Cutting Plane

Teknik ini merupakan teknik penyelesaian yan telah terlebi dahulu dikenal dan digunakan orang sebelum kita mengenal teknik *branch-and-bound*. Pendekatan yang dilakukan dalam teknik *cutting plane* adalah dengan mebuat pembatas tambahan yang memotong ruang fisibel dari programa linier (LP) relaksasi sehingga dapat mengeliminasi solusi yang tidak integer. Proses pemotongan akan terus berlangsung sehingga diperoleh solusi dengan seluruh variabel (yang dikehendaki) berharga integer.

Keberhasilan tekni ini sangat terbatas, tergantung pada struktur persoalan yang dihadapi. Artinya, hanya persoalan tertentu yang dapat diselesaikan dengan teknik ini. Karena itu sekarang teknik ini hampir tidak pernah digunakan.

9. Teori Permainan

9.1. Elemen-elemen Dasar Teori Permainan

Perhatikan persoalan *two-person zero-sum game* dengan matriks pembayaran seperti pada tabel berikut ini:

	Pemain B		
	B1	B2	B3
A1	6	9	2
A2	8	5	4

Beberapa pengertian dari persoalan diatas adalah sebagai berikut:

1. Bilangan-bilangan yang ada didalam matriks pembayaran (*payoff matrix*) menyatakan *outcome* atau pembayaran dari strategi permainan yang berbeda. *Payoff* ini diartikan

sebagai suatu ukuran kefektifan seperti uang, persentase daerah pemasaran, atau utilitas. Berdasarkan perjanjian, dalam *two-person zero-sum game* ini bilangan-bilangan positif menyatakan perolehan (keuntungan) bagi pihak yang ditulis pada baris sebagai pemain yang akan memaksimalkan, dan sekaligus merupakan kerugian bagi pihak yang ditusi pada kolom sebagai pemain yang akan meminimalkan.

2. Strategi adalah tindakan pilihan. Dalam hal ini diasumsikan bahwa strategi ini tidak dapat dibolak balik.
3. Aturan permainan menjelaskan tentang bagaimana cara para pemain memilih strategi-strategi mereka.
4. Suatu strategi dinyatakan dominan apabila setiap *payoff* yang ada pada suatu strategi bersifat superior (paling tinggi) dibandingkan dengan setiap *payoff* pada strategi lainnya. Aturan dominasi ini dapat digunakan untuk mengurangi ukuran matriks payoff dan menyederhanakan perhitungan.
5. Nilai permaianan menyatakan ekspektasi *outcome* per permainan jika kedua pemain melakukan strategi terbaik (strategi optimal) mereka. Suatu permainan dikatakan *fair* (adil) jika nilai permainannya nol, dan dinyatakan tidak *fair* jika nilai permainannya bukan nol.
6. Strategi optimal adalah strategi yang menjadikan seorang pemain berada pada posisi pilihan terbaik, tanpa memperhatikan tindakan-tindakan pemain lawannya. Artinya posisi pilihan terbaik ini adalah bahwa setiap penyimpangan dari strategi optimal ini akan menyebabkan turunnya *payoff*.
7. Tujuan model permainan adalah untuk mengidentifikasi strategi optimal bagi masing-masing pemain.

Sebagai akibat digunakannya beberapa asumsi, maka praktis nilai teri permainan ini menjadi agak terbatas. Namun, ide pembuatan keputusan pada situasi persaingan ini merupakan inti dari keputusan-keputusan manajerial.

9.2. Two Person Zero-sum Game

Ada dua jenis persoalan *two-person zero-sum game* yang biasa dijumpai. Jenis pertama adalah permainan yang posisi pilihan terbaiknya bagi setiap pemain dicapai dengan memilih satu strategi tunggal sehingga permainannya disebut permainan strategi murni (*pure-strategy game*). Jenis yang kedua adalah permainan yang kedua pemainnya melakukan pencampuran terhadap strategi-strategi yang berbeda dengan maksud untuk mencapai posisi pilihan terbaik. Dengan demikian, jenis yang kedua ini disebut permainan strategi campuran (*mixed-strategy game*).

9.3. Pure-Strategy Game

Pada *pure-strategy game*, pemain yang akan memaksimalkan akan mengidentifikasi strategi optimalnya dengan menggunakan kriteria maksimum, sedangkan pemain yang akan meminimalkan akan mengidentifikasi strategi optimalnya dengan menggunakan kriteria minimaks. Jika nilai maksimin sama dengan nilai minimaks, maka permainan telah terpecahkan. Untuk menguji hal ini, nilai tersebut harus merupakan nilai maksimalbagi kolom yang bersangkutan. Dalam kasus seperti ini, maka telah tercapai titik keseimbangan. Titik ini dikenal dengan nama titik sadel (*saddle point*).

Jika nilai maksimin tidak sama dengan nilai minimaks, maka titik keseimbangan tidak akan tercapai. Hal ini berarti bahwa *saddle point*-nya tidak ada dan permainan ini tidak dapat diselesaikan dengan strategi murni. Akibatnya, suatu permainan yang tidak mempunyai *saddle point* harus diselesaikan dengan menggunakan strategi campuran.

Perhatikan suatu situasi ketika dua buah perusahaan besar sedang dalam proses perencanaan strategi advertensi masing-masing. Asumsikan bahwa perusahaan A mempuyai dua buah strategi ddan perusahaan B mempunyai tiga buah strategi. Struktur strategy *payoff*-nya adalah sebagai berikut:

		Perusahaan B			Minimal baris
		B1	B2	B3	
Perusahaan A	A1	1	9	2	1
	A2	8	5	4	4 maksimin
Maksimal kolom		8	9	4 Minimaks	

Matriks *payoff* pada tabel diatas adalah untuk pemain yang akan memaksimalkan (Perusahaan A). Jika A memilih strategi A1 maka B akan memilih strategi B1, sehingga *payoff* untuk A adalah 1. Jika A memilih strategi A2 maka B akan memilih strategi B3 sehingga *payoff* untuk A adalah 4. Dengan demikian jelas bahwa perusahaan A akan berada pada posisi pilihan terbaik jika ia melakukan suatu strategi tunggal, yaitu A2.

