

Modul/Penuntun/Panduan

Nama Mhs :

NIM :

Hari/Jam :

No. Komp :

PRAKTIKUM RISET OPERASI AGRIBISNIS

(SAGB3053)

Software Aplikasi:

POM-QM

(Production and Operation Quantity Methode)

TIM PENYUSUN

Oleh:

TIM Pengasuh Mata Kuliah Praktikum Riset Operasi Agribisnis (SAGB3053)
Departemen Agribisnis, Fakultas Pertanian, Universitas Syiah Kuala

&

Laboratorium Statistik Dan Sosial, Fakultas Pertanian, Universitas Syiah Kuala



Fakultas Pertanian

Universitas Syiah Kuala

2024



KATA PENGANTAR

Panduan praktikum ini ditujukan sebagai pelengkap kuliah Riset Operasi pada Fakultas Pertanian Universitas Syiah Kuala dan untuk menambah pemahaman mahasiswa terhadap materi perkuliahan. Buku panduan ini terdiri dari dua bagian. Bagian pertama berisi pengenalan software POM-QM for Windows sebagai sarana untuk mempermudah perhitungan dan bagian dua berisi materi-materi perkuliahan riset operasi yang telah disusun berdasarkan keeratan hubungan antar materinya.

Pemilihan software POM-QM for Windows didasarkan pada kemudahan dalam pengoperasiannya, tampilan grafis yang menarik, kemudahan dalam memahami output hasil perhitungan dan membuat kesimpulan serta tidak memerlukan spesifikasi komputer yang tinggi.

Tim penyusun mengucapkan terima kasih kepada para dosen yang telah memberikan dukungan untuk menyelesaikan buku panduan ini. Tim juga dengan senang hati menerima segala bentuk saran dan kritik demi perbaikan panduan praktikum ini di masa yang akan datang.

Semoga bermanfaat.

Tim Penyusun

DAFTAR ISI

| | |
|---|----|
| KATA PENGANTAR | i |
| DAFTAR ISI | ii |
| | |
| BAGIAN PERTAMA: PENGENALAN PROGRAM POM-QM for Windows.. | 1 |
| BAB I MEMULAI PROGRAM | 2 |
| BAB II MENU UTAMA | 6 |
| BAB III MEMULAI MODUL | 11 |
| | |
| BAGIAN KEDUA: MODUL PRAKTIKUM | 13 |
| BAB IV LINEAR PROGRAMMING | 14 |
| BAB V INTEGER PROGRAMMING | 20 |
| BAB VI MODEL TRANSPORTASI | 24 |
| BAB VII MODEL PENUGASAN | 28 |
| BAB VIII PEMOGRAMAN DINAMIS | |
| BAB IX INVENTARIS | |
| BAB VIII GOAL PROGRAMMING | 31 |
| BAB IX NETWORKS | 35 |
| BAB X GAME THEORY | 40 |
| BAB XI WAITING LINE | 44 |
| DAFTAR PUSTAKA | 55 |

BAGIAN PERTAMA

Pengenalan Program POM-QM for Windows

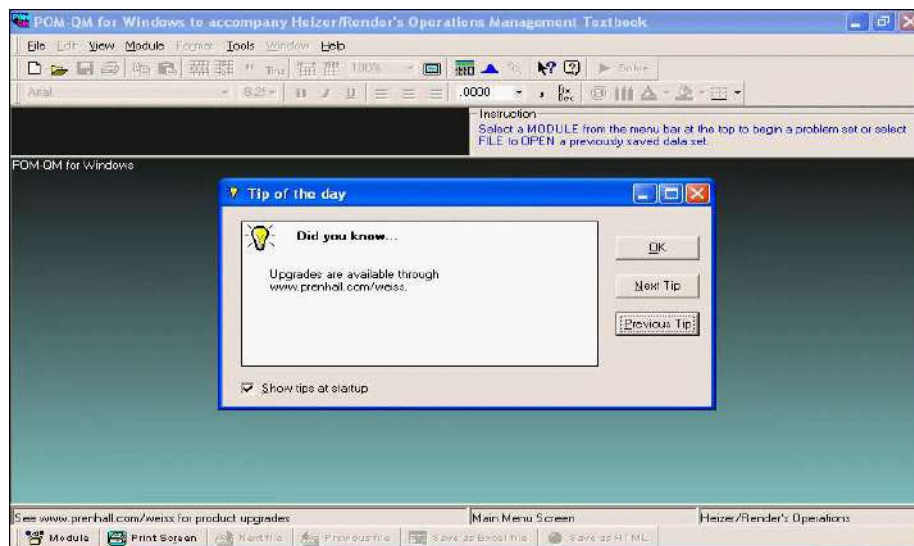
BAB I

MEMULAI PROGRAM

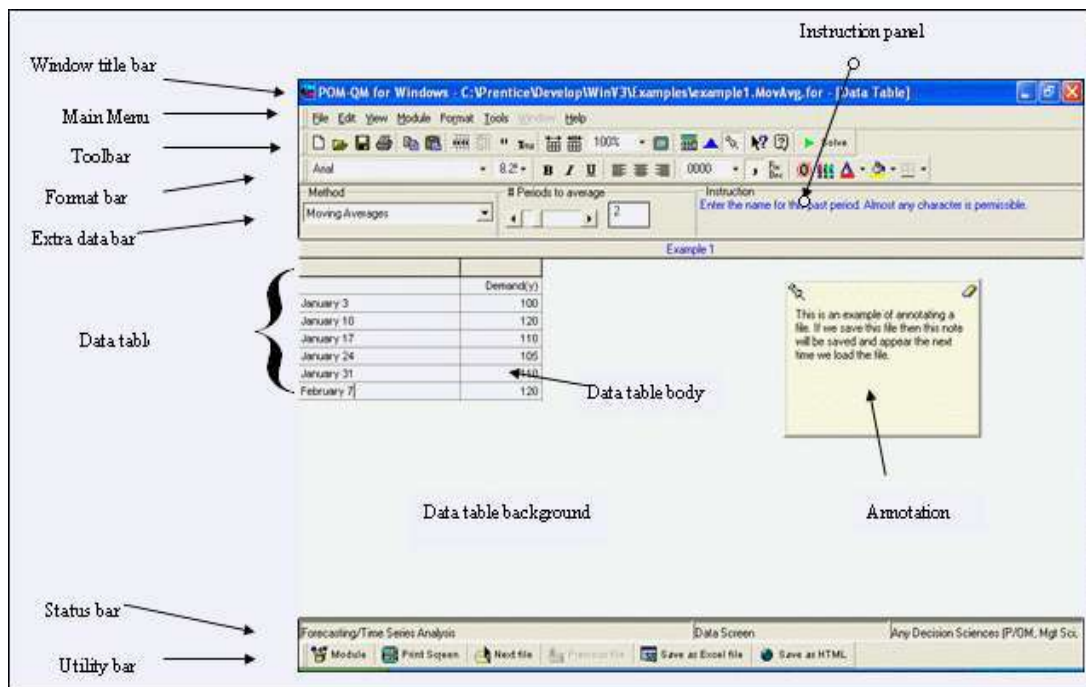
Program POM-QM for Windows umumnya memiliki kesamaan dengan program-program lain yang bisa dijalankan di bawah system operasi Windows. Untuk memulai program ini (program harus sudah diinstall pada perangkat computer) adalah dengan cara double klik pada ikon di layar desktop computer. Cara lain adalah klik tombol **Start** → **Programs** → **POM-QM for Windows 4** → **POM-QM for Windows 4**, kemudian muncul layar pembuka seperti di bawah ini.



Layar utama dari program akan muncul dalam beberapa detik setelah layar pembuka muncul. Pertama kali layar ini muncul, layar Tip of the Day akan muncul seperti di bawah ini.

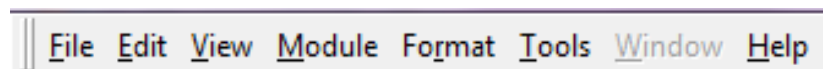


Untuk menunjukkan semua komponen dari layar kerja POM-QM, berikut telah diaktifkan contoh suatu modul dengan datanya.





Bagian atas layar menunjukkan bar nama atau title standar dari tampilan Windows. Pada awalnya program memberi nama *POM-QM for Windows* (atau *POM for Windows* atau *QM for Windows*) lengkap dengan tombol minimize, maximize, dan close.

Di bawah bar nama terdapat bar menu utama yang terdiri dari pilihan **File**, **Edit**, **View**, **Module**, **Format**, **Tools**, **Window**, dan **Help** yang akan dijelaskan pada bagian Menu Utama. Pada awal program diaktifkan, menu Edit dan Window tidak aktif. Kedua menu ini akan aktif ketika data telah dimasukkan dan ada hasil output dari olahan data.



Kemudian di bawah menu utama ada toolbar standar (disebut juga ribbon). Toolbar ini terdiri dari shortcut untuk beberapa perintah yang biasa digunakan. Salah satu tombol yang penting pada toolbar ini adalah tombol SOLVE ► pada ujung kanan toolbar. Tombol ini yang harus ditekan atau diklik untuk mendapatkan output setelah data dimasukkan. Atau bisa juga digunakan **File**, **Solve** atau tekan tombol

[F9] pada keyboard. Setelah menekan tombol SOLVE, tombol ini akan berubah menjadi tombol EDIT . Ini digunakan apabila table data akan diubah atau diedit. Untuk modul linear programming dan transportation, ada satu lagi tombol perintah yang akan muncul, yaitu tombol STEP  (tidak ditunjukkan pada gambar di bawah ini). Tombol ini digunakan untuk melihat setiap iterasi dari perhitungan.



Kemudian ada format toolbar yang sama dengan toolbar yang ada pada Ms. Excel, Word, dan program Windows lainnya.



Pada bagian bawah layar terdapat bar utility yang terdiri dari enam panel yang menginformasikan module yang sedang aktif, print screen, data yang terbuka, menyimpan data dengan format xls dan html.



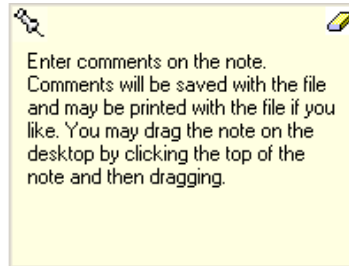
Pada bagian tengah terdapat table data utama yang terdiri dari kepala table, baris dan kolom. Jumlah baris dan kolom tergantung dari modul dan jenis permasalahan.

| Example | | | | | | |
|--------------|----|----|----|----|-----|---------------|
| | X1 | X2 | X3 | | RHS | Equation form |
| Maximize | 0 | 0 | 0 | | | Max |
| Constraint 1 | 0 | 0 | 0 | <= | 0 | <= 0 |
| Constraint 2 | 0 | 0 | 0 | <= | 0 | <= 0 |
| Constraint 3 | 0 | 0 | 0 | <= | 0 | <= 0 |
| Constraint 4 | 0 | 0 | 0 | <= | 0 | <= 0 |

Di atas tabel data terdapat bar data tambahan untuk menempatkan informasi masalah tambahan untuk menunjukkan fungsi tujuan (meminimalkan atau memaksimalkan), memilih metode, atau memberikan beberapa nilai. Ini umumnya muncul di atas data. Di sebelah kanan panel data tambahan adalah panel instruksi. Di sini selalu ada instruksi untuk membantu Anda untuk menentukan apa yang harus dilakukan. Ketika data yang akan dimasukkan ke dalam tabel data, instruksi ini akan menjelaskan jenis data (integer, real, positif, dan lain-lain) yang akan dimasukkan. Lokasi instruksi dapat diubah dengan menggunakan menu View.



Terdapat juga form untuk memberi keterangan atau komentar. Ketika file tersebut disimpan, komentar akan disimpan; ketika file tersebut ditampilkan, komentar akan muncul dan komentar dapat dicetak jika diinginkan.



Pada bawah layar adalah status bar. Panel paling kiri menampilkan nama modul dan submodel yang dipilih, sebagaimana dicontohkan dalam ilustrasi ini di mana modul Peramalan dan submodel adalah Analisis Time Series. Panel tengah berisi jenis layar (data, hasil, menu, grafik, dll) dan panel paling kanan memiliki nama buku (jika buku teks telah dipilih). Status bar dapat disembunyikan dengan menggunakan menu View opsi. Panel ini tidak dapat dipindahkan.

