**OPERATION RESEARCH (PENELITIAN OPERASIONAL)**

1. **Gambaran Umum Penelitian Operasional**
   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. **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. **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.

* + 1. 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.

* + 1. 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.

* + 1. 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.

* + 1. 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. **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. **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.

1. **Programa Linier**
   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 x1 dan x2 sebagai jumlah unit dari produk 1 dan produk 2 yang diproduksi per menit. Dengan demikian x1 dan x2 menjadi variabel keputusan dari model ini, dan tujuannya adalah memilih harga-harga x1 dan x2 sehingga diperoleh nilai maksimum dari:

z = 3x1 + 5x2

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 x1 ≤ 4. Dengan cara yang sama, pabrik 2 memiliki pembatas 2x2 ≤ 12. Persentase kapasitas pabrik 3 digunakan dengan cara memilih x1 dan x2 sebagai produk-produk baru tersebut sehingga ukuran produksinya adalah 3x1 + 2x2. Karena itu secara matematis pembatas dari pabrik 3 ini adalah 3x1 + 2x2 ≤ 18. Karena ukuran produksi ini tidak mungkin berharga negatif, maka variabel-variabel keputusan ini harus dibatasi sehingga berharga nonnegatif dengan x1 ≥ 0 dan x2 ≥ 0.

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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | z | = 3x1 + 5x2 | |  |  |
|  |  | x1 |  | ≤ 4 |  |  |
|  |  |  | 2x2 | ≤ 12 |  |  |
|  |  | 3x1 | + 2x2 | ≤ 18 |  |  |
|  |  |  |  |  |  |  |
|  |  | x1 | ≥ 0, x2 | ≥ 0 |  |  |

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 xi, pembatasan tanda akan menunjukkan apakah xi harus nonnegatif (xi ≥ 0) atau xi tidak terbatas tanda.
  1. **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:

* + 1. Fungsi yang dimaksimalkan, yaitu c1 x1 + c2 x2 + … + cn xn, disebut sebagai fungsi tujuan.
    2. Pembatas-pembatas atau konstrain.
    3. Sebanyak m buah konstrain pertama sering disebut sebagai konstrain fungsional atau pembatas teknologis.
    4. Pembatas xj ≥ 0 disebut sebagai konstrain nonnegatif.
    5. Variabel xj adalah variabel keputusan.
    6. Konstanta-konstanta aij, bi, cj 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 = c1 x1 + c2 x2 + … + cn xn

* + 1. Beberapa konstrain fungsionalnya memiliki ketidaksamaan dalam bentuk lebih besar atau sama dengan.

Contoh : ai1 x1 + ai2 x2 + … + ain xn ≥ bi

untuk beberapa harga i

* + 1. Beberapa konstrain fungsionalnya memiliki bentuk persamaan

Contoh : ai1 x1 + ai2 x2 + … + ain xn = bi

untuk beberapa harga i

* + 1. Menghilangkan konstrain nonnegatif untuk beberapa variabel keputusan.

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

* 1. **Asumsi dalam Model Programa Linier**

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

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

* + 1. Asumsi kepastian (*certainty*)
    2. Setiap parameter, yaitu koefisien fungsi tujuan, ruas kanan, dan koefisien teknologis diasumsikan dapat diketahui secara pasti.

1. **Teknik Pemecahan Model Programa Linier**
   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 x1, dan x2 sebagai sumbu-sumbunya. Langkah pertama adalah mengidentifikasi harga-harga (x1, x2) yang memenuhi pembatas-pembatas yang ada dengan cara menggambarkan garis-garis yang harus membatasi daerah harga-harga yang diperbolehkan. Ingat bahwa pembatas nonnegatif x1 ≥ 0 dan x2 ≥ 0 akan menyebabkan (x1, x2) harus berada pada sisi positif dari sumbu-sumbunya (pada kuadran I). Setelah itu, perhatikan bahwa pembatas x1 ≤ 4 berarti bahwa (x1, x2) tidak boleh berada di sebelah kanan garis x1 = 4. Demikian pula dalam menggambarkan pembatas 2x2 ≤ 12 berarti bahwa (x1, x2) tidak boleh berada di atas garis 2x2 = 12 atau (x2 = 6). Untuk menggambarkan pembatas terakhir 3x1 + 2x2 ≤ 18 adalah dengan cara menentukan titik (x1, x2) yang memenuhi garis 3x1 + 2x2 = 18. Perlu diingat bahwa titik-titik yang memenuhi pembatas 3x1 + 2x2 ≤ 18 adalah titik-titik di bawah garis 3x1 + 2x2 = 18. Setelah seluruh pembatas digambarkan maka akan diperoleh daerah berlakunya harga-harga (x1, x2) seperti pada grafik berikut.