Sekarang perhatikan tabel matriks *payoff* diatas dari segi kepentingan pemain yang akan meminimalkan (Perusahaan B). Perhatikan bahwa strategi B3 mendominasi strategi B2, sehingga perusahaan B tidak akan pernah memilih B2. Dengan demikian, maka kolom B2 dapat dieliminasi dari matriks *payoff* tanpa mempengaruhi nilai permainan ini. Jika strategi B1 dipilih, maka jelas perusahaan A akan memilih A2, dan B akan kehilangan 8 unit . jika strategi B3 dipilih perusahaan A masih akan memilih A2, tetapi kerugian yang diderita B hanya 4 unit. Dengan demikian, perusahaan B akan berada pada posisi pilihan terbaiknya jika dia melakukan suatu strategi tunggal, yaitu strategi B3.

Dari uraian diatas jelaslah bahwa persoalan *two-person zero-sum game* diatas adalah permainan dengan strategi murni, yang mempunyai suatu *saddle point* dengan nilai 4. Strategi optimal bagi perusahaan A adalah A2, dan strategi optimal bagi perusahaan B adalah B3.

Konklusi dari kriteria maksimin dan kriteria minimaks adalah sebagai berikut:

Kriteria maksimin (untuk pemain yang memaksimalkan)

> Dapatkan nilai minimal dari masing-masing baris. Nilai terbesar (maksimal) dari nilai-nilai maksimal ini adalah nilai maksimin. Dengan demikian, maka untuk permainan denagn strategi murni ini, strategi optimalnya dalah baris tempat nilai maksimin terletak.

Kriteria minimaks (untuk pemain yang meminimalkan)

> Dpatkan nilai maksimum pada masing-masing kolom. Nilai terkecil (minimal) dari nilai-nilai maksimal ini adalah nilai minimaks. Dengan demikian, maka untuk permainan denagn strategi murni ini, strategi optimalnya adalah kolom tempat nilai minimaks terletak.

Karena nilai maksimin sama dengan nilai minimaks (= 4), maka pertanyaan berikutnya adalah: apakah permainan ini mempunyai *saddle point*? Jawabnya adalah ya, karena nilai 4 merupakan nilai maksimum pada kolomnya dan sekaligus nilai maksimal pada barisnya. Setelah kita tahu bahwa *saddle point* ini ada, maka kita dapat menyatakan bahwa strategi optimal bagi A adalah A2 dan strategi optimal bagi B adalah B3.

9.4. Mixed Strategy Game

Seperti telah dijelaskan diatas, pada *game* yang tidak mempunyai *saddle point*, penyelesaiannya harus dilakukan dengan menggunakan strategi campuran.

Perhatikan matriks *payoff* dari suatu game berikut ini;

		Pemain B		Minimum baris
		1	2	3	
Pemain A	1	0	-2	2	-2 ← maksimin
	2	5	4	-3	-3
	3	2	3	-4	-4
Maksimumkan kolom		5	4	2	
				Minimaks	

Karena nilai maksimin tidak sama dengan nilai minimaks, maka permainan diatas tidak mempunyai *saddle point*. Pada *game* ini, jika A memilih strategi 1, maka B memilih strategi 2; tetapi jika B memilih strategi 2, maka A memilih strategi 2, sehingga B akan memilih strategi 3 dan A memilih strategi 1. Demikian seterusnya sehingga permainan seperti ini dikenal sebagai permainan yang tidak stabil (*unstable game*). Berbeda dengan *pure-strategy game*, pada permainan yang tidak mempunyai *saddle point* ini para pemain dapat memainkan seluruh strateginya sesuai dengan set probabilitas yang telah ditetapkan. Tetapkan bahwa:

xi = probabilitas pemain A memilih strategi i (i = 1, 2, ..., m)

yi = probabilitas pemain A memilih strategi j (j = 1, 2, ..., n)

di mana m dan n dalah banyaknya strategi yang dapat dijalankan. Kita tahu bahwa

$$\sum_{i=1}^{m} x_i = \sum_{j=1}^{n} y_j = 1$$

x1, yj ≥ 0 untuk setiap i dan j

Dengan demikian matriks *payoff*-nya dapat digambarkan sebagai berikut:

			B			
		Y1	Y2	...	Yn
	x1	a11	a12		
A	x2	a21	a22		
	.	.	.	...	
	.	.	.	...	
	.	.	.	...	
	xm	am1	am2	...	amn

Solusi persoalan strategi campuran ini masih didasarkan pada kriteria maksimin dan minimaks. Perbedaannya adalah bahwa A akan memilih xi yang memaksimalkan ekspektasi *payoff* terkecil pada suatu kolom, sedangkan B memilih yj yang dapat meminimalkan ekspektasi *payoff* terbesar pada suatu baris.

Nilai-nilai diatas adalah nilai-nilai maksimin dan minimaks dari ekspektasi *payoff*. Seperti halnya pada kasus strategi murni, pada strategi campuran ini pun berlaku hubungan:

Minimaks ekspektasi payoff ≥ maksimin ekspektasi *payoff*.

10. Programa Dinamis

Programa dinamis adalah suatu teknik matematis yang biasanya digunakan untuk membuat suatu keputusan dari serangkaian keputusan yang saling berkaitan. Tujuan utama model ini adalah untuk memudahkan penyelesaian persoalan optimasi yang memiliki karakteristik tertentu. Ide dasar programa dinamis ini adalah membagi persoalan menjadi beberapa bagian yang lebih kecil sehingga memudahkan penyelesaiannya. Akan tetapi, berbeda dengan programa linier, pada persoalan programa dinamis ini tidak ada formulasi matematis yang standar. Karena itu persamaan-persamaan yang terpilih untuk digunakan harus dikembangkan agar dapat memenuhi masing-masing situasi yang dihadapi. Dengan demikian, maka antara persoalan yang satu dengan persoalan lainnya dapat memiliki struktur penyelesaian persoalan berbeda.