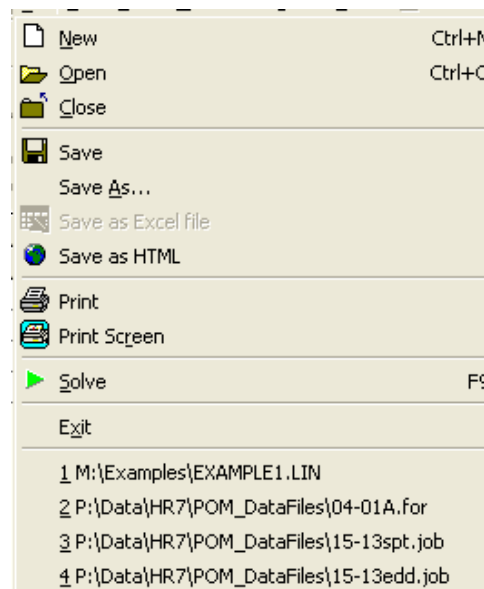


BAB II MENU UTAMA






Pada umumnya program-program yang dijalankan pada system operasi Windows, selalu terdapat menu utama yang mempermudah pengguna dalam menggunakan program tersebut. Dalam program POM-QM terdapat menu atau pilihan File, Edit, View, Module, Format, Tools, Window, dan Help



A. File

Menu File berisi pilihan yang biasa dijumpai pada program-program windows seperti di bawah ini.



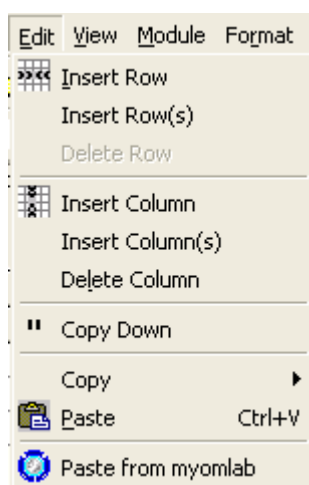
Keterangan menu File

| | |
|--|---|
| New  | memulai file baru |
| Open  | membuka data yang sudah tersimpan dalam harddisk/media penyimpanan data |
| Save  | menyimpan data atau untuk menggantikan data yang telah diperbaharui |
| Save as | menyimpan data yang baru |
| Save as Excel File  | menyimpan data dengan extensi file excel |
| Save as HTML  | Menyimpan data dalam format HTML |



| | |
|--|---|
| Print  | Menampilkan layar cetak kertas dan mencetak di kertas |
| Print Screen  | Menampilkan layar cetakan pada kertas |
| Solve ▶ | Melihat hasil perhitungan. Selain menggunakan sub menu Solve, dapat juga menekan F9 |
| Exit | Mengakhiri atau keluar dari program |

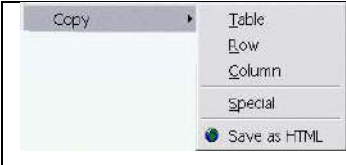

B. Edit

Perintah-perintah dalam menu Edit terlihat seperti di bawah ini:



Keterangan menu edit.

| | |
|---|--|
| Insert Row  | Menyisipkan baris pada table data dimana kursor diletakkan |
| Insert Rows(s) | Menyisipkan baris lebih dari satu baris |
| Delete Row | Menghapus baris pada table dimana kursor diletakkan |
| Insert Column  | Menyisipkan kolom pada table data dimana kursor diletakkan |
| Insert Columns(s) | Menyisipkan kolom lebih dari satu kolom |
| Delete Column | Menghapus kolom pada table dimana kursor diletakkan |
| Copy Down | Menyalin data dari satu sel ke semua sel di bawahnya |
| Copy | Menyalin table, baris, kolo, special dan sebagai HTML |


| | |
|---|-----------------------------|
|  | |
| Paste  | Menempel objek yang disalin |

C. View

Menu View terdiri dari beberapa pilihan untuk menyesuaikan tampilan layar.

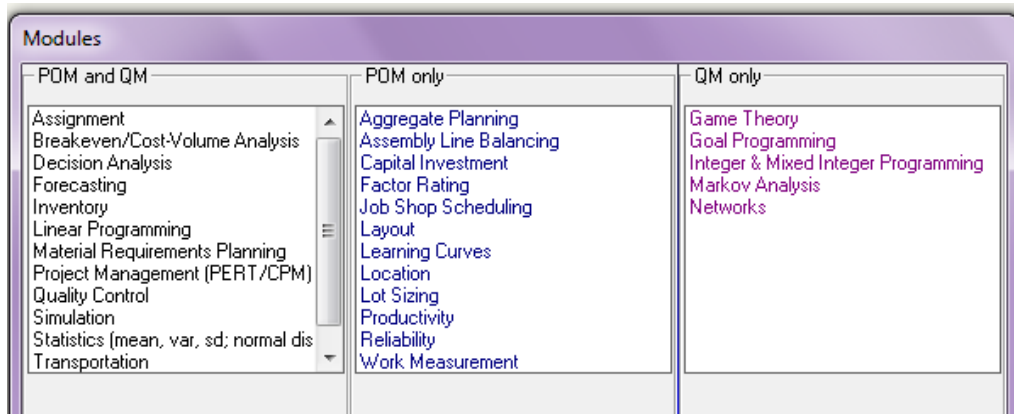


Keterangan menu view

| | |
|---|--|
| Toolbars | Toolbar dapat disesuaikan seperti tampilan Windows atau dapat di set kembali seperti bentuk asli |
| Instruction | Perintah untuk memindahkan letak kotak instruction |
| Status Bar | Mengaktifkan atau me-nonaktifkan status bar |
| Full Screen  | Menampilkan keseluruhan layar dengan menghilangkan semua bar |
| Zoom | Mengurangi atau menambah ukuran kolom data |
| Original Colors dan Monochrome | Merubah tampilan ke warna asli atau hitam putih |

D. Module

Menu module terdiri dari semua modul yang ada pada ulasan Production and Operations Management dan Quantitative Methods.



Beberapa modul di atas akan dibahas pada Bab Modul Praktikum.

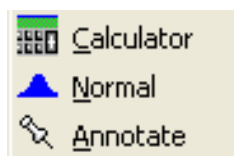
E. Format

Menu format terdiri dari beberapa pilihan untuk tampilan data dan table solusi, seperti pilihan warna, huruf, merubah tampilan garis table solusi, merubah ukuran kolom table, merubah title, membuat angka decimal pada angka, dan lain-lain.






F. Tools

Menu Tool digunakan untuk membantu dalam melakukan perhitungan dan menyisipkan catatan untuk pengingat.

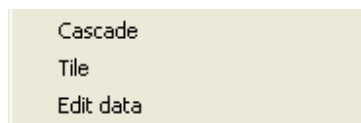


Keterangan menu tool

| | |
|---|--|
| Calculator  | Menampilkan kalkulator |
| Normal  | Menampilkan kalkulator distribusi normal |
| Annotate  | Menampilkan catatan |

G. Window

Menu window dapat menampilkan layar solusi (solution screen). Menu ini memiliki pilihan untuk menampilkan semua layar solusi (cascade dan tile) dan untuk merubah data (edit data)



H. Help

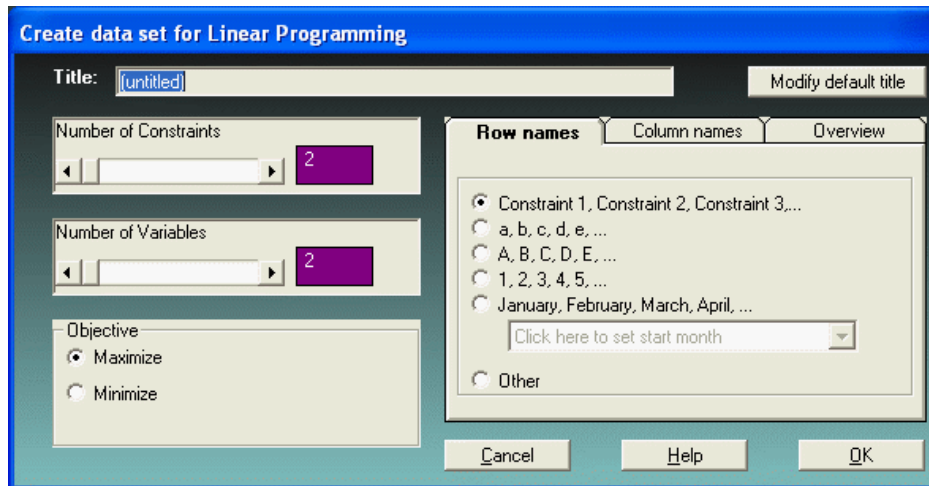
Menu help berisi semua informasi mengenai software QM for Windows.



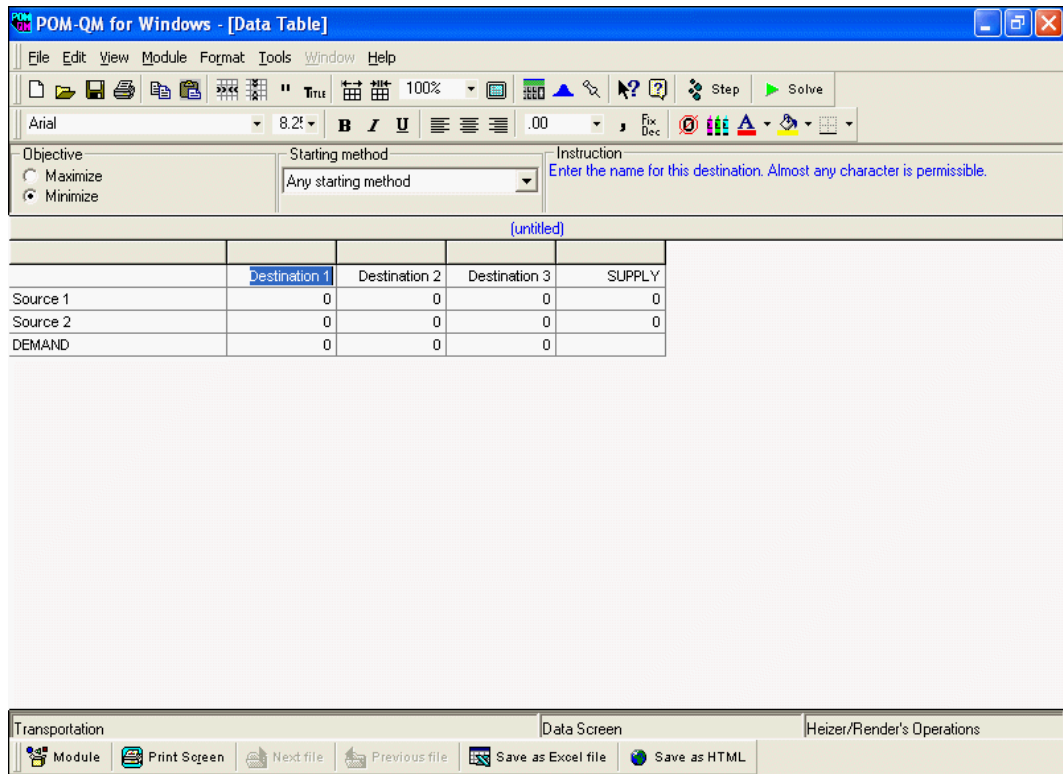
BAB III

MEMULAI MODUL

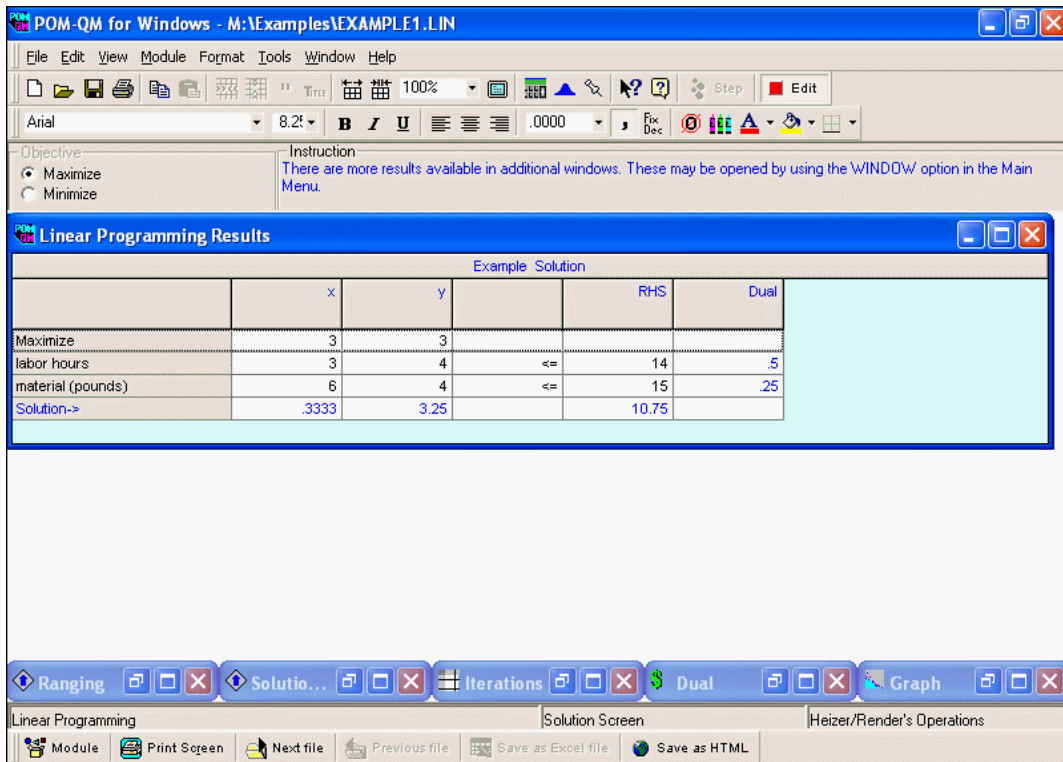
Secara umum untuk memulai suatu modul, menu yang dipilih adalah Module dan memilih modul yang diinginkan. Kemudian pilih menu File dan diikuti New untuk memulai data set, atau pilih Open untuk menampilkan data yang telah disimpan. Gambar di bawah ini adalah contoh tampilan yang muncul ketika akan membuat suatu set data. Tampilan untuk membuat set data sangat mirip untuk setiap modul, hanya sedikit perbedaan tergantung dari tipe permasalahan.