Arah panah pada setiap garis pembatas menunjukkan arah berlakunya harga (x1, x2) pada masing-masing pembatas. Karena kita harus mendapatkan harga (x1, x2) 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 = 3x1 + 5x2. Caranya adalah dengan menggambarkan sebuah garis z yang telah mempunyai koefisien arah :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tg α = | x2 | = | - | 3 | (berharga negatif karena α berada pada kuadran II) |
| x1 | 5 |

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 (x1, x2) yang memaksimalkan z = 3x1 + 5x2.

Harga (x1, x2) pada titik optimal diperoleh dengan cara menentukan titik potong garis ED (pembatas ke-2) dengan garis CD (pembatas ke-3) sebagai berikut :

2x2 = 12

3x1 + 2x2 = 18 –

* 3x1 = - 6

sehingga diperoleh harga-harga x1 = 2 dan x2 = 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.

* + 1. 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 :

x1 = lamanya promosi dalam acara hiburan

x2 = lamanya promosi dalam acara olahraga

formulasi persoalan :

minimalkan z = 5x1 + 10x2

berdasarkan

7x1 + 2x2 ≥ 28

2x1 + 12x2 ≥ 24

x1 ≥ 0, x2 ≥ 0

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

Contoh 1:

x1 + 2x2 ≤ 6

kita tambahkan *slack* S1 ≥ 0 pada ruas kiri sehingga diperoleh persamaan:

x1 + 2x2 +S1 = 6, S1 ≥ 0

Jika pembatas di atas menyatakan batas penggunaan suatu sumber, maka S1 akan menyatakan banyaknya sumber yang tidak terpakai.

Contoh 2:

3x1 + 2x2 – 3x3 ≥ 5

karena ruas kirinya tidak lebih kecil dari ruas kanan, maka harus dikurangkan variabel S2 ≥ 0 pada ruas kiri sehingga diperoleh persamaan :

3x1 + 2x2 – 3x3 – S2 = 5, S2 ≥ 0

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

Contoh:

2x1 – 3x2 – 7x3 = -5, secara matematis adalah sama dengan -2x1 + 3x2 + 7x3 = 5

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

Contoh:

2 < 4 adalah sama dengan -2 > -4

2x1 – x2 ≤ -5 adalah sama dengan -2x1 + x2 ≥ 5

* + - 1. Pembatas dengan ketidaksamaan yang ruas kirinya berada dalam tanda mutlak dapat diubah menjadi dua ketidaksamaan.

Contoh 1:

untuk b ≥ 0, |a1x1 + a2x2| ≤ b adalah sama dengan a1x1 + a2x2 ≤ b

dan a1x1 + a2x2 ≥ -b.

Contoh 2:

untuk q ≥ 0, |p1x1 + p2x2| ≥ q adalah sama dengan p1x1 + p2x2 ≥ q

atau p1x1 + p2x2 ≤ -q.

* + 1. Variabel

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

yi = yi’ – yi” dimana yi’ dan yi” ≥ 0

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

* + 1. 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 = 5x1 + 2x2 + 3x3

secara matematis adalah sama dengan :

minimalkan (-z) = -5x1 – 2x2 – 3x3

* 1. **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 = c1x1 + c2x2 + … + cnxn

Berdasarkan :

a11x1 + a12x2 + … + a1nxn = b1

a21x1 + a22x2 + … + a2nxn = b2

.

.

.

am1x1 + am2x2 + … + amnxn = bm

xi ≥ 0 (i = 1, 2, …, n)

Jika didefinisikan :

a11 a12 … a1n x1 b1

a11 a12 … a1n x2 b1

A = . ; x = ; b =

.

am1 am2 … amn xn bm

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 x2 lebih besar daripada x1, dan fungsi tujuannya maksimasi, maka solusi akan bergerak searah dengan peningkatan x2 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 = 3x1 + 5x2 + OS1 + OS2 + OS3

Berdasarkan pembatas:

x1 + S1 = 4

2x2 + S2 = 12

3x2 + 2x2 + S3 = 18

X1, x2, S1, S2, S3 ≥ 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 | n | = n ! / [ (n - m) ! 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.

* 1. **Kasus Khusus dalam Penggunaan Algoritma Simpleks**
     1. Degenerasi
     2. Solusi Optimal Banyak
     3. Solusi tak Terbatas
  2. **Menyelesaikan Persoalan lP dengan Pembatas Bertanda ≥ dan atau =**

1. **Teori Dualitas dan Analisis Kepekaan**
   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 + .... + cnxn

Berdasarkan pembatas :

a11x1 + a12x2 + ... +a1nxn <= b1

a21x1 + a22x2 + ... +a2nxn <= b2

.