10.1. Ilustrasi Programa Dinamis

Seorang *salesman* harus berangkat dari suatu kota ke kota lainnya. Di antara kota asal dan kota tujuan terdapat beberapa kota lain yang dapat digunakan sebagai tempat persinggahan sementara. Meskipun kota awal dan kota tujuan akhir diketahui, untuk tiap kota yang akan ditempuhnya, *salesman* tersebut memiliki beberapa alternatif dengan ongkos berbeda. Data ongkos yang harus dibayar jika *salesman* itu meninggalkan kota i dan menuju kota j (c_{ij}) adalah sebagai berikut:

	2	3	4
1	2	4	3

	5	6	7
2	7	4	6
3	3	2	4
4	4	1	5

	8	9
5	1	4
6	6	3
7	3	3

	10
8	3
9	4

Rute manakah yang dapat menimbulkan ongkos terkecil?

Dari persoalan di atas, ada empat tahap (*stage*) yang harus dijalani untuk melakukan perjalanan dari kota (*state*) asal di 1 ke tujuan di 10. Untuk menyelesaikan persoalan ini, pertama yang harus diingat bahwa keputusan terbaik yang dibuat pada suatu *stage* belum tentu menghasilkan keputusan optimal secara menyeluruh. Berdasarkan strategi pemilihan ini, maka rute yang memberikan ongkos terkecil adalah 1->2->6->9->10 dengan ongkos total 13. Akan tetapi jika mau sedikit berkorban pada salah satu *stage* maka akan diperoleh penghematan lebih besar. Sebagai contoh: 1->4->6 lebih murah (secara keseluruhan) daripada 1->2->6.

Salah satu pendekatan yang dapat dilakukan untuk menyelesaikan persoalan ini adalah dengan menggunakan cara coba-coba (*trial and error*). Akan tetapi, jumlah rute yang dapat dilalui cukup banyak (ada 18) sehingga menghitung ongkos total untuk masing-masing rute itu akan sangat membosankan. Dengan programa dinamis, persoalan itu dapat diselesaikan secara lebih sederhana.

Pemecahan persoalan dengan program dinamis ini dimulai dengan mengambil bagian kecil dari suatu persoalan dan mencari solusi optimalnya. Kemudian, bagian persoalan itu diperluas sedikit demi sedikit, dan dicari solusi optimalnya yang baru. Demikian seterusnya hingga persoalan asal terpecahkan secara lengkap.

10.2. Karakteristik Persoalan Programa Dinamis

Persoalan *salesman* pada ilustrasi di atas merupakan contoh model dasar dari persoalan programa dinamis. Persoalan tersebut dirancang untuk memberikan suatu interpretasi secara fisik dari struktur yang abstrak dari persoalan programa dinamis. Oleh karena itu, salah satu cara untuk mengenal situasi yang dapat diformulasikan sebagai persoalan programa dinamis ini adalah dengan memperhatikan bahwa struktur dasar persoalan programa dinamis ini merupakan analogi dari persoalan *salesman* diatas.

Berikut diberikan beberapa gambaran dasar yang menandai persoalan programa dinamis:

1. Persoalan dapat dibagi menjadi beberapa tahap (*stage*), yang pada masing-masing *stage* diperlukan adanya suatu keputusan.
2. Masing-masing *stage* terdiri atas sejumlah *state* yang berhubungan dengan *stage* yang bersangkutan.
3. Hasil dari keputusan yang diambil pada setiap *stage* ditransformasikan dari *state* yang bersangkutan ke *state* yang berikutnya pula.
4. Keputusan terbaik pada suatu *stage* bersifat independen terhadap keputusan yang dilakukan pada *stage* sebelumnya.
5. Prosedur pemecahan persoalan dimulai dengan mendapatkan cara (keputusan) terbaik untuk setiap *state* dari *stage* terakhir.
6. Ada suatu hubungan timbal-balik yang mengidentifikasi keputusan terbaik untuk setiap *state* pada *stage* n, berdasarkan keputusan terbaik untuk setiap *state* dan *stage* (n + 1). Pada ilustrasi di atas hubungan ini adalah:

   $$fn^*(s) = \min_{x_n} \{ c_{sx_n} + f_n + 1 * (x_n) \}$$

   Oleh karena itu, untuk mendapat keputusan terbaik jika akan bergerak dari *state* s pada *stage* n, terlebih dahulu harus didapatkan nilai terbaik dari x_n pada *stage* (n + 1). Dalam hal ini ditetapkanlah:

   a. Variabel (atau vektor) x_n sebagai variabel keputusan pada *stage* n (n = 1, 2, ..., N).
   b. $fn(s, x_n)$ sebagai nilai fungsi tujuan yang akan dimaksimalkan atau diminimalkan, dengan catatan bahwa sistem akan berawal di *state* s pada *stage* n dan x_n telah terpilih sehingga $f(s, x_n) = c_{sx_n} + f_n + 1 * (x_n)$.
   c. $f_n^*(s)$ sebagai nilai maksimal/minimal dari $f_n(s, x_n)$ untuk seluruh nilai x_n yang mungkin.

   Maka bentuk hubungan timbal-baliknya adalah:

   $$f_n^*(s) = \underset{x_n \quad x_n}{\text{maks/min}} \{ f_n (s, x_n) \}$$

7. Dengan menggunakan hubungan timbal-balik ini, prosedur penyelesaian persoalan bergerak mundur *stage* demi *stage*, pada setiap *stage* berusaha diperoleh keputusan optimal untuk masing-masing *state* hingga akhirnya diperoleh keputusan optimal yang menyeluruh, mulai dari *stage* awal.

10.3. Programa Dinamis Deterministik

Pada bagian ini akan dikemukakan pendekatan programa dinamis sebagai persoalan deterministik, di mana *state* pada *stage* berikutnya sepenuhnya ditentukan oleh *state* dan keputusan pada *stage* saat ini.

Suatu cara untuk mengategorikan persoalan programa deterministik ini adalah dengan melihat bentuk fungsi tujuan. Sebagai contoh, fungsi tujuannya mungkin meminimalkan jumlah kontribusi dari masing-masing *stage* atau dapat pula memaksimalkannya atau meminimalkannya hasil perkaliannya, dan sebagainya. Cara pengategorian yang lain didasarkan pada keadaan dari kumpulan (set) *state* pada suatu *stage*. Artinya apakah *state* s_n dapat direpresentasikan sebagai variabel state diskrit atau kontinu, atau mungkin diperlukan suatu vektor *state* (lebih dari satu variabel).