Bagian atas adalah kotak isian title atau nama permasalahan. Kotak ini dapat diisi atau diabaikan. Secara default akan tertulis [untitled]. Kemudian pada bagian bawah adalah menentukan banyaknya baris dan kolom. Baris dan kolom akan mempunyai nama yang berbeda tergantung dari modul. Untuk contoh, dalam pemograman linear, baris adalah “constraint” dan kolom adalah “variable”. Setelah itu menentukan fungsi tujuan (objective) atau tipe data dan nama baris dan kolom. Tab overview memberikan penjelasan singkat tentang model dan data yang dimasukkan. Setelah semua isian lengkap, klik tombol [OK] dan akan muncul layar (data screen) berisi table dimana data akan diisi.



Setelah data diisi, klik tombol Solve untuk mendapatkan hasil perhitungan yang ditampilkan di layar output.



BAGIAN KEDUA

MODUL PRAKTIKUM

BAB III

LINEAR PROGRAMMING

Linear programming merupakan suatu model matematis untuk menggambarkan masalah yang dihadapi yang terdiri dari fungsi tujuan (berupa suatu persamaan linear) dan fungsi kendala (berupa beberapa persamaan atau pertidaksamaan linear).

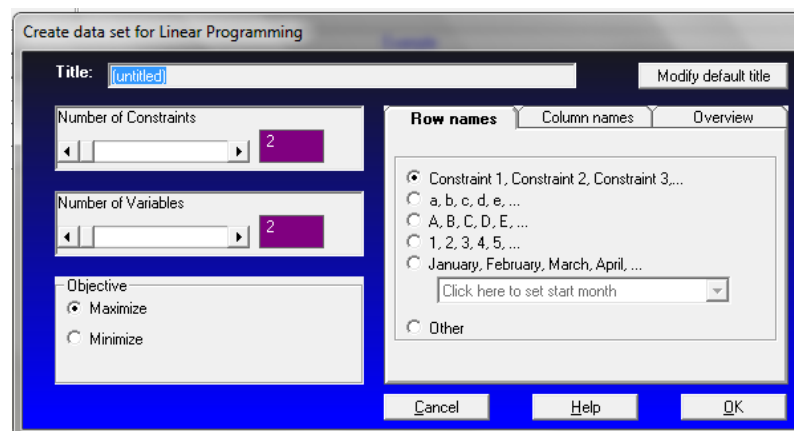
Fungsi tujuan menggambarkan apa yang ingin dicapai perusahaan dengan menggunakan sumber daya yang ada dalam bentuk maksimasi (misalnya untuk laba, penerimaan, produksi, dan lain-lain) atau minimasi (misalnya untuk biaya, waktu dan lain-lain). Fungsi kendala menggambarkan kendala-kendala yang dihadapi perusahaan dalam kaitannya dengan pencapaian tujuan tersebut, misalnya mesin, tenaga kerja, dan lain-lain.

Contoh 1 (kasus maksimum).

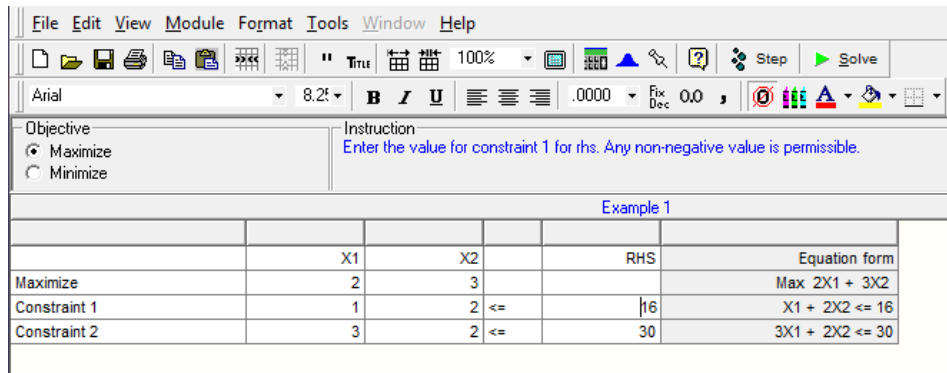
$$\begin{aligned} \text{Max} \quad & 2x_1 + 3x_2 \\ \text{Kendala} \quad & x_1 + 2x_2 \leq 16 \\ & 3x_1 + 2x_2 \leq 30 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Cara memasukkan data:

1. Klik Module – Linear Programming
2. Klik File – New, sehingga muncul tampilan *Create data set for Linear Programming*:



3. Tentukan title dari kasus, jumlah batasan (*constraint*), jumlah variable, nama baris dan kolom data set yang akan diisi. Untuk contoh, title: Example, jumlah kendala: 2, jumlah variable: 2. Nama baris dan kolom bisa dipilih dari beberapa pilihan atau diisi pada tampilan data set.
4. Klik Ok, sehingga muncul tampilan data set.
5. Isi data set dan klik Solve untuk melihat hasil output.



| | X1 | X2 | | RHS | Equation form |
|--------------|----|----|----|-----|-----------------|
| Maximize | 2 | 3 | | | Max 2X1 + 3X2 |
| Constraint 1 | 1 | 2 | <= | 16 | X1 + 2X2 <= 16 |
| Constraint 2 | 3 | 2 | <= | 30 | 3X1 + 2X2 <= 30 |

Keterangan tampilan data set:

- Fungsi objektif (*objective function*), untuk menentukan pilihan fungsi tujuan.
- Koefisien fungsi objektif (*objective function coefficients*), dapat berupa nilai positif atau negatif.
- Koefisien fungsi kendala (*constraint coefficients*), dapat berupa nilai positif atau negatif.
- Koefisien *Right-hand side (RHS)*. Nilai batasan optimal pada fungsi kendala. Nilai ini harus non-negatif.
- *Equation form*. Kolom yang berada pada ujung kanan data set, berisi bentuk persamaan fungsi tujuan dan fungsi kendala.

Output dan Analisis

Modul linear programming akan menampilkan layar output, yaitu: *Linear Programming Result, Ranging, Original Problem w/answers, Iterations, Dual dan Graph* (jika jumlah variable lebih dari dua, grafik tidak dapat ditampilkan).

Linear Programming Result

| Example 1 Solution | | |
|--------------------|----------|-------|
| Variable | Status | Value |
| X1 | Basic | 7 |
| X2 | Basic | 4.5 |
| slack 1 | NONBasic | 0 |
| slack 2 | NONBasic | 0 |
| Optimal Value (Z) | | 27.5 |

Ranging

| Example 1 Solution | | | | | |
|--------------------|------------|---------------|--------------|-------------|-------------|
| Variable | Value | Reduced Cost | Original Val | Lower Bound | Upper Bound |
| X1 | 7 | 0 | 2 | 1.5 | 4.5 |
| X2 | 4.5 | 0 | 3 | 1.3333 | 4 |
| Constraint | Dual Value | Slack/Surplus | Original Val | Lower Bound | Upper Bound |
| Constraint 1 | 1.25 | 0 | 16 | 10 | 30 |
| Constraint 2 | .25 | 0 | 30 | 16 | 48 |

Dual

| Example 1 Solution | | | | | |
|-------------------------|--------------|--------------|----|--|----|
| Original Problem | | | | | |
| Maximize | X1 | X2 | | | |
| Constraint 1 | 1 | 2 | <= | | 16 |
| Constraint 2 | 3 | 2 | <= | | 30 |
| Dual Problem | | | | | |
| Minimize | Constraint 1 | Constraint 2 | | | |
| X1 | 16 | 30 | | | |
| X2 | 1 | 3 | >= | | 2 |
| X2 | 2 | 2 | >= | | 3 |

Original Problem w/answers

| Example 1 Solution | | | | | | |
|--------------------|----|-----|-------------|------|------|--|
| | X1 | X2 | | RHS | Dual | |
| Maximize | 2 | 3 | | | | |
| Constraint 1 | 1 | 2 | <= | 16 | 1.25 | |
| Constraint 2 | 3 | 2 | <= | 30 | .25 | |
| Solution-> | 7 | 4.5 | Optimal Z-> | 27.5 | | |

Dari semua tampilan output di atas, dapat disimpulkan bahwa nilai optimum (maksimum) yang dapat dicapai adalah 27,5 dengan menghasilkan x_1 sebanyak 7 unit dan x_2 sebanyak 4,5 unit tanpa ada sumber daya yang tersisa (jumlah x_1 dan x_2 habis

digunakan). Pada tabel ranging terlihat bahwa Reduced cost untuk produk x_1 dan x_2 adalah 0. Ini menunjukkan bahwa jumlah x_1 dan x_2 yang dihasilkan sudah optimal. Tabel juga menampilkan original value untuk x_1 dan x_2 yang dipakai yaitu 16 unit dan 30 unit, habis terpakai untuk menghasilkan x_1 dan x_2 . Ini dapat dilihat dari nilai 0 pada kolom Slack/Surplus. Kolom Dual Value (the shadow price) menunjukkan tambahan nilai fungsi tujuan jika ada tambahan input (constraint 1 dan constraint 2) sebesar 1 unit. Pada contoh di atas, fungsi tujuan akan naik sebesar 1,25 unit jika ada kenaikan nilai kendala 1 sebesar 1 unit; dan fungsi tujuan akan naik sebesar 0,25 unit jika ada kenaikan nilai kendala 2 sebesar 1 unit.

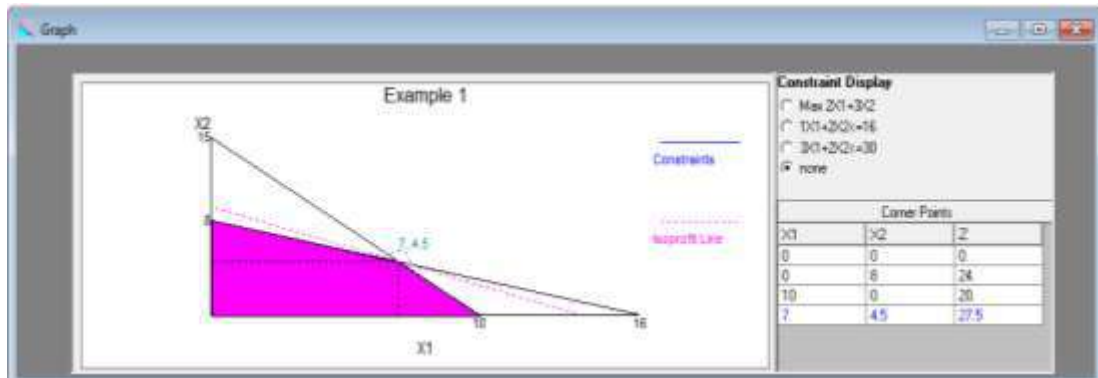
Nilai Lower Bound dan Upper Bound digunakan untuk melakukan analisis sensitivitas. Analisis sensitivitas merupakan analisis yang bertujuan untuk memberikan jawaban atas seberapa jauh perubahan dibenarkan tanpa merubah solusi optimum atau tanpa menghitung solusi optimum baru dari awal yang dinyatakan dengan nilai batas atas dan batas bawah (Lower Bound dan Upper Bound). Batas atas dan batas bawah koefisien fungsi tujuan untuk variabel x_1 adalah 1,5 sampai 4,5, sedangkan untuk variabel x_2 yaitu 1,33 sampai 4. Ini berarti nilai koefisien bisa diubah sesuai dengan batas atas dan batas bawah yang dianjurkan karena pada rentang nilai koefisien, fungsi tujuan ini tidak akan merubah nilai optimalnya. Batas atas dan batas bawah koefisien fungsi tujuan untuk batasan kendala 1 adalah 10 sampai 30 dan untuk kendala 2 adalah 16 sampai 48. Ini berarti nilai koefisien bisa diubah sesuai dengan batas atas dan batas bawah yang dianjurkan karena pada rentang nilai koefisien, fungsi tujuan ini tidak akan merubah nilai keuntungan yang diperoleh.