.

.

am1x1 + am2x2 + ... amnXn <= bm

x1.x2, ...., xn <= 0

Dual :

Minimalkan : w = b1y1 + b2y2 + .... + bmym

Berdasarkan pembatas:

a11y1 + a21y2 + .... + am1ym >= c1

a12y1 + a22y2 + .... + am2ym >= c2

.

.

.

a1ny1 + a2ny2 + .... + amnymn >= cn

y1, y2, ... , ym >= 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.
  1. **Hubungan Primal-Dual**
  2. **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
   1. **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

* + 1. *Entering variable* (kondisi optimalitas)
       1. 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.
       2. Untuk persoalan minimasi, *entering variable* adalah variabel dengan rasio terkecil, sedangkan untuk persoalan maksimasi, *entering variable* adalah variabel dengan rasio absolut terkecil.
  1. **Beberapa Perumusan Penting**
  2. **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.

* 1. **Shadow Prices**

Untuk mendefinisikan kegunaan konsep *shadow price* ini, misalkan kita memiliki persoalan maksimasi dengan m pembatas dimana bi 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 bi sebesar 1 unit, yaitu dari bi menjadi (bi + 1)

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

* 1. **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 CBVB-1) adalah dual fisibel.

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

* + - 1. Perubahan koefisien fungsi tujuan pada variabel nonbasis
      2. Perubahan kolom variabel nonbasis
      3. 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 CBVB-1) yaitu tetap dua fisibels

1. **Tipe-tipe Khusus Persoalan Programa Linier**
   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.
   1. **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 ≥ | m |  | n |
| Σai | = | Σ |
| i=1 |  | j=1 |

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).

* 1. **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 cij. Karena satu pekerjaan ditugaskan hanya pada satu mesin, maka *supply* yang dapat digunakan pada setiap sumber adalah 1 (atau ai = 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 bj = 1, untuk seluruh j). Jika ada suatu pekerjaan yang tidak dapat ditugaskan pada mesin tertentu, maka cij 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.

1. **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.
  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.
  1. **Persoalan Rute Terpendek**
  2. **Persoalan Rentang Pohon Minimum**
  3. **Persoalan Aliran Maksimal**

1. **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. **Simbol-simbol yang digunakan**

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

* + 1. Anak panah (*arrow*)/

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.

* + 1. 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.

* + 1. 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*.
  1. **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*.

* 1. **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) = TE(j) = 0

EF(i, j) = TS(i, j) + t(i, j)

= TE(i) + t(i, j)

* + 1. 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.

* 1. **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.

* 1. **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.

* 1. **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.

* 1. **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:

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

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

Tp = *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.

* 1. **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.

1. **Programa Bilangan Bulat**
   1. **Pendahuluan**
   2. **Programa Linier Relaksasi**
   3. **Memformulasikan Persoalan Programa Bilangan Bulat**
   4. **Menyelesaikan Persoalan IP Murni dengan Teknik Branch-and-Bound**
   5. **Menyelisaikan Persoalan IP Campuran dengan Teknik Branch-and-Bound**
   6. **Menyelesaikan Persoalan Ransel**
   7. **Menyelesaikan Persoalan IP dengan Metode Enumerasi Implisit**
   8. **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.

1. **Teori Permainan**
   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.

* 1. **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*).

* 1. **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.

* 1. **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;

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

m n

Σ xi = Σ yj = 1

i=1 j=1

x1, yj ≥ 0 untuk setiap i dan j

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | B | | |  |
|  | | Y1 | Y2 | ... | Yn |
| A | x1 | a11 | a12 |  |  |
| 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

* 1. **Solusi Grafis dari Permainan (2 x n) dan (m x 2)**
  2. **Solusi Permainan (m x n) dengan Programa Linier**

1. **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.

* 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 (cij) adalah sebagai berikut:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 2 | 3 | 4 |  |  | 5 | 6 | 7 |  |  | 8 | 9 |  |  | 10 |  |  |  |  |
| 1 | 2 | 4 | 3 |  | 2 | 7 | 4 | 6 |  | 5 | 1 | 4 |  | 8 | 3 |  |  |  |  |
|  |  |  |  |  | 3 | 3 | 2 | 4 |  | 6 | 6 | 3 |  | 9 | 4 |  |  |  |  |
|  |  |  |  |  | 4 | 4 | 1 | 5 |  | 7 | 3 | 3 |  |  |  |  |  |  |  |

Rute manakah yang dapat menimbulkan ongkos terkecil?