10.4. Programa Dinamis Probabilistik

Berbeda dengan programa dinamis deterministik, pada programa dinamis probabilistik ini *stage* berikutnya tidak dapat seluruhnya ditentukan oleh *state* dan keputusan pada *stage* saat ini, tetapi ada suatu distribusi kemungkinan mengenai apa yang akan terjadi. Namun, distribusi kemungkinan ini masih seluruhnya ditentukan oleh *state* dan keputusan pada *stage* saat ini.

Akibat staruktur probabilistik ini, maka hubungan antara $f_n(s_n, x_n)$ dengan $f_{n+1}(s_{n+1})$ menjadi lebih rumit daripada untuk programa dinamis deterministik. Bentuk yang tepat untuk hubungan ini akan tergantung pada bentuk fungsi tujuan secara keseluruhan.

Sebagai ilustrasi, misalkan tujuannya adalah meminimalkan ekspektasi jumalh kontribusi dari masing-masing *stage*. Pada kasus ini $f_n(s_n, x_n)$ menyatakan ekspektasi jumlah minimal dari *stage* n ke muka, berdasarkan state s_n dan keputusan x_n pada *stage* n maka:

$$f_n(s_n, x_n) = \sum_{i=1}^{N} p_i \left[c_i + f^*_{n+1}(i)\right]$$

Dengan

$$f^*_{n+1}\,(sn+1) = \min_{x_{n+1}} f_{n+i}\,(s_{n+1}, x_{n+1})$$

di mana minimasi ini diambil dari nilai-nilai x_{n+1} yang fisibel.

11. Teori Probabilitas

11.1. Ruang Sampel dan Peristiwa

Untuk dapat mendalami teori probabilitas, pertama-tama perlu penguraian secara menyeluruh dari setiap kemungkinan *outcome* yang dapat terjadi dari suatu eksperimen. Setiap kemungkinan *outcome* ini didefinisikan sebagai titik sampel (elemen), sementara semua titik sampel ini didefinisikan sebagai ruang sampel.

Apabila suatu ruang sampel (biasa dinotasikan sebagai s) memiliki jumlah elemen yang terbatas, maka penulisan ruang sampel tersebut dapat dilakukan dengan mencantumkan seluruh elemen dari ruang sampel tersebut. Sebagai contoh, suatu ruang sampel s yang merupakan semua *outcome* yang mungkin terjadi dari pelemparan sebuah mata uang dapat dituliskan s = {H, T}. Untuk ruang sampel yang mengandung banyak elemen ataupun yang jumleh elemennya tidak terbatas, penulisannya diuraikan oleh suatu pernyataan. Sebagai contoh, suatu eksperimen yang meneliti lebih dari 1 juta penduduk suatu kota dituliskan: s= {x | x suatu kota dengan penduduk lebih dari 1 juta}.

Contoh:

Misalkan dari suatu proses produksi diambil 3 buah item secara acak untk diperiksa, apakah suatu item dalam keadaan aik (B) atau rusak (R). Ruang sampel yang memberikan informasi secara rinci dalah:

s = {BBB, BBR, BRB, RBB, RBR, RRB, RRR}

Pada setiap eksperimen mungkin kit hanya tertarik pada terjadinya suatu elemen yang spesifik ddlam ruang sampel, yang dikenal sebagai *event* (peristiwa/kejadian). Sebagai contoh, apabila kita tertarik pada *event* A yang merupakan kejadian di mana item yang rusak lebih besar dari 1, maka *event* A dituliskan:

A = {BRR, RBR, RRB, RRR}

Dalam hal ini, setiap *event* harus mengandung satu atau lebih titik sampel dan merupakan kumpulan bagian dari ruang sampel. Dengan kata lain, suatu *event* merupakan subset dari ruang sampel.

11.2. Probabilitas Suatu Event dan Hukum-hukumnya

Untuk menggambarkan sidat terjadinya suatu *event* dari probabilitas tertentu, teori matematis yang berhubungan dengan probabilitas untuk ruang sampel yang terbatas menyediakan kumpulan angka yang disbut *bobot* (*weight*) yang berkisar antara 0 dan 1. Jumlah bobot semua titik sampel (elemen) dalam suatu ruang sampel diatur sama dengan 1. Apabila suatu titik sampel tertentu memberikan keyakinan kepada kita akan terjadi dalam menjalankan suatu eksperimen, maka titik sampel tersebut diberi bobot mendekati 1. Sebaliknya, apabila suatu titik sampel hampir tidak mungkin terjadi, maka bobotnya mendekati nilai 0.

Untuk mengetahui besar kemungkinan (probabilitas) terjadinya suatu *event* A, maka semua bobot dari titik sampel yang merupakan elemen dari *event* A harus dijumlahkan. Hasil penjumlahan ini sering disebut sebagai *ukuran* dari A atau probabilitas dari *event* A yang dinotasikan sebagai P (A). Dengan kata lain, probabilitas terjadinya *event* A adalah jumlah bobot dari semua titik sampel yang terdapat pada *event* A sehingga:

$0 \leq P(A) \leq 1$

$P(\phi) = 0$

$P(S) = 1$

Apabila suatu eksperimen menghasilkan N *outcome* yang bersifat *equallylikely*, di mana n *outcome* merupakan elemen dari *event* A, maka probabilitas terjadinya *event* A adalah : $P(A) = n/N$.