Iterasi

| Example 1 Solution | | | | | | |
|--------------------|-----------------|----------|---------|---------|--------------|--------------|
| Cj | Basic Variables | Quantity | 2 X1 | 3 X2 | 0 slack 1 | 0 slack 2 |
| Iteration 1 | | | | | | |
| 0 | slack 1 | 16 | 1 | 2 | 1 | 0 |
| 0 | slack 2 | 30 | 3 | 2 | 0 | 1 |
| | zj | 0 | 0 | 0 | 0 | 0 |
| | cj-zj | | 2 | 3 | 0 | 0 |
| Iteration 2 | | | | | | |
| 3 | X2 | 8 | 0.5 | 1 | 0.5 | 0 |
| 0 | slack 2 | 14 | 2 | 0 | -1 | 1 |
| | zj | 24 | 1.5 | 3 | 1.5 | 0 |
| | cj-zj | | 0.5 | 0 | -1.5 | 0 |
| Iteration 3 | | | | | | |
| 3 | X2 | 4.5 | 0 | 1 | 0.75 | -0.25 |
| 2 | X1 | 7 | 1 | 0 | -0.5 | 0.5 |
| | zj | 27.5 | 2 | 3 | 1.25 | .25 |
| | cj-zj | | 0 | 0 | -1.25 | -0.25 |

Grafik

Pada tampilan grafik, feasible area selalu diwarnai. Pada bagian kanan tampilan adalah table dari titik sudut dari *feasible area* dan nilai fungsi tujuan dari masing-masing titik sudut



Soal latihan

1. Memaksimumkan suatu fungsi tujuan (profit) dalam menghasilkan dua produk x dan y dengan dua kendala yaitu jam kerja dan bahan untuk memproduksi produk tersebut:

$$\text{Maximize } 3x + 3y$$

$$\text{Kendala } 3x + 4y \leq 14 \text{ (labor hours)}$$

$$6x + 4y \leq 15 \text{ (pounds of material)}$$

$$x, y \geq 0$$

2. PT Sirup Segar Benar memproduksi dua jenis sirup dengan rasa jeruk dan anggur. Untuk memproduksi sirup rasa jeruk dibutuhkan biaya produksi sebesar Rp. 7.000,- dan sirup rasa anggur sebesar Rp. 8.000,-. Produk sirup tersebut melewati dua mesin dalam produksinya, yaitu mesin pencampuran dan pengolahan. Kapasitas mesin dapat digunakan minimum/lebih dari 15.000 jam untuk mesin pencampuran dan 9.750 jam untuk mesin pengolahan. Untuk memproduksi sirup jeruk diperlukan 30 jam di mesin pencampuran dan 15 jam di mesin pengolahan. Sementara itu untuk memproduksi sirup anggur diperlukan 15 jam di mesin pencampuran dan 12 jam di mesin pengolahan. Berapa unit sirup jeruk dan anggur yang diproduksi agar biaya produksi kedua sirup tersebut minimum.

$$\begin{aligned}
 3. \text{ Min } Z & \quad 8X_1 + 2X_2 \\
 \text{Kendala} & \quad 2X_1 - 6X_2 \leq 12 \\
 & \quad 5X_1 + 4X_2 \geq 40 \\
 & \quad X_1 + 2X_2 \geq 12 \\
 & \quad X_2 \leq 6 \\
 & \quad X_1, X_2 \geq 0
 \end{aligned}$$

4. Valley Wine Company memproduksi dua jenis anggur – Valley Nectar dan Valley Red. Anggur-anggur ini diproduksi dari 64 ton buah anggur yang didapatkan perusahaan pada musim ini. Satu kelompok produksi 1.000 galon Nectar membutuhkan 4 ton buah anggur, dan satu kelompok produksi Red membutuhkan 8 ton. Tetapi produksi terbatas oleh ketersediaan hanya 50 yard kubik tempat penyimpanan dan 120 jam waktu pemrosesan. Sekelompok produksi dari masing-masing anggur membutuhkan 5 yard kubik tempat penyimpanan. Waktu pemrosesan untuk satu kelompok produksi Nectar adalah 15 jam dan waktu pemrosesan untuk satu kelompok produksi Red adalah 8 jam. Permintaan untuk masing-masing jenis anggur terbatas kurang lebih sebanyak satu kelompok produksi. Laba untuk satu kelompok produksi Nectar adalah sebesar \$9.000, dan untuk kelompok produksi Red sebesar \$12.000. Perusahaan tersebut ingin menentukan jumlah kelompok produksi 1.000 galon Nectar dan Red yang harus diproduksi untuk memaksimalkan laba.

- a. Rumuskan model.
 - b. Berapa banyak waktu pemrosesan yang tersisa dan tidak terpakai pada solusi optimal.
 - c. Apa pengaruhnya terhadap solusi optimal jika kita meningkatkan ruang penyimpanan yang tersedia dari 50 menjadi 60 yard kubik.
5. Kalo Fertilizer Company membuat pupuk dengan menggunakan dua bahan kimia yang memberikan nitrogen, fosfat dan kalium. Satu pon bahan 1 menyumbang 10 ons nitrogen dan 6 ons fosfat., sementara satu pon bahan 2 menyumbang 2 ons nitrogen, 6 ons fosfat dan 1 ons kalium. Bahan 1 harganya \$3 per pon, dan bahan 2 harganya \$5 perpon. Perusahaan tersebut ingin mengetahui berapa pon dari masing-masing bahan kimia yang harus dimasukkan ke dalam satu kantung pupuk untuk memenuhi persyaratan minimum 20 ons nitrogen, 36 ons fosfat, dan 2 ons kalium dengan meminimalkan biaya.
 6. Supermarket Kroeger menjual kacang polong merk sendiri dan beberapa merk nasional. Toko tersebut menghasilkan laba sebesar \$0,28 perkaleng kacang polongnya sendiri dan \$0,19 untuk sembarang merk nasional. Toko tersebut mempunyai 6 kaki persegi ruang pajang yang tersedia untuk kacang polong kalengan, dan setiap kaleng kacang polong menempati 9 inchi persegi dari ruangan itu. Catatan titik penjualan menunjukkan bahwa setiap minggu toko tersebut tidak pernah menjual mereknya sendiri lebih dari separuh jumlah mereka menjual merk nasional. Toko tersebut ingin mengetahui berapa banyak kaleng kacang polong mereknya sendiri dan berapa banyak kaleng merk nasional yang harus disimpan dalam persediaan setiap minggunya diruang pajang untuk memaksimalkan laba.
 7. Seorang mahasiswa agribisnis Unsyiah memelihara sebuah kebun besar di mana ditanami kubis, tomat dan bawang merah untuk membuat dua jenis asinan – *Asin-enak* dan *Asin-sedap*. *Asin-enak* dibuat terutama dari kubis, sedangkan *Asin-sedap* memakai lebih banyak tomat dibandingkan *Asin-enak*. Kedua asinan menggunakan bawang merah, dan sedikit paprika dan rempah-rempah yang jumlahnya dapat diabaikan. Sebotol *Asin-enak* mengandung 8 ons kubis, 3 ons tomat, dan 3 ons bawang merah, sedangkan sebotol *Asin-sedap* mengandung 6 ons tomat, 6 ons kubis, dan 2 ons bawang merah. Mahasiswa

tersebut menanam 120 pon kubis, 90 pon tomat, dan 45 pon bawang merah dan dapat memproduksi paling banyak 24 lusin botol asinan. Usaha tersebut menghasilkan laba sebesar \$2,25 dari sebotol *Asin-enak* dan \$1,95 dari sebotol *Asin-sedap*. Sang mahasiswa ingin mengetahui berapa banyak dari masing-masing jenis asinan yang harus diproduksi untuk menghasilkan laba paling besar.

8. Seorang petani mempunyai tanah pertanian seluas 50 *ekar* untuk menanam stroberi dan tomat. Petani tersebut mempunyai 300 jam kerja yang tersedia perminggu dan 800 ton pupuk, dan ia mempunyai kontrak untuk ruang pengiriman untuk maksimum dari hasil 26 *ekar* stroberi dan hasil dari 37 *ekar* tomat. Satu *ekar* stroberi membutuhkan 10 jam kerja dan 8 ton pupuk, sementara satu *ekar* tomat membutuhkan 3 jam kerja dan 20 ton pupuk. Laba satu *ekar* stroberi adalah \$400, dan dari satu *ekar* tomat adalah \$300. Petani tersebut ingin mengetahui berapa jumlah *ekar* stroberi dan tomat yang harus ditanam untuk memaksimalkan laba.
9. Seorang petani yang memiliki 7 ha tanah sedang memikirkan berapa ha tanah yang harus ditanami jagung dan berapa ha yang harus ditanami gandum. Dia mengetahui bahwa jika ditanami jagung, setiap ha tanah akan menghasilkan 10 ton jagung. Untuk ini diperlukan 4 jam-orang setiap minggunya. Jika ditanami gandum, hasilnya adalah 25 ton/ha dan di perlukan 10 jam-orang/minggu. Setiap kg jagung dapat dijual seharga Rp. 30, sedangkan harga jual gandum adalah Rp. 40/kg. saat ini petani tersebut hanya memiliki 40 jam-orang setiap minggunya karena ada peraturan pemerintah yang mengharuskan setiap petani untuk menghasilkan gandum paling sedikit 30 ton setiap kali panen, bagaimanakah formulasi dan hasil perhitungan persoalan ini agar petani tsb dapat menggarap tanahnya secara optimal.

BAB IV INTEGER PROGRAMMING

Penyelesaian sebuah kasus pemograman linear mungkin menghasilkan nilai optimal variable-variabel keputusan yang berupa bilangan pecahan. Jika variable-variabel keputusan tersebut mewakili item-item yang tidak bias dipecah, misal: manusia, saham, mesin, dan lain-lain; maka keputusan optimal tentunya tidak bias diimplementasikan. Pemograman Bilangan Bulat (*Integer Programming*) adalah sebuah model matematis yang memungkinkan hasil penyelesaian kasus pemograman linear yang berupa bilangan pecahan diubah menjadi bilangan bulat tanpa meninggalkan optimalitas penyelesaian.

Contoh 1.

$$\begin{aligned} \text{Max} \quad & 2x_1 + 3x_2 \\ \text{Kendala} \quad & x_1 + 2x_2 \leq 16 \\ & 3x_1 + 2x_2 \leq 30 \\ & x_1, x_2 \geq 0 \text{ dan integer} \end{aligned}$$

Cara memasukkan data:

1. Klik Module – Integer & Mixed Integer Programming
2. Klik File – New, sehingga muncul tampilan *Create data set for Integer & Mixed Integer Programming*:
3. Tentukan title dari kasus, jumlah batasan (*constraint*), jumlah variable, nama baris dan kolom data set yang akan diisi. Untuk contoh, title: Example 4, jumlah kendala: 2, jumlah variable: 2. Nama baris dan kolom bisa dipilih dari beberapa pilihan atau diisi pada tampilan data set.
4. Klik Ok, sehingga muncul tampilan data set.
5. Isi data set dan klik Solve untuk melihat output hasil.

| Objective | | Maximum number of iterations | | Maximum level (depth) in procedure | |
|---|---------|-----------------------------------|----|------------------------------------|-----------------|
| <input checked="" type="radio"/> Maximize | | <input type="text" value="1000"/> | | <input type="text" value="50"/> | |
| <input type="radio"/> Minimize | | | | | |
| Example 4 | | | | | |
| | X1 | X2 | | RHS | Equation form |
| Maximize | 2 | 3 | | | Max: 2X1 + 3X2 |
| Constraint 1 | 1 | 2 | <= | 16 | X1 + 2X2 <= 16 |
| Constraint 2 | 3 | 2 | <= | 30 | 3X1 + 2X2 <= 30 |
| Variable type | Integer | Integer | | | |

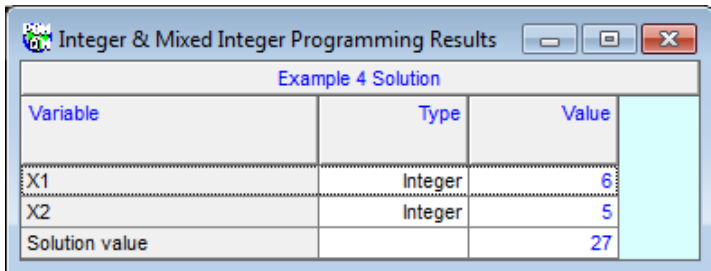
Keterangan tampilan data set

- Komponen dan entri data hampir sama dengan pemrograman linier. Perbedaannya adalah bahwa layar input memiliki satu baris tambahan untuk mengidentifikasi jenis variabel baik sebagai bilangan real, integer, atau 0/1.
- Fungsi objektif (*objective function*), untuk menentukan pilihan fungsi tujuan.
- Koefisien fungsi objektif (*objective function coefficients*), dapat dimasukkan sebagai nilai-nilai numerik.
- Koefisien fungsi kendala (*constraint coefficients*), dapat berupa nilai positif atau negatif.
- Koefisien sisi kanan atau *Right-hand side (RHS)*. Nilai batasan optimal pada fungsi kendala. Nilai ini harus non-negatif.
- *Equation form*. Kolom yang berada pada ujung kanan data set, berisi bentuk persamaan fungsi tujuan dan fungsi kendala.
- Jenis variabel. Kotak drop-down akan mengubah jenis variabel dari "integer" untuk "real" untuk "0/1." untuk mengubah semua variabel sekaligus dengan mengklik pada kolom paling kiri.
- Jumlah maksimum iterasi dan tingkat maksimum (kedalaman), untuk membuat perhitungan iterasi lebih rinci.