* 1. **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 { csxn + fn + 1 \* (xn) }

xn

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

Dalam hal ini ditetapkanlah:

* + - 1. Variabel (atau vektor) xn sebagai variabel keputusan pada *stage* n (n = 1, 2, ..., N).
      2. fn(s, xn) sebagai nilai fungsi tujuan yang akan dimaksimalkan atau diminimalkan, dengan catatan bahwa sistem akan berawal di *state* s pada *stage* n dan xn telah terpilih sehingga f (s, xn) = csxn + fn + 1 \* (xn).
      3. fn\*(s) sebagai nilai maksimal/minimal dari fn(s, xn) untuk seluruh nilai xn yang mungkin.

Maka bentuk hubungan timbal-baliknya adalah:

fn\*(s) = maks/min { fn (s, xn) }

xn xn

* + 1. 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.
  1. **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* sn dapat direpresentasikan sebagai variabel state diskrit atau kontinu, atau mungkin diperlukan suatu vektor *state* (lebih dari satu variabel).

* 1. **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 fn (sn, xn) dengan fn + 1 (sn + 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 fn (sn, xn) menyatakan ekspektasi jumlah minimal dari *stage* n ke muka, berdasarkan state sn dan keputusan xn pada *stage* n maka:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| fn (sn, xn) = | N |  |  |  |  |
| Σ pi [ci + fn\*+1 (i)] | |  |  |  |
| i=1 |  |  |  |  |

Dengan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| f\*n+1 (sn+1) = | min fn+i (sn+1, xn+1) | | |  |  |
|  | xn+1 |  |  |  |  |

di mana minimasi ini diambil dari nilai-nilai xn+1 yang fisibel.

1. **Teori Probabilitas**
   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.

* 1. **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 ≤ P(A) ≤ 1

P(ɸ) = 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 ᴗ B) = P(A) + P(B) – P(A ᴖ B)

Hal ini dikenal sebagai hukum penjumlahan

* 1. **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 | A) = | P(A ᴖ B) | Dengan P(A) ≥ 0 |
|  | P(A) |

Sehingga P (A ᴖ B) = P (A) P (B | A)

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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | P (B | A) = | P(A ᴖ B) | = | P(A) . P(B) | = | P(B) |
|  | P(A) | P(A) |

|  |  |  |  |
| --- | --- | --- | --- |
| Sebagai analogi diperoleh: | P (B | A) = | P(A ᴖ B) |  |
| P(A) |

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

* 1. **Variabel Random dan Beberapa Macam Distribusinya**
  2. **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* si teetentu, ekspresi x(t, si) menyatakan suatu fungsi waktu tunggal, sedangkan untuk suatu harga ti tertentu x(ti, s) adalah suatu besaran yang bergantung pada s, dan disebut sebagai variabel random. Akhirnya, x(ti, si) 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:

1. Suatu keluarga dari fungsi waktu (t dan s berupa variabel).
2. Suatu fungsi waktu tunggal (t variabel, s tetap).
3. Suatu variabel random (t tetap, s variabel).
4. Suatu bilangan tunggal (t tetap, s tetap).
5. **Proses Keputusan Markov**
   1. **Ilustrasi Persoalan Keputusan Markov**
   2. **Membangun Matriks Probabilitas Transisi**
   3. **Model Programa**
   4. **Model 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.

1. **Teori Antrian**
   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

1. 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.

1. 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*.

1. 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)

1. 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)

* 1. **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.

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

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

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

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

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

Jika λn adalah konstan untuk semua n, maka dapat ditulis sebagai λ. Jika µn konstan untuk semua n ≥ 1, maka dapat ditulis sebagai µ. Disini µn = Sµ jika n ≥ S sehingga seluruh pelayan (sejumlah S) sibuk. Dalam hal ini 1/λ menyatakan ekspektasi waktu di antara kedatangan, sedangkan 1/µ menyatakan ekspektasi waktu pelayanan.

ρ = λ/ Sµ 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.

Lq : ekspektasi panjang antrian.

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

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

*Hubungan antara L dan W*

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

L = λ W

Lq = λ Wq

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

W = Wq + 1/µ

kalikan dengan λ, didapat:

L = Lq + P

* 1. **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.

* 1. **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.)

* 1. **Solusi Steady State**
  2. **Model Single Server**
  3. **Model Multiple Server**
  4. **Model Disiplin Prioritas**
  5. **Model Swalayan**