Sering probabilitas suatu *event* dihitung berdasarkan probabilitas *event* lainnya yang sudah diketahui. Hal ini dapat dilakukan apabila *event* tersebut merupakan gabungan atau merupakan komplemen dari *event* yang lain. Jelasnya, untuk dua buah *event* A dan B, maka

$P(A \cup B) = P(A) + P(B) - P(A \cap B)$

Hal ini dikenal sebagai hukum penjumlahan

11.3. Probabilitas Terkondisi

Probabilitas terjadinya *event* B apabila sudah terjadi event A disebut probabilitas terkondisi, dan dinotasikan sebagai P (B | A) dan dibaca: probabilitas terjadinya *event* B di mana *event* A sudah terjadi. Besarnya probabilitas tersebut ditentukan dengan rumus:

$$P(B \mid A) = \frac{P(A \cap B)}{P(A)} \quad \text{Dengan } P(A) \geq 0$$

Sehingga $P(A \cap B) = P(A)\, P(B \mid A)$

Apabila A dan B dalah *events* yang bersifat independen, dimana P(A) > 0 dan P(B) > 0, maka P (A ∩ B) = P(A) . sehingga probabilitas terkondisinya adalah:

$$P(B \mid A) = \frac{P(A \cap B)}{P(A)} = \frac{P(A) \,.\, P(B)}{P(A)} = P(B)$$

Sebagai analogi diperoleh: $P(B \mid A) = \frac{P(A \cap B)}{P(A)}$

Jika A dan B adalah *events* yang independen, maka P (A | B) = P(A)

11.4. Variabel Random

Variabel random (acak) adalah suatu fungsi yang harganya merupakan bilangan riil dan ditentukan oleh setiap elemen dari suatu ruang sampel. Apabila ruang sampel berisi sejumlah elemen yang terbatas maka ruang sampel tersebut disebut sebagai ruang sampel diskrit, dan variabel randomnya disebut sebagai variabel random diskrit. Sebaliknya, apabila jumlah elemen pada ruang sampel itu tidak terbatas, maka ruang sampel tersebut disebut sebagai ruang sampel kontinu, dan variabel randomnya disebut sebagai variabel random kontinu. Dalam hal ini, variabel random diskrit akan mempresentarikan data yang dapat dihitung, sedangkan variabel random kontinu mempresentasikan data yang dapat diukur.

Nilai-nilai probabilitas variabel random itu akan membangun bentuk distribusi probabilitas tertentu, tergantung pada macam percobaan dan karakter variabel randomnya. Pada dasarnya, distribusi probabilitas dari variabel random ini dikategorikan sebagai distribusi probabilitas diskrit dan distribusi probabilitas kontinu.

11.5. Proses Stochastic

Perhatikan suatu eksperimen E yang dispesifikasikan oleh *outcome* s yang membentuak suatu ruang (*space*) S oleh subset tertentu dari S yang disebut *event*, dan oleh probabilitas dari *events* tersebut. Untuk setiap *outcome* s itu ada suatu fungi waktu x(t,s) yang harganya bisa berupa bilangan riil atau bilangan kompleks. Maka kita telah membuat satu "keluarga" fungsi, yang masing-masing fungsi berlaku untuk masing-masing s. Keluarga fungsi disebut proses *stochastic*.

Proses *stochastic* ini dapat dipandang sebagai suatu fungsi dari dua variabel t dan s, di mana domain s adalah set S dan domain t adalah set bilangan riil. Untuk suatu *outcome* s_i teetentu, ekspresi $x(t, s_i)$ menyatakan suatu fungsi waktu tunggal, sedangkan untuk suatu harga t_i tertentu $x(t_i, s)$ adalah suatu besaran yang bergantung pada s, dan disebut sebagai variabel random. Akhirnya, $x(t_i, s_i)$ dalah suatu bilangan murni.

Kita dapat menggunakan notasi x(t) untuk menyatakan suatu proses *stochastic* yang menghilangkan ketergantungannya pada s, maka x(t) ini menyatakan empat hal yang berbeda, yaitu:

a. Suatu keluarga dari fungsi waktu (t dan s berupa variabel).
b. Suatu fungsi waktu tunggal (t variabel, s tetap).
c. Suatu variabel random (t tetap, s variabel).
d. Suatu bilangan tunggal (t tetap, s tetap).

12. Proses Keputusan Markov

12.1. Ilustrasi Persoalan Keputusan Markov

Untuk dapat memahami model penyelesaiandari persoalan keputusan markov, berikut ini dikemukakan sebuah ilustrasi dari persoalan keputusan yang sangat sederhana, sebagai berikut:

Kondisi sebuah mesinyang digunakan dalam suatu proses produksi diketahui menurun dengan cepat, baik dalam kualitas maupun *output*-nya. Karena itu, terhadap mesin tersebut dilakukan pemeriksaan secara periodik, yaitu pada setiap akhir bulan. Setelah dilakukan serangkaian pemeriksaan, kondisi mesin ini dicatat dan diklasifikasikan ke dalam salah satu dari tiga keadaan (*state*) berikut ini:

State	Kondisi
1	Baik
2	Cukup
3	Rusak

Jika X_t adalah *state* mesin setelah dilakukan pemeriksaan pada akhir bulan ke-t, maka urut-urutan dari *state* $\{X_t\}$ ini dapat dipandang sebagai suatu proses *stochastic*. Misalkan probabilitas transisi selama periode 1 bulan dari suatu *state* ke *state* lainnya adalah sebagai berikut:

$$\text{State bulan yad}$$

$$\text{State pada bulan ini} \quad \begin{array}{c c} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 0,2 & 0,5 & 0,3 \\ 0 & 0,5 & 0,5 \\ 0 & 0 & 1 \end{bmatrix} \end{array} = p^1$$

Dari matriks transisi p^1 di atas jelas bahwa sekali mesin itu rusak (*state* 3), maka akan tetap rusak. Kondisi di atas akan berubah apabila terhadap mesin itu dilakukan perbaikan (*overhaul*). Jadi, jika *overhaul* tidak dilakukan, maka probabilitas transisinya akan tetap seperti pada p^1. Tetapi jika dilakukan *overhaul*, maka matriks transisinya adalah p^2, yaitu sebagai berikut:

$$p^2 = \begin{array}{c c} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 0,3 & 0,6 & 0,1 \\ 0,1 & 0,6 & 0,3 \\ 0,05 & 0,4 & 0,55 \end{bmatrix} \end{array}$$

Struktur ongkos (penerimaan atau pengeluaran) selama periode 1 bulan akan tergantung pada *state* masing-masing matriks transisi. Jika diketahui bahwa struktur ongkos apabila tidak dilakukan *overhaul* adalah R^1, dan struktur ongkos apabila dilakukan *overhaul* adalah R^2, (dalam satuan juta rupiah), di mana:

$$R^1 = || r_{ij}^1 || = \begin{array}{c c} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 1 \\ 2 \end{array} & \begin{bmatrix} 7 & 6 & 3 \\ 0 & 5 & 1 \end{bmatrix} \end{array}$$

$$\begin{array}{cccc} 3 & 0 & 0 & -1 \end{array}$$

dan

$$R^2 = || r_{ij}^2 || = \begin{array}{c c} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 6 & 5 & -1 \\ 7 & 4 & 0 \\ 6 & 3 & -2 \end{bmatrix} \end{array}$$

Keputusan apakah yang sebaiknya dilakukan (melakukan *overhaul* apa tidak)?