Output dan Analisis

Output ditunjukkan oleh tampilan sederhana dengan variabel dan nilai-nilainya.

Integer & Mixed Integer Programming Result



| Variable | Type | Value |
|----------------|---------|-------|
| X1 | Integer | 6 |
| X2 | Integer | 5 |
| Solution value | | 27 |

Dari tampilan output di atas, variable keputusan x1 dan x2 dikehendaki berupa bilangan bulat (integer), maka nilai optimum (maksimum) yang dapat dicapai adalah 27 dengan menghasilkan x1 sebanyak 6 unit dan x2 sebanyak 5. Berbeda dengan hasil yang didapat dari pemrograman linear, $x_1 = 7$ dan $x_2 = 4,5$ (bilangan pecahan) yang memberikan nilai maksimum 27,5.

Iterasi

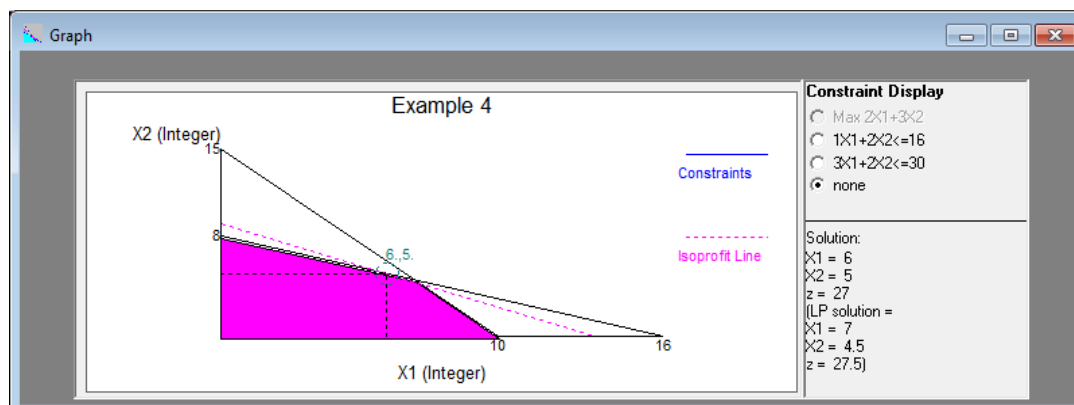
| Iteration Results | | | | | | |
|--------------------|-------|------------------|---------------|----------------|------|-----|
| Example 4 Solution | | | | | | |
| Iteration | Level | Added constraint | Solution type | Solution Value | X1 | X2 |
| | | | Optimal | 27 | 6 | 5 |
| 1 | 0 | | NONinteger | 27.5 | 7 | 4.5 |
| 2 | 1 | X2 <= 4 | NONinteger | 26.67 | 7.33 | 4 |
| 3 | 2 | X1 <= 7 | INTEGER | 26 | 7 | 4 |
| 4 | 2 | X1 >= 8 | Suboptimal | 25 | 8 | 3 |
| 5 | 1 | X2 >= 5 | INTEGER | 27 | 6 | 5 |

Dengan menggunakan metode **Cabang dan Batas (Branch and Bound)**, table iterasi menunjukkan perubahan nilai x1 dan x2 optimal yang berada disekitar nilai x1 dan x2 yang dihasilkan pada pemrograman linear.

Original Problem w/answer

| Original Problem w/answers | | | | |
|----------------------------|---------|---------|-------------|-----|
| Example 4 Solution | | | | |
| | X1 | X2 | | RHS |
| Maximize | 2 | 3 | | |
| Constraint 1 | 1 | 2 | <= | 16 |
| Constraint 2 | 3 | 2 | <= | 30 |
| Variable type | Integer | Integer | | |
| Solution-> | 6 | 5 | Optimal Z-> | 27 |

Grafik



Grafik menunjukkan perbandingan hasil yang didapat dengan hasil yang ada pada pemrograman linear.

Soal Latihan

1. Memaksimalkan suatu fungsi tujuan (profit) dalam menghasilkan dua produk x dan y dengan dua kendala yaitu jam kerja dan bahan untuk memproduksi produk tersebut:

$$\text{Maximize} \quad 3x + 3y$$

$$\text{Kendala} \quad 3x + 4y \leq 14 \text{ (labor hours)}$$

$$6x + 4y \leq 15 \text{ (pounds of material)}$$

$$x, y \text{ integer}$$

2. PT Sirup Segar Benar memproduksi dua jenis sirup dengan rasa jeruk dan anggur. Untuk memproduksi sirup rasa jeruk dibutuhkan biaya produksi sebesar Rp. 7.000,- dan sirup rasa anggur sebesar Rp. 8.000,-. Produk sirup tersebut melewati dua mesin dalam produksinya, yaitu mesin pencampuran dan pengolahan. Kapasitas mesin dapat digunakan minimum/lebih dari 15.000 jam untuk mesin pencampuran dan 9.750 jam untuk mesin pengolahan. Untuk memproduksi sirup jeruk diperlukan 30 jam di mesin pencampuran dan 15 jam di mesin pengolahan. Sementara itu untuk memproduksi sirup anggur diperlukan 15 jam di mesin pencampuran dan 12 jam di mesin pengolahan. Berapa unit sirup jeruk dan anggur yang diproduksi agar biaya produksi kedua sirup tersebut minimum.

3. $\text{Min } Z \quad 8X_1 + 2X_2$

$$\text{Kendala} \quad 2X_1 - 6X_2 \leq 12$$

$$5X_1 + 4X_2 \geq 40$$

$$X_1 + 2X_2 \geq 12$$

$$X_2 \leq 6$$

$$X_1, X_2 \text{ integer}$$

BAB VI

MODEL TRANSPORTASI

Model ini digunakan untuk memecahkan masalah transportasi, yaitu mengatur dan mendistribusikan produk dari sumber-sumber yang menyediakan produk ke tempat-tempat yang membutuhkan untuk mencapai efisiensi biaya transportasi. Alokasi produk harus memperhatikan biaya distribusi dari satu tempat ke tempat lain karena adanya perbedaan dari biaya-biaya tersebut.

Contoh kasus

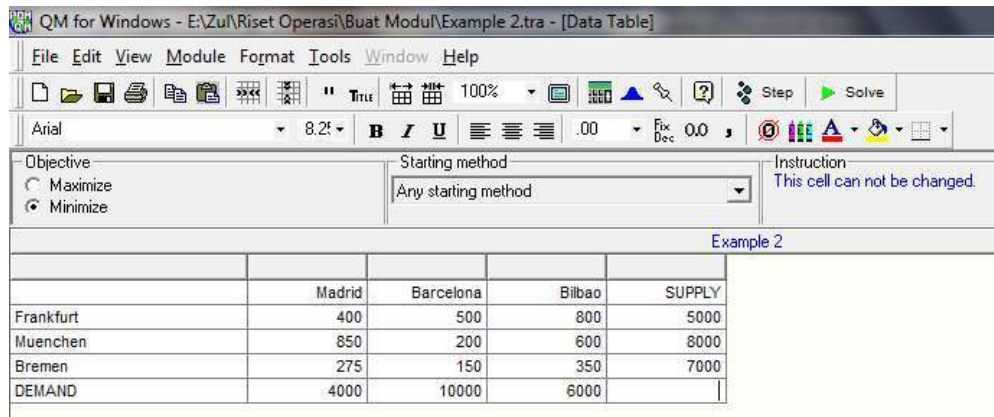
Moonbucks Co. saat ini memiliki tiga pabrik kopi yang berkapasitas berbeda-beda yakni dilokasi Frankfurt dengan kapasitas 5.000 unit, Muenchen 8.000 unit dan Bremen 7.000 unit. Perusahaan ini ingin mendistribusikan kopi hasil produksi tersebut ke tiga kota besar di Spanyol yakni Madrid, Barcelona dan Bilbao. Adapun berturut-turut estimasi permintaan kopi ke tiga kota tersebut adalah 4.000 unit, 10.000 unit dan 6.000 unit. Sedangkan untuk data biaya transportasi dicantumkan sebagai berikut:

- a. Frankfurt ke Madrid \$400, Barcelona \$500 dan Bilbao \$800.
- b. Muenchen ke Madrid \$850, Barcelona \$200 dan Bilbao \$600.
- c. Bremen ke Madrid \$275, Barcelona \$150 dan Bilbao \$350.

Berdasarkan data di atas, hitunglah biaya transportasi minimum dari perusahaan. (Gunakan sembarang metode/*Any starting method*).

Cara memasukkan data:

1. Klik Module – Transportation.
2. Klik File – New, sehingga muncul layar *Create data set for Transportation*.
3. Isikan Title, Number of Source (sumber), Number of Destination (tujuan), Objective (fungsi tujuan), Row name dan Column name (Row name dan Column name dapat diisi pada layar data set).
4. Klik Ok, lalu masukkan data seperti tampilan di bawah ini.



5. Klik Solve untuk melihat output.

Keterangan tampilan data set:

Objective. Menentukan fungsi tujuan

Shipping costs. Informasi utama mengenai biaya dari daerah asal ke daerah tujuan

Supplies. Kolom pada ujung kanan berisi jumlah persediaan daerah asal.

Demands. Baris demand berisi jumlah permintaan daerah tujuan.

Starting method. Empat pilihan dalam menyelesaikan metode transportasi:

1. Any method (QM biasanya menggunakan VAM)
2. Northwest corner method
3. Vogel's approximation method (VAM)
4. Minimum cost method

Output dan Analisis

Modul transportasi akan menampilkan enam layar output sebagai berikut:

Transportation Shipments

| Example 2 Solution | | | |
|------------------------------------|--------|-----------|--------|
| Optimal solution value = \$5950000 | Madrid | Barcelona | Bilbao |
| Frankfurt | 4000 | 1000 | |
| Muenchen | | 8000 | |
| Bremen | | 1000 | 6000 |

Marginal Costs

| Example 2 Solution | | | |
|--------------------|--------|-----------|--------|
| | Madrid | Barcelona | Bilbao |
| Frankfurt | | | 100 |
| Muenchen | 750 | | 200 |
| Bremen | 225 | | |

Final Solution Table

| Example 2 Solution | | | |
|--------------------|--------|-----------|--------|
| | Madrid | Barcelona | Bilbao |
| Frankfurt | 4000 | 1000 | [100] |
| Muenchen | [750] | 8000 | [200] |
| Bremen | [225] | 1000 | 6000 |

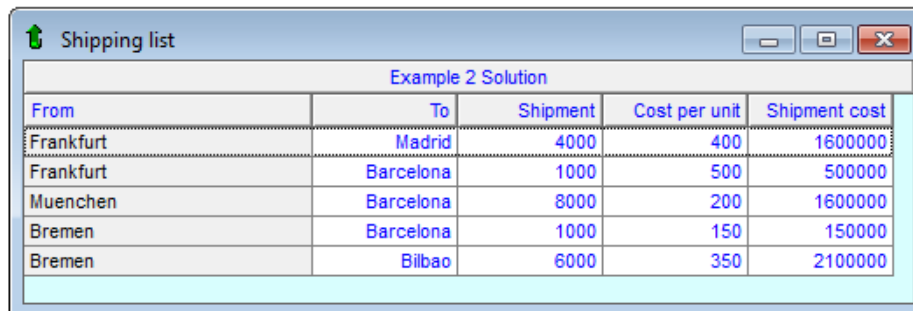
Iterations

| Example 2 Solution | | | |
|--------------------|--------|-----------|--------|
| | Madrid | Barcelona | Bilbao |
| Iteration 1 | | | |
| Frankfurt | 4000 | 1000 | (100) |
| Muenchen | (750) | 8000 | (200) |
| Bremen | (225) | 1000 | 6000 |

