**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 variabel 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. **Kasus Khusus**
  2. **Bentuk Standar Model Programa Linier**
  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 = 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)
  1. **Kasus Khusus dalam Penggunaan Algoritma Simpleks**
     1. Degenerasi
     2. Solusi Optimum 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**
   2. **Model Transshipment**
   3. **Model Penugasan**
2. **Analisis Jaringan**
   1. **Terminologi Jaringan**
   2. **Persoalan Rute Terpendek**
   3. **Persoalan Rentang Pohon Minimum**
   4. **Persoalan Aliran Maksimal**
3. **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**
  2. **Perhitungan Kelonggaran Waktu**
  3. **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**
2. **Teori Permainan**
   1. **Elemen-elemen Dasar Teori Permainan**
   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*).

* 1. **Pure Strategy Game**
  2. **Mixed Strategy Game**
  3. **Solusi Grafis dari Permainan (2 x n) dan (m x 2)**
  4. **Solusi Permainan (m x n) dengan Programa Linier**

1. **Programa Dinamis**
   1. **Ilustrasi Programa Dinamis**
   2. **Karakteristik Persoalan Programa Dinamis**
   3. **Programa Dinamis Deterministik**
   4. **Programa Dinamis Probabilistik**
2. **Teori Probabilitas**
   1. **Ruang Sampel dan Peristiwa**
   2. **Probabilitas Suatu Event dan Hukum-hukumnya**
   3. **Probabilitas Terkondisi**
   4. **Variabel Random dan Beberapa Macam Distribusinya**
   5. **Proses Stochastic**
3. **Proses Keputusan Markov**
   1. **Ilustrasi Persoalan Keputusan Markov**
   2. **Membangun Matriks Probabilitas Transisi**
   3. **Model Programa**
   4. **Model dengan Stage Tidak Terbatas**
4. **Teori Antrian**
   1. Struktur dasar model-model antrian
   2. Terminologi dan notasi
   3. Contoh sistem antrian
   4. Proses birth and death
   5. Solusi steady state
   6. Model single server
   7. Model multiple server
   8. Model disiplin prioritas
   9. Model swalayan