Perhatikan bahwa elemen-elemen r_{ij}^2 dan R^2 ini telah memperhitungkan ongkos perbaikan (*overhaul*). Sebagai contoh, jika sistem berada pada *state* 1, dan tetap pada *state* 1 selama 1 bulan yang akan datang, maka pendapatan yang akan diperoleh adalah $r_{11}^2 = 6$, bandingkan dengan $r_{11}^1 = 7$ jika overhaul tidak dilakukan.

Untuk menentukan jenis persoalan keputusan yang harus diselesaikan, pertama-tama harus diketahui apakah mesin-mesin ini akan terus dioperasikan dalam beberapa bulan tertentu yang lamanya terbatas, atau tidak didefinisikan kapan batas pengoperasiannya. Dengan kata lain, apakah persoalan keputusan ini mempunyai *stage* yang terbatas atau *stage* yang tidak terbatas. Untuk kedua jenis persoalan ini keputusan yang harus dilakukan adalah sama, yaitu menentukan apakah *overhaul* itu perlu dilakukan atas nilai ekspektasi pendapatan yang maksimal.

Jenis persoalan lainnya adalah pengevaluasian ekspektasi pendapatan sebagai hasil dari suatu tindakan yang telah ditetapkan apabila suatu *state* dari sistem terjadi. Misalnya, apabila diputuskan untuk melakukan *overhaul* apabila mesin dalam kondisi rusak (*state 3*). Proses pembuatan keputusan untuk kasus ini dijelaskan oleh suatu *stationary policy*.

Perlu diketahui bahwa setiap *stationary policy* akan berkaitan dengan matriks transisi dan matriks ongkos yang berbeda yang umumnya dapat dibentuk dari matriks-matriks P^1, P^2, R1, dan R^2.

Sebagai contoh, *stationary policy* untuk melakukan *overhaul* hanya jika mesin dalam kondisi rusak (*state 3*), matriks transisi dan matriks ongkosnya adalah P dan R sebagai berikut:

$$P = \begin{bmatrix} 0{,}020 & 0{,}50 & 0{,}30 \\ 0 & 0{,}50 & 0{,}50 \\ 0{,}05 & 0{,}40 & 0{,}55 \end{bmatrix}, \quad R = \begin{bmatrix} 7 & 6 & 3 \\ 0 & 5 & 1 \\ 6 & 3 & -2 \end{bmatrix}$$

Matriks P dan R ini berbeda dari matriks P^1 dan R^1 hanya pada baris ketiga saja, yang diambil langsung dari P^2 dan R^2. Alasannya adalah karena P^2 dan R^2 adalah matriks-matriks yang dihasilkan apabila *overhaul* dilakukan pada setiap *state*.

12.2. Membangun Matriks Probabilitas Transisi

Untuk memperoleh matriks probabilitas transisi atau matriks P seperti pada contoh diatas, diperlukan pengamatan yang teliti terhadap kondisi sistem (yang diamati) pada satu periode ke periode berikutnya. Sebagai ilustrasi berikut ini dikemukakan suatu kasus yang berkenaan dengan perilaku langganan sabun deterjen. Misalkan di suatu daerah dipasarkan empat merek sabun deterjen A, B, C, dan D. Terhadap para pemakai deterjen di daerah tersebut telah dilakukan penelitian dengan menyebar daftar isian (kuesioner). Jumlah responden yang mengembalikan daftar isian tersebut ada 1.000 orang, dan diasumsikan bahwa ukuran sampel ini cukup representatif. Data yang diperoleh berupa jumlah pelanggan masing-masing merek, kemudian dicatat dan dinyatakan sebagai data jumlah pelanggan pada periode pertama. Berdasarkan pemikiran bahwa pelanggan dapat mengubah pilihannya dari satu merek ke merek lainnya (misalnya karena promosi khuus, persaingan harga, dan lain-lain), maka pada akhir periode dilakukan penelitian ulang.

Tabel berikut ini menunjukkan data jumlah pelanggan masing-masing merek pada periode pertama, perubahan jumlah pelanggan yang terjadi pada suatu periode, dan jumlah pelanggan pada periode kedua. Periode dapat berupa bulan, triwulan, semester, tahun, dan sebagainya.

Merek	Jumlah pelanggan periode pertama	Perubahan selama periode		Jumlah pelanggan periode kedua
		Pindah ke	Pindah dari	
A	220	50	45	225
B	300	60	70	290
C	230	25	25	230
D	250	40	35	225
Total	1.000	175	175	1.000

Tabel tersebut memberikan informasi bahwa pada awal periode, jumlah pelanggan merek A ada 220 orang. Selama periode berlangsung, terjadi perubahan, yaitu responden yang semula tidak memilih merek A beralih ke merek A sebanyak 50 orang. Sebaliknya yang dari semula memilih merek A berubah menjadi pelanggan merek lain sebanyak 45 orang. Dengan demikian, pada akhir periode atau awal periode kedua, jumlah pelanggan merek A sebanyak 225 orang (220 + 50 – 45). Begitu juga merek-merek yang lainnya. Sayang sekali bahwa data tersebut tidak menjelaskan tentang dari merek mana saja ke-50 pelanggan baru yang pindah ke merek A itu, dan pindah ke merek mana saja ke-45 pelanggan meninggalkan merek A itu.