Shipments with costs

| Example 2 Solution | | | |
|--------------------|----------------|----------------|----------------|
| | Madrid | Barcelona | Bilbao |
| Frankfurt | 4000/\$1600000 | 1000/\$500000 | |
| Muenchen | | 8000/\$1600000 | |
| Bremen | | 1000/\$150000 | 6000/\$2100000 |

Shipping list



| Example 2 Solution | | | | |
|--------------------|-----------|----------|---------------|---------------|
| From | To | Shipment | Cost per unit | Shipment cost |
| Frankfurt | Madrid | 4000 | 400 | 1600000 |
| Frankfurt | Barcelona | 1000 | 500 | 500000 |
| Muenchen | Barcelona | 8000 | 200 | 1600000 |
| Bremen | Barcelona | 1000 | 150 | 150000 |
| Bremen | Bilbao | 6000 | 350 | 2100000 |

Dari table Transportation Shipments didapat biaya transportasi minimum dari tiga kota asal ke tiga kota tujuan adalah senilai \$5.950.000,-, dengan rincian biaya pengiriman seperti yang terlihat pada table shipments with costs dan shipping list.. Pengiriman kopi dari Frankfurt ke Madrid sebanyak 4.000 unit dengan biaya \$1.600.000,-, dari Frankfurt ke Barcelona sebanyak 1.000 unit dengan biaya pengiriman \$500.000,- dan seterusnya. Table marginal costs menerangkan tambahan biaya pengiriman dari kota asal ke kota tujuan. Pengiriman dari kota Frankfurt ke Bilbao akan meningkatkan biaya pengiriman sebesar \$100 per unit, sehingga kebutuhan kopi di kota Bilbao cukup dipenuhi dari kota Bremen. Begitu juga pengiriman kopi dari kota Muenchen ke Madrid akan meningkatkan biaya pengiriman sebesar \$750 per unit dan seterusnya.

Soal latihan

Dari contoh soal pada latihan, banyaknya permintaan kopi berubah sebagai berikut: Madrid 4.000 unit, Barcelona 9.200 unit dan Bilbao 5.000 unit, sedangkan persediaan kopi tidak berubah. Berdasarkan data tersebut, hitunglah biaya transportasi minimum dengan menggunakan ketiga metode transportasi.

BAB VII

MODEL PENUGASAN

Model penugasan digunakan untuk mengalokasikan pekerjaan kepada subjek/orang tertentu agar diperoleh hasil yang optimal (biaya minimal/keuntungan maksimal/waktu minimal, dan lain-lain). Alat analisis ini menggunakan metode Hungarian, dimana metode ini bersifat saling meniadakan, artinya apabila satu subjek telah mengerjakan satu pekerjaan maka tidak dapat mengerjakan pekerjaan lain (satu subjek mengerjakan satu pekerjaan dan satu pekerjaan dikerjakan oleh satu subjek).

Contoh kasus

Dalam upaya menciptakan penjualan maksimum, manajer pemasaran PT Agri Jaya menugaskan empat karyawan pilihannya untuk memasuki empat pasar sasaran. Data pendapatan karyawan pada masing-masing pasar (dalam jutaan rupiah) adalah sebagai berikut:

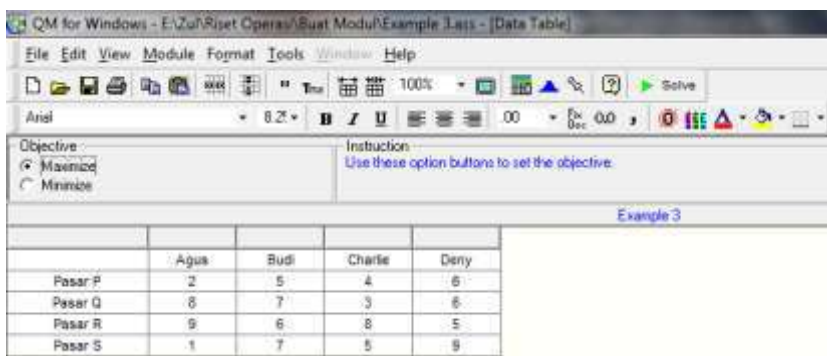
| Pasar | Karyawan | | | |
|-------|----------|------|---------|------|
| | Agus | Budi | Charlie | Deny |
| P | 2 | 5 | 4 | 6 |
| Q | 8 | 7 | 3 | 6 |
| R | 9 | 6 | 8 | 5 |
| S | 1 | 7 | 5 | 9 |

Bantulah manajer pemasaran tersebut dalam mengalokasikan karyawannya agar pendapatan maksimum dan berapa pendapatan maksimumnya.

Cara memasukkan data:

1. Klik Module – Assignment
2. Klik File – New, sehingga muncul tampilan *Create data set for Assignment*.
3. Isikan Title, Number of Jobs (pekerjaan), Number of Machines (mesin/subjek), Objective (fungsi tujuan), Row name dan Column name (Row name dan Column name dapat diisi pada layar data set).
4. Klik Ok, lalu masukkan data seperti tampilan di bawah ini.





5. Klik Solve untuk melihat output.

Output dan Analisis

Modul assignment akan menampilkan tiga layar output sebagai berikut:

Assignment List

| JOB | Assigned to | Profit |
|--------------|-------------|-----------|
| Pasar P | Budi | 5 |
| Pasar Q | Agus | 8 |
| Pasar R | Charlie | 8 |
| Pasar S | Deny | 9 |
| Total | | 30 |

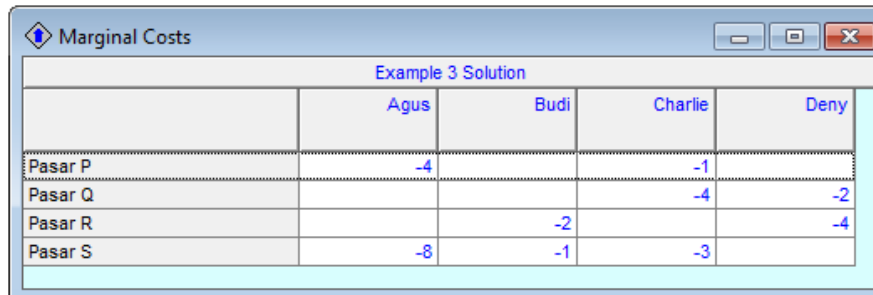
Assignments

| | Agus | Budi | Charlie | Deny |
|-------------------------------|----------|----------|----------|----------|
| Optimal solution value = \$30 | | | | |
| Pasar P | 2 | Assign 5 | 4 | 6 |
| Pasar Q | Assign 8 | 7 | 3 | 6 |
| Pasar R | 9 | 6 | Assign 8 | 5 |
| Pasar S | 1 | 7 | 5 | Assign 9 |

Dari table assignment list, manajer dapat melihat bahwa keuntungan maksimum yang akan didapat adalah 30 juta rupiah jika menempatkan keempat karyawannya sebagai berikut: Budi ditempatkan di pasar P dengan mendapatkan keuntungan 5 juta rupiah, Agus di pasar Q dengan keuntungan 8 juta rupiah, Charlie di pasar R mendapatkan 8

juta rupiah dan Deny di pasar S dengan keuntungan 9 juta rupiah. Tabel marginal costs memberikan informasi berkurangnya keuntungan jika para karyawan ditempatkan pada keempat pasar. Jika Agus ditempatkan di pasar P, perusahaan akan mengalami kerugian sebesar 4 juta rupiah dan seterusnya.

Marginal Costs



| | Agus | Budi | Charlie | Deny |
|---------|------|------|---------|------|
| Pasar P | -4 | | -1 | |
| Pasar Q | | | -4 | -2 |
| Pasar R | | -2 | | -4 |
| Pasar S | -8 | -1 | -3 | |

Soal latihan

Tabel berikut menunjukkan data untuk masalah penugasan. Tujuannya adalah untuk menetapkan setiap tenaga penjualan untuk wilayah dengan total biaya minimum. Catatan: Bruce tidak diijunkan bekerja di negara bagian Pennsylvania.

| | Mort | Cippy | Bruce | Beth | Lauren | Eddie | Brian |
|--------------|------|-------|-------|------|--------|-------|-------|
| Pennsylvania | 12 | 54 | * | 87 | 54 | 89 | 98 |
| New Jersey | 33 | 45 | 87 | 27 | 34 | 76 | 65 |
| New York | 12 | 54 | 76 | 23 | 87 | 44 | 62 |
| Florida | 15 | 37 | 37 | 65 | 26 | 96 | 23 |
| Canada | 42 | 32 | 18 | 77 | 23 | 55 | 87 |
| Mexico | 40 | 71 | 78 | 76 | 82 | 90 | 44 |
| Europe | 12 | 34 | 65 | 23 | 44 | 23 | 12 |

BAB VIII GOAL PROGRAMMING

Model goal programming (pemrograman bertujuan) sangat mirip dengan model pemrograman linear, hanya berbeda pada fungsi tujuan. Pemrograman linear memiliki satu tujuan, sedangkan goal programming dapat memiliki beberapa fungsi tujuan. Perhatikan contoh berikut.

Misalkan sebuah perusahaan memproduksi dua produk (x_1 dan x_2). Persyaratan sumber daya dan keuntungan ditunjukkan pada tabel berikut.

| | Produk 1 (x_1) | Produk 2 (x_2) | Available |
|---------------------|--------------------|--------------------|-----------|
| Keuntungan per unit | 16 | 12 | |
| Jam kerja per unit | 3 | 6 | 72 |
| Material per unit | 2 | 1 | 30 |

Selain itu, perusahaan memiliki tujuan sebagai berikut:

1. Total keuntungan setidaknya harus 250.
2. Butuh waktu untuk mengatur produksi untuk Produk 2, sehingga harus diproduksi dalam batch minimal 5.
3. Permintaan saat ini untuk Produk 1 adalah 14 unit, sehingga tepatnya 14 unit harus diproduksi.

Permasalahan ini sangat mirip dengan program linear, tapi mempunyai tiga tujuan, bukan satu tujuan. Goal programming ditentukan oleh jumlah variabel dan jumlah kendala atau tujuan. Jangan menghitung pembatasan non-negatif sebagai kendala. Dengan demikian, dalam contoh ini, ada dua variabel dan lima kendala atau tujuan (dua kendala dan tiga gol). Informasi yang dimasukkan seperti yang ditunjukkan berikut ini.

Cara memasukkan data:

1. Klik Module – Goal Programming
2. Klik File – New, sehingga muncul tampilan *Create data set for Goal Programming*.
3. Tentukan title dari kasus, jumlah batasan (*constraint*), jumlah variable, nama baris dan kolom data set yang akan diisi. Untuk contoh, title: Example 5, jumlah

kendala: 5, jumlah variable: 2. Nama baris dan kolom bisa dipilih dari beberapa pilihan atau diisi pada tampilan data set.

4. Klik Ok, sehingga muncul tampilan data set.
5. Isi data set dan klik Solve untuk melihat hasil output.

| Goal Programming - Goals ordered by priorities | | | | | | | |
|--|--------|----------|--------|----------|----|----|-------|
| | Wt(d+) | Prty(d+) | Wt(d-) | Prty(d-) | X1 | X2 | RHS |
| Labor hours | 0 | 0 | 0 | 0 | 3 | 6 | <= 72 |
| Materials | 0 | 0 | 0 | 0 | 2 | 1 | <= 30 |
| Product 1 - demand | 1 | 1 | 1 | 1 | 1 | 0 | = 14 |
| Profit | 0 | 0 | 1 | 2 | 16 | 12 | = 250 |
| Product 2 - min batch size | 0 | 0 | 1 | 3 | 0 | 1 | = 5 |

Beberapa informasi identik dengan pemrograman linear, tetapi ada beberapa perbedaan, (1) tidak ada fungsi tujuan, (2) ada empat kolom tambahan di awal (kiri) tabel sebelum variabel keputusan. Kolom-kolom tambahan ini digunakan untuk tujuan dan bukan untuk kendala (dapat dilihat bahwa mereka adalah 0).

Tujuan/kendala (*Goals/Constraints*) dalam setiap baris tabel. Dua baris pertama merupakan kendala, karena ini adalah kendala, maka empat kolom di kiri tidak digunakan (masukkan 0). Tiga baris selanjutnya merupakan tujuan, dan ada dua hal yang harus diperhatikan. Karena ini adalah tujuan, maka tanda berturut-turut adalah "=". Nilai-nilai di bawah x_1 dan x_2 berfungsi untuk menciptakan tujuan dalam hubungannya dengan variabel d_+ dan d_- , yang menunjukkan berapa banyak kita bisa capai lebih atau di bawah tujuan.