Jika kemudian penelitian dilanjutkan dan diperoleh data rinci mengenai perubahan pelanggan untuk masing-masing merek, maka matriks probabilitas transisinya dapat ditentukan. Misalkan data tersebut adalah seperti pada tabel berikut ini:

Merek	Jml pelanggan periode pertama	Tambahan dari merek				Pengurangan ke merek				Jml pelanggan periode kedua
		A	B	C	D	A	B	C	D	
A	220	0	40	0	10	0	20	10	15	225
B	300	20	0	25	15	40	0	5	25	290
C	230	10	5	0	0	0	25	0	0	230
D	250	15	25	0	0	10	15	10	0	155
Total	1.000									1.000

Data di atas memberikan informasi bahwa dari sejumlah 220 pelanggan merek A pada periode pertama, telah beralih menjadi pelanggan merek B sebanyak 20 orang, menjadi pelanggan merek C sebanyak 10 orang, dan menjadi pelanggan merek D sebanyak 15 orang. Maka jumlah pelanggan pelanggan yang pada periode pertama memilih merek A dan pada periode kedua masih tetap memilih merek A (bukan pelanggan baru) adalah sebanyak (220 - 20 - 10 - 15) = 175 orang. Dengan kata lain: probabilitas bahwa pelanggan merek A pada periode pertama tetap menjadi pelanggan merek A, pada periode kedua adalah sebesar 175/220 = 0,796. Selanjutnya, probabilitas bahwa pelanggan merek A pada periode pertama yang berubah menjadi pelanggan merek B pada periode kedua adalah sebesar 20/220 = 0,091.

Apabila perhitungan dilanjutkan maka akan diperoleh matriks probabilitas transisi sebagai berikut:

$$\begin{array}{c} A \\ B \\ C \\ D \end{array}\begin{bmatrix} 175/220 = 0{,}796 & 20/220 = 0{,}091 & 10/220 = 0{,}046 & 15/220 = 0{,}067 \\ 40/300 = 0{,}133 & 230/300 = 0{,}767 & 5/300 = 0{,}017 & 25/300 = 0{,}083 \\ 0/230 = 0 & 25/230 = 0{,}109 & 205/230 = 0{,}891 & 0/230 = 0 \\ 10/250 = 0{,}040 & 15/250 = 0{,}060 & 10/250 = 0{,}040 & 215/250 = 0{,}860 \end{bmatrix}$$

atau dengan singkat ditulis sebagai berikut:

$$P = \begin{bmatrix} 0{,}796 & 0{,}091 & 0{,}046 & 0{,}067 \\ 0{,}133 & 0{,}767 & 0{,}017 & 0{,}083 \\ 0 & 0{,}109 & 0{,}891 & 0 \\ 0{,}040 & 0{,}060 & 0{,}040 & 0{,}860 \end{bmatrix}$$

12.3. Model Programa Dinamis dengan Stage Terbatas

Misalkan , si pengambil keputusan dari persoalan perbaikan mesin diatas merencanakan untuk menghentikan pengoperasian mesin itu dalam N bulan. Maka persoalannya adalah menentukan tindakan optimal (*overhaul* atau tidak) untuk masing-masing bulan selama waktu perencanaan. Optimalitas di sini didefinisikan sedemikian sehingga dia dapat memperoleh ekspektasi pendapatan tertinggi pada akhir bulan ke-N.

Penyelesaian persoalan dengan *stage* terbatas dapat digeneralisasi dalam dua cara. Pertama, fungsi probabilitas transisi dan fungsi pendapatannya tidak perlu sama untuk setiap bulannya. Kedua, si pengambil keputusan itu dapat menggunakan faktor potongan terhadap ekspektasi pendapatan dari *stage* yang berurutan sehingga nilai f_1 (i) akan menyatakan nilai pada saat ini dari ekspektasi pendapatan pada seluruh *stage*.

12.4. Model Programa Dinamis dengan Stage Tidak Terbatas

Perilaku jangka panjang dari suatu proses Markov ditandai oleh ketidaktergantungannya pada *state* awal (*initial state*) dari sistemnya. Dalam hal ini, sistem tersebut dikatakan telah mencapay *steady state* (keadaan tetap).

Ada dua metode yang dapat digunakan untuk menyelesaikan persoalan dengan *stage* yang tidak terbatas ini. Metode pertama disebut metode enumerasi sempurna yang mengenumerasi seluruh *stationary policy*, hingga diperoleh solusi optimalnya. Metode ini hanya dapat digunakan apabila jumlah *stationary policy*-nya tidak terlalu besar sehingga masih dapat dihitung. Metode yang kedua adalah metode *policy iteration*, yang mampu mengurangi kesulitan perhitungan pada metode pertama. Metode ini umumnya bersifat efisien, dalam arti dapat mencapai solusi optimal dalam jumlah iterasi yang kecil. Namun kedua metode tersebut akan menghasilkan solusi optimal yang sama.

13. Teori Antrian

13.1. Struktur Dasar Model-Model Antrian

Proses-proses yang terjadi pada model antrian dapat dijelaskan sebagai berikut:

Unit-unit (langganan) yang memerlukan pelayanan yang diturunkan dari suatu sumber input memasuki sistem antrian dan ikut dalam antrian. Dalam waktu-waktu tertentu anggota antrian ini dipilih untuk dilayani. Pemilihan ini didasarkan pada suatu aturan tertentu yang disebut “disiplin pelayanan” atau *service dicipline*. Pelayanan yang diperlukan dilaksanakandengan suatu “mekanisme pelayanan” tertentu (*service mechanism*). Setelah itu unit-unit (langganan) tersebut meninggalkan sistem antrian.

1. Sumber Input

   Suatu karakteristik yang perlu diketahui dari sumber input ini adalah ukurannya (jumlahnya), yaitu jumlah total unit yang memerlukan pelayanan dari waktu ke waktu atau disebut jumlah total langganan potensial

2. Antrian

   Karakteristik suatu antrian ditentukan oleh jumlah unit maksimal yang boleh ada di dalam sistem. Antrian ini dikatakan terbatas atau tidak terbatas, tergantung pada apakah jumlah unitnya terbatas atau tidak terbatas.

3. Disiplin pelayanan

   Disiplin pelayanan berkaitan dengan cara memilih anggota antria yang akan dilayani. Sebagai contoh, displin pelayanan ini dapat berupa *first come-first served* (yang datang lebih dulu dilayani lebih dahulu), atau random, atau dapat pula berdasarkan prosedur prioritas tertentu. Jika tidak ada keterangan apa-apa tentang disiplin pelayanan ini, maka asumsi yang biasa digunakan adalah *first come-first served*.

4. Mekanisme pelayanan

Mekanisme pelayanan ini terdiri atas satu atau lebih fasilitas pelayanan masing-masing terdiri atas satu atau lebih saluran pelayanan paralel. Jika ada lebih dari satu fasilitas pelayanan, maka unit-unit yang memerlukan pelayanan akan dilayani oleh serangkaian fasilitas pelayanan ini (saluran pelayanan seri)

5. Proses antrian dasar

   Suatu garis penungguan tunggal (yang suatu saat bisa saja kosong) berbentuk di depan suatu fasilitas pelayanan tunggal, di mana ada satu atau beberapa pelayanan. Setiap unit (langganan) yang diturunkan oleh sutu sumber input dilayani oleh salah satu dari pelayan-pelayan yang ada, mungkin setelah unit itu menunggu dalam antrian (garis penungguan)

13.2. Terminologi dan Notasi

Terminologi dan notasi yang digunakan adalah sebagai berikut:

Keadaan sistem: jumlah langganan (unit) dalam sistem antrian

Panjang antrian: jumlah langganan (unit) yang menunggu pelayanan = keadaan sistem dikurangi jumlah unit yang sedang dilayani.

E_n : keadaan dimana ada n *calling unit* pada sistem antrian

$P_n(t)$: kemungkinan bahwa tepat ada n *calling unit* pada sistem antrian pada saat t

S : jumlah pelayan (untuk saluran pelayanan paralel) pada sistem antrian

λ_n : tingkat kedatangan rata-rata (ekspektasi jumlah kedatangan per satuan waktu) dari *calling unit* baru jika ada n unit dalam sistem

μ_n : tingkat pelayanan rata-rata (ekspektasi jumlah unityang dapat selesai dilayani per satuan waktu) jika ada n unit dalam sistem

Jika λ_n adalah konstan untuk semua n, maka dapat ditulis sebagai λ. Jika μ_n konstan untuk semua $n \geq 1$, maka dapat ditulis sebagai μ. Disini $\mu_n = S\mu$ jika $n \geq S$ sehingga seluruh pelayan (sejumlah S) sibuk. Dalam hal ini $1/\lambda$ menyatakan ekspektasi waktu di antara kedatangan, sedangkan $1/\mu$ menyatakan ekspektasi waktu pelayanan.

$\rho = \lambda / S\mu$ adalah faktor penggunaan (utilisasi) untuk fasilitas pelayanan, yaitu ekspektasi perbandingan dari waktu sibuk para pelayan

Jika suatu sistem antrian telah mulai berjalan, keadaan sistem (jumlah unit dalam sistem) akan sangat dipengaruhi oleh *state* (keadaan) awal dan waktu yang telah dilalui. Dalam keadaan seperti ini, sistem dikatakan dalam kondisi transien. Tetapi lama kelamaan keadaan sistem akan independen terhadap *state* awal tersebut, dan juga terdapat waktu yang dilaluinya. Keadaan sistem seperti ini dikatakan berada dalm kondisi *steady state*. Teori antrian cenderung memusatkan pada kondisi *steady state* karena kondisi transien lebih sukar dianalisis.

Notasi-notasi berikut ini digunakan untuk sistem dalam kondisi *steady state*:

Pn : kemungkinan bahwa tepat ada n *calling unit* dalam sistem antrian.

L : ekspektasi panjang garis.

L_q : ekspektasi panjang antrian.

W : ekspektasi waktu menunggu dalam sistem (termasuk waktu pelayanan).

W_q : ekspektasi waktu menunggu dalam antrian (tidak termasuk waktu pelayanan).

Hubungan antara L dan W

Asumsikan bahwa λ_n adalah konstan untuk semua n sehingga cukup ditulis λ. Maka dalam proses antrian yang *steady state* didapat:

$$L = \lambda W$$

$$L_q = \lambda W_q$$

Sekarang asumsikan bahwa waktu pelayanan rata-rata dalah konstan untuk semua n ≥ 1 sehingga cukup ditulis sebagai 1/μ, maka:

$$W = W_q + 1/\mu$$

kalikan dengan λ, didapat:

$$L = L_q + P$$

13.3. Contoh Sistem Antrian

Salah satu kelas sistem antrian yang paling sering dijumpai dalam keadaan sehari-hari adalah sistem antrian komersial, di mana langganan memperoleh pelayanan dari organisasi-organisasi komerisl. Bebrapa dari sistem ini menyangkut pelayanan dari orang ke orang pada suatu lokasi yang tetap seperti misalnya tempat potong rambut, bank, kafetaria, pompa bensin, dan lain-lain.

Kelas lain yang juga penting adalah sistem pelayanan transportasi. Beberapa dari sitem ini seperti langganannya berupa kendaraan/alat angkut. Contoh: mobil-mobil yang menunggu di gerbang tol atau *traffic light*, truk yang menunggu untuk dimuati atau di bongkar, pesawat yang menunggu untuk mendarat atau lepas landas dari suatu landasan. Contoh yang lebih spesifik dari sitem semacam ini adalah tempat parkir, dalam hal ini mobil-mobil sebagai langganan dan areal parkir sebagai pelayan. Di sini tidak ada antrian karena langganan yang datang akan pergi ke tempat parkir lain jika tempat parkir telah penuh.

13.4. Proses Birth and Death

Kebanyakan model dasar antrian menganggap bahwa *input* (unit kedatangan) dan *output* (*leaving unit*) dari sistem antrian terjadi menurut proses *birth-death* (kelahiran-kematian). Kelahiran adalah kedatangan *calling unit* yang baruu dalam sistem antrian, sedngakan kematian adalah keberangkatan unit yang telah dilayani. Proses kelahiran dan kematian ini terjadi secara random yang rata-rata terjadinya hanya tergantiung pada keadaan yang sedang berlangsung (*current state*) dari sistem (jumlah *calling unit* dalam sistem antrian.)