Misalnya, baris 3 dalam tabel untuk $x_1 - (d_{1+}) + (d_{1-}) = 14$.

Jika x_1 di bawah 14, d_{1-} merupakan jumlah di bawah ini, tetapi jika x_1 di atas 14, d_{1+} merupakan jumlah lebih.

Demikian pula, baris berikutnya (baris 4) terdiri dari:

$$16x_1 + 12x_2 - (d_{2+}) + (d_{2-}) = 250.$$

Dengan demikian, d_{2+} dan d_{2-} mewakili jumlah profit di atas 250 dan di bawah 250.

Secara matematis, tujuannya adalah d_{1+} , d_{1-} , d_{2+} , d_{2-} , d_{3+} , d_{3-} .

Bagaimana kita mengurutkan keenam tujuan ini berdasarkan tingkat prioritas atau kepentingannya?

Prioritas dan bobot: Pertama, ada enam gol atau tujuan (d_{1+} , d_{1-} , d_{2+} , d_{2-} , d_{3+} , d_{3-}) dalam contoh ini, tapi itu tidak masalah jika tujuan keuntungan 250 adalah melebihi yang diperkirakan, atau jika lebih dari lima unit Produk 2 yang diproduksi.

Oleh karena itu, baik bobot dan prioritas dari dua gol ini telah diatur ke 0. Prioritas untuk empat gol lainnya berkisar dari 1 sampai 3 Yang dimaksud dengan prioritas yang berbeda adalah urutan tujuan yang memuaskan. Dengan kata lain, tujuan dengan Prioritas 1 harus dicapai sebelum tujuan dengan Prioritas 2, yang harus dipenuhi sebelum gol dengan Prioritas 3, dan seterusnya. Dalam contoh ini, 14 unit Produk 1 harus dilakukan, maka menjamin tingkat keuntungan minimal 250, maka menjamin tingkat bets minimal lima untuk Produk 2.

Output dan analisis

Layar berikut akan menampilkan solusi ringkas (juga tersedia tabel simpleks goal untuk ditampilkan, seperti grafik untuk masalah dua dimensi).

| Goal Programming - Goals ordered by priorities solution | | | | |
|---|--|-----------------------|-------------------|-------------------|
| Item | | | | |
| Decision variable analysis | | Value | | |
| X1 | | 14 | | |
| X2 | | 2 | | |
| Priority analysis | | Nonachievement | | |
| Priority 1 | | 0 | | |
| Priority 2 | | 2 | | |
| Priority 3 | | 3 | | |
| Constraint Analysis | | RHS | d+ (row i) | d- (row i) |
| Labor hours | | 72 | 0 | 18 |
| Materials | | 30 | 0 | 0 |
| Product 1 - demand | | 14 | 0 | 0 |
| Profit | | 250 | 0 | 2 |
| Product 2 - min batch size | | 5 | 0 | 3 |

Solusi optimal didapat dengan memproduksi 14 unit Produk 1 dan 2 unit Produk 2. Prioritas 1 akan tercapai (nonachievement adalah 0), sedangkan prioritas 2 dan 3 tidak akan tercapai. Ingat, $d+$ adalah jumlah dimana tujuan telah terlampaui, dan $d-$ adalah jumlah dibawah tujuan. Analisis kendala menunjukkan bahwa 18 jam digunakan lebih sedikit daripada yang tersedia; jumlah material yang tersedia habis digunakan; tujuan 3 tercapai; keuntungan (tujuan/kendala 4) kurang 2 dari yang diharapkan untuk dicapai dan tujuan/kendala 5 ukuran bets 3 berada di bawah yang diharapkan dicapai.

Soal Latihan

Perhatikan kasus pemrograman linear berikut,

$$\text{Min} \quad 2X_1 + 3X_2$$

$$\text{Kendala} \quad 2X_1 + X_2 \geq 10$$

$$X_1 + 2X_2 \leq 14$$

$$X_1 + 4X_2 \geq 12$$

$$X_1 - 8X_2 \leq 0$$

$$X_1 \leq 10$$

$$X_1 + X_2 = 10$$

$$X_1, X_2 \geq 0$$

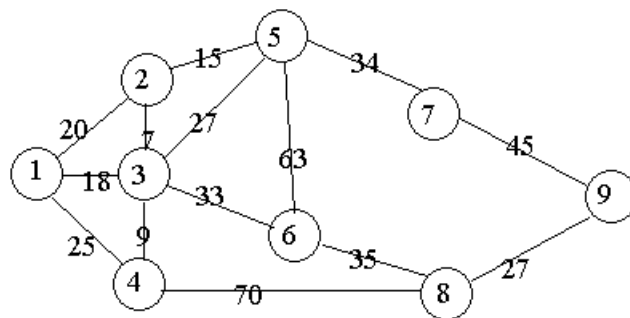
Ubah kasus pemrograman linear tersebut menjadi kasus goal programming sehingga semua kendala fungsional menjadi kendala-kendala sasaran dengan memperhatikan urutan prioritas sebagai berikut:

- Peminimuman biaya produksi yang tercermin pada fungsi tujuan.
- Produksi X_1 tidak boleh kurang dari 8 unit.
- Kendala yang lain mempunyai prioritas yang sama.

BAB IX NETWORKS

Jaringan (networks) merupakan sebuah istilah untuk menandai model-model yang secara visual bisa diidentifikasi sebagai sebuah system jaringan yang terdiri dari rangkaian noda (*node*) dan kegiatan (*activity*). Model visual ini sangat bermanfaat untuk memahami sebuah system yang sedang dianalisis.

Ada tiga model yang dibahas dalam modul ini: minimum spanning tree (pohon rentang minimum), shortest route (jalur terpendek), dan maximal flow (aliran maksimal). Diagram berikut digunakan untuk ketiga contoh. Untuk memulai salah satu dari tiga submodel, perlu untuk menunjukkan jumlah cabang. Dalam contoh ini ada 14 cabang. Untuk memasukkan setiap cabang, harus diberikan simpul awal dan simpul akhir.



A. Minimum Spanning Tree

Tujuan minimum spanning tree adalah untuk meminimalkan total biaya. Pada minimum spanning tree, n node harus terhubung satu sama lain menggunakan $n - 1$ cabang yang menggambarkan biaya.

Cara memasukkan data:

6. Klik Module – Networks.
7. Klik File – New – Minimum Spanning Tree, sehingga muncul tampilan *Create data set for Networks/Minimum Spanning Tree*.
8. Tentukan title dari kasus, jumlah cabang (*branches*), dan nama baris data yang akan diisi. Untuk contoh, title: Example 6, jumlah cabang: 14. Nama baris bisa dipilih dari beberapa pilihan atau diisi pada tampilan data set.

9. Klik Ok, sehingga muncul tampilan data set.
10. Isi data set dan klik Solve untuk melihat hasil output.

| Starting node for iterations | | Instruction | |
|--------------------------------------|------------|---|------|
| ◀ <input type="text" value="1"/> ▶ | | Enter the starting node for this branch. Node numbers are integers between 1 and 90 | |
| Minimum Spanning Tree Example | | | |
| Branch name | Start node | End node | Cost |
| Branch 1 | 1 | 2 | 20 |
| Branch 2 | 1 | 3 | 18 |
| Branch 3 | 1 | 4 | 25 |
| Branch 4 | 2 | 3 | 7 |
| Branch 5 | 2 | 5 | 15 |
| Branch 6 | 3 | 4 | 9 |
| Branch 7 | 3 | 5 | 27 |
| Branch 8 | 3 | 6 | 33 |
| Branch 9 | 4 | 8 | 70 |
| Branch 10 | 5 | 6 | 63 |
| Branch 11 | 5 | 7 | 34 |
| Branch 12 | 6 | 8 | 35 |
| Branch 13 | 7 | 9 | 45 |
| Branch 14 | 8 | 9 | 27 |

Output dan analisis

Ada dua output dari analisis minimum spanning tree yaitu networks result dan solution steps.

| Starting node for iterations | | Instruction | | | | |
|--|------------|--|------|---------|------|--|
| ◀ <input type="text" value="1"/> ▶ | | There are more results available in additional windows. These may be opened by using the WINDOW option in the Main Menu. | | | | |
| Networks Results | | | | | | |
| Minimum Spanning Tree Example Solution | | | | | | |
| Branch name | Start node | End node | Cost | Include | Cost | |
| Branch 1 | 1 | 2 | 20 | | | |
| Branch 2 | 1 | 3 | 18 | Y | 18 | |
| Branch 3 | 1 | 4 | 25 | | | |
| Branch 4 | 2 | 3 | 7 | Y | 7 | |
| Branch 5 | 2 | 5 | 15 | Y | 15 | |
| Branch 6 | 3 | 4 | 9 | Y | 9 | |
| Branch 7 | 3 | 5 | 27 | | | |
| Branch 8 | 3 | 6 | 33 | Y | 33 | |
| Branch 9 | 4 | 8 | 70 | | | |
| Branch 10 | 5 | 6 | 63 | | | |
| Branch 11 | 5 | 7 | 34 | Y | 34 | |
| Branch 12 | 6 | 8 | 35 | Y | 35 | |
| Branch 13 | 7 | 9 | 45 | | | |
| Branch 14 | 8 | 9 | 27 | Y | 27 | |
| Total | | | | | 178 | |

Dari output didapat delapan cabang yang menghubungkan seluruh node (kolom include yang ditandai dengan Y) dan biaya minimum menghubungkan sembilan node adalah 178. Tabel menampilkan urutan cabang-cabang yang ditambahkan diilustrasikan pada layar berikut:

| Solution steps | | | | |
|--|---------------|-------------|------|-----------------|
| Minimum Spanning Tree Example Solution | | | | |
| Branch | Starting node | Ending node | Cost | Cumulative cost |
| Branch 2 | 1 | 3 | 18 | 18 |
| Branch 4 | 2 | 3 | 7 | 25 |
| Branch 6 | 3 | 4 | 9 | 34 |
| Branch 5 | 2 | 5 | 15 | 49 |
| Branch 8 | 3 | 6 | 33 | 82 |
| Branch 11 | 5 | 7 | 34 | 116 |
| Branch 12 | 6 | 8 | 35 | 151 |
| Branch 14 | 8 | 9 | 27 | 178 |

B. Shortest Route

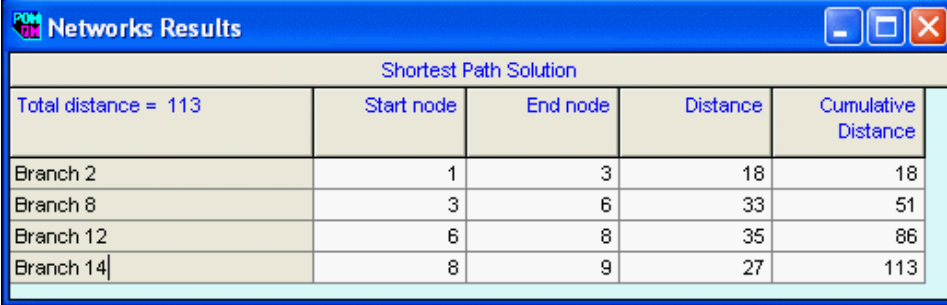
Tujuan shortest route adalah untuk menemukan jalan atau jarak terpendek dari satu node ke node lain. Contoh kasus ditunjukkan pada gambar sebelumnya. Program akan mencari jalan terpendek dari nomor simpul minimum ke nomor simpul maksimum (1 sampai 9) dalam contoh ini.

Cara memasukkan data:

1. Klik Module – Networks.
2. Klik File – New – Shortest Route, sehingga muncul tampilan *Create data set for Networks/Shortest Route*.
3. Tentukan title dari kasus, jumlah cabang (*branches*), tipe jaringan: undirect atau direct; dan nama baris data yang akan diisi. Untuk contoh, title: Example 7, jumlah cabang: 14. Nama baris bisa dipilih dari beberapa pilihan atau diisi pada tampilan data set.
4. Klik Ok, sehingga muncul tampilan data set.
5. Isi data set dan klik Solve untuk melihat hasil output.

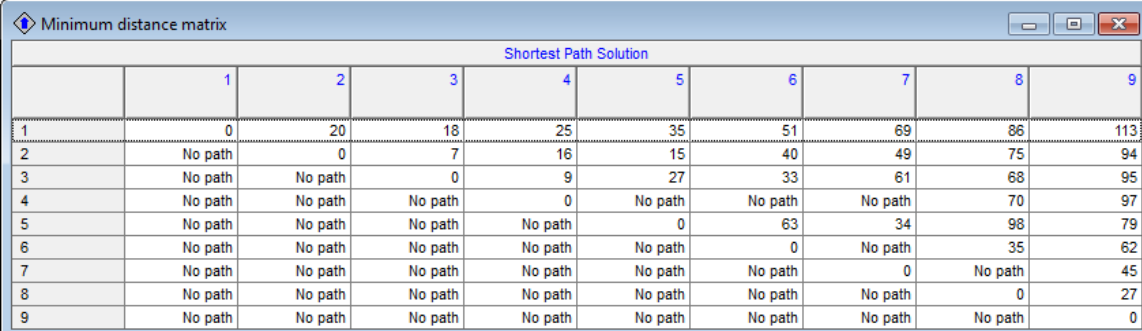
| Network type | | Origin | Destination | Instruction |
|---|--------------------------------|----------|-------------|--|
| <input checked="" type="radio"/> Undirected | <input type="radio"/> Directed | 0 | 0 | Use the scroll bar or the text box to enter the destination. Setting at 0 will use the minimum to maximum node |
| Shortest Path | | | | |
| | Start node | End node | Distance | |
| Branch 1 | 1 | 2 | 20 | |
| Branch 2 | 1 | 3 | 18 | |
| Branch 3 | 1 | 4 | 25 | |
| Branch 4 | 2 | 3 | 7 | |
| Branch 5 | 2 | 5 | 15 | |
| Branch 6 | 3 | 4 | 9 | |
| Branch 7 | 3 | 5 | 27 | |
| Branch 8 | 3 | 6 | 33 | |
| Branch 9 | 4 | 8 | 70 | |
| Branch 10 | 5 | 6 | 63 | |
| Branch 11 | 5 | 7 | 34 | |
| Branch 12 | 6 | 8 | 35 | |
| Branch 13 | 7 | 9 | 45 | |
| Branch 14 | 8 | 9 | 27 | |

Output dan analisis



| Shortest Path Solution | | | | |
|------------------------|------------|----------|----------|---------------------|
| Total distance = 113 | Start node | End node | Distance | Cumulative Distance |
| Branch 2 | 1 | 3 | 18 | 18 |
| Branch 8 | 3 | 6 | 33 | 51 |
| Branch 12 | 6 | 8 | 35 | 86 |
| Branch 14 | 8 | 9 | 27 | 113 |

Empat cabang dimasukkan dalam jalur terpendek, yaitu jalur 1-3-6-8-9, dengan total jarak 113. Selain itu, program ini menghitung total jarak minimum dari setiap node ke node lain sebagai berikut.



| Shortest Path Solution | | | | | | | | | |
|------------------------|---------|---------|---------|---------|---------|---------|---------|---------|-----|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1 | 0 | 20 | 18 | 25 | 35 | 51 | 69 | 86 | 113 |
| 2 | No path | 0 | 7 | 16 | 15 | 40 | 49 | 75 | 94 |
| 3 | No path | No path | 0 | 9 | 27 | 33 | 61 | 68 | 95 |
| 4 | No path | No path | No path | 0 | No path | No path | No path | 70 | 97 |
| 5 | No path | No path | No path | No path | 0 | 63 | 34 | 98 | 79 |
| 6 | No path | No path | No path | No path | No path | 0 | No path | 35 | 62 |
| 7 | No path | No path | No path | No path | No path | No path | 0 | No path | 45 |
| 8 | No path | No path | No path | No path | No path | No path | No path | 0 | 27 |
| 9 | No path | No path | No path | No path | No path | No path | No path | No path | 0 |

C. Maximal Flow

Tujuan maximal flow adalah untuk memaksimalkan aliran dari awal (source) sampai akhir (sink). Nilai masing-masing cabang mewakili kapasitasnya, dan angka kedua merupakan reverse capacity (kapasitas dalam arah yang berlawanan).

Cara memasukkan data:

1. Klik Module – Networks.
2. Klik File – New – Maximal Flow, sehingga muncul tampilan *Create data set for Networks/Maximal Flow*.
3. Tentukan title dari kasus, jumlah cabang (*branches*), dan nama baris data yang akan diisi. Untuk contoh, title: Example 7, jumlah cabang: 14. Nama baris bisa dipilih dari beberapa pilihan atau diisi pada tampilan data set.
4. Klik Ok, sehingga muncul tampilan data set.
5. Isi data set dan klik Solve untuk melihat hasil output.

Output dan analisa.

Di bagian atas output, source dan sink dapat diatur.

Source: 1 Sink: 9 Instruction: There are more results available in additional windows. These may be opened by using the WINDOW option in the Main Menu.

Networks Results

Maximal Flow Solution

| Branch name | Start node | End node | Capacity | Reverse capacity | Flow |
|----------------------|------------|----------|----------|------------------|------|
| Maximal Network Flow | 61 | | | | |
| Branch 1 | 1 | 2 | 20 | 0 | 18 |
| Branch 2 | 1 | 3 | 18 | 0 | 18 |
| Branch 3 | 1 | 4 | 25 | 0 | 25 |
| Branch 4 | 2 | 3 | 7 | 0 | 3 |
| Branch 5 | 2 | 5 | 15 | 0 | 15 |
| Branch 6 | 3 | 4 | 9 | 0 | 2 |
| Branch 7 | 3 | 5 | 27 | 0 | 19 |
| Branch 8 | 3 | 6 | 33 | 0 | 0 |
| Branch 9 | 4 | 8 | 70 | 0 | 27 |
| Branch 10 | 5 | 6 | 63 | 0 | 0 |
| Branch 11 | 5 | 7 | 34 | 0 | 34 |
| Branch 12 | 6 | 8 | 35 | 0 | 0 |
| Branch 13 | 7 | 9 | 45 | 0 | 34 |
| Branch 14 | 8 | 9 | 27 | 0 | 27 |

Jumlah aliran maksimal adalah 61, dan arus sepanjang cabang dapat dilihat pada gambar. Iterasi diberikan pada layar berikut. Harap dicatat bahwa ada beberapa langkah iterasi yang berbeda yang dapat diambil untuk sampai pada aliran maksimal yang sama.

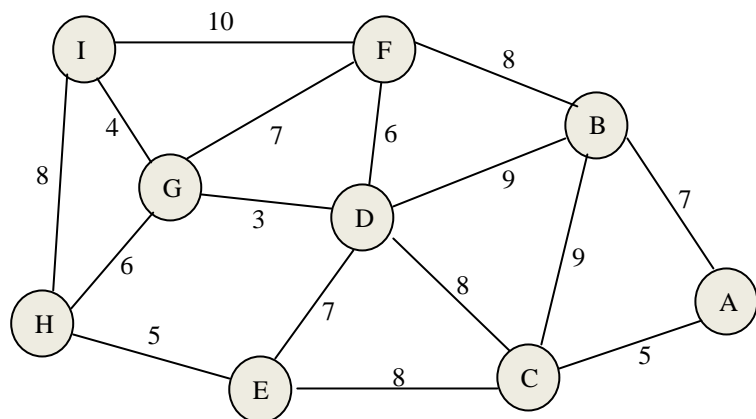
Iterations

Maximal Flow Solution

| Iteration | Path | Flow | Cumulative Flow |
|-----------|-----------------------|------|-----------------|
| 1 | 1-> 4-> 8-> 9 | 25 | 25 |
| 2 | 1-> 3-> 5-> 7-> 9 | 18 | 43 |
| 3 | 1-> 2-> 5-> 7-> 9 | 15 | 58 |
| 4 | 1-> 2-> 3-> 4-> 8-> 9 | 2 | 60 |
| 5 | 1-> 2-> 3-> 5-> 7-> 9 | 1 | 61 |

Tugas Latihan

Bagaimana cara terbaik untuk menghubungkan seluruh node di dalam system jaringan berikut?



BAB X GAME THEORY

Teori permainan secara umum digunakan untuk menyelesaikan masalah yang berkaitan dengan tindakan sebuah unit bisnis (misalnya) untuk memenangkan persaingan dalam usaha yang digelutinya. Seperti diketahui, bahwa dalam praktek sehari-hari, setiap unit usaha atau organisasi pada umumnya harus berhadapan dengan para pesaing. Untuk memenangkan persaingan itulah, diperlukan analisis dan pemilihan strategi pemasaran tepat, khususnya strategi bersaing yang paling optimal bagi unit usaha atau organisasi yang bersangkutan.

Sebuah permainan zero sum ditunjukkan oleh tabel yang memberikan payoff (hadiah) kepada pemain baris (Pemain 1) dari pemain kolom (Pemain 2). Tabel permainan memiliki satu baris untuk masing-masing strategi pemain 1 dan satu kolom untuk masing-masing strategi pemain 2. Dalam hal ini, strategi yang diterapkan bisa jadi seperti menetapkan harga murah, sedang atau mahal dari suatu produk; atau penamaan lain tergantung dari permainan.

Perhatikan table matriks permainan antar dua pemain berikut:

| | | Pemain 2 | | | | |
|----------|------------|------------|------------|------------|------------|------------|
| | | Strategy 1 | Strategy 2 | Strategy 3 | Strategy 4 | Strategy 5 |
| Pemain 1 | Strategy 1 | 10 | -12 | 34 | 75 | 67 |
| | Strategy 2 | 38 | 57 | 96 | 28 | -33 |

Jika pemain 1 memilih Strategi 1 dan pemain 2 memilih Strategi 1, pemain 2 membayar pemain 1 sebesar 10. Jika pemain 1 memilih Strategi 1 dan pemain 2 memilih Strategi 2, pemain kolom membayar -12 atau, dengan kata lain, pemain 2 menerima 12 dari pemain 1. Pemain 1 dan pemain 2 masing-masing harus memilih strategi tanpa mengetahui strategi apa yang dipilih lawan. Ada kalanya para pemain memilih strategi murni (*pure strategy*) dan ada kalanya memilih strategi campuran (*mixed strategy*).

Cara memasukkan data:

1. Klik Module – Game Theory
2. Klik File – New, sehingga muncul tampilan *Create data set for Game Theory*.
3. Tentukan title dari kasus, Number of Row Strategies (jumlah strategi baris), Number of Column Strategies (jumlah strategi kolom), nama baris dan kolom data set yang akan diisi. Untuk contoh, title: Example 8, Number of Row Strategies: 2, Number of Column Strategies: 5. Nama baris dan kolom bisa dipilih dari beberapa pilihan atau diisi pada tampilan data set.
4. Klik Ok, sehingga muncul tampilan data set.
5. Isi data set dan klik Solve untuk melihat hasil output.

Instruction
Enter the name for this column strategy. Almost any character is permissible. maximum number of decimals to

Example

| | Col strat1 | Col strat2 | Col strat3 | Col strat4 | Col strat 5 |
|----------------|------------|------------|------------|------------|-------------|
| Row strategy 1 | 10 | -12 | 34 | 75 | 67 |
| Row strategy 2 | 38 | 57 | 96 | 28 | -33 |

Output dan analisis

Hasil dari contoh di atas ditunjukkan dalam tampilan berikut:

| Example Solution | | | | | | |
|------------------------|------------|------------|------------|------------|-------------|---------|
| | Col strat1 | Col strat2 | Col strat3 | Col strat4 | Col strat 5 | Row Mix |
| Row strategy 1 | 10 | -12 | 34 | 75 | 67 | .5325 |
| Row strategy 2 | 38 | 57 | 96 | 28 | -33 | .4675 |
| Column Mix---> | 0 | .5917 | 0 | 0 | .4083 | |
| Value of game (to row) | 20.2544 | | | | | |

Pemain 1 harus memainkan strategi 1 sebesar 53,25 persen dan strategi 2 sebesar 46,75 persen. Pemain 2 harus memainkan strategi 2 sebesar 59,17 persen dan strategi 5 sebesar 40,83 persen dan tidak menerapkan Strategi 1, 3, atau 4. Jika mereka mengikuti campuran ini, (diharapkan) nilai dari permainan ini adalah pemain 2 akan membayar pemain 1 sebesar 20,2544. Artinya, jika mereka memainkan permainan ini dalam waktu jangka waktu tertentu, pemain 1 akan mendapatkan *payoff* sebesar -12, 67, 57, dan -33, dan rata-rata mencapai 20,2544.

Maximin and Minimax

Nilai maximin (nilai terbesar dari yang terkecil) untuk pemain 1, akan muncul dalam kolom berlabel "Row minimum" yaitu -12 dan -33 pada gambar berikut. Maka nilai yang diambil adalah -12.

Nilai minimax (nilai terkecil dari yang terbesar) untuk pemain 2 muncul di baris berlabel "Column maximum" adalah 38, 57, 96, 75, dan 67. Nilai minimax adalah yang terbaik (terendah) adalah 38. Nilai dari permainan ini adalah antara maximin dan minimax sebagai muncul dalam game ini, dengan nilai 20,2544, yaitu antara -12 dan 38.

| Example Solution | | | | | | | |
|--------------------|------------|------------|------------|------------|-------------|-------------|---------|
| | Col strat1 | Col strat2 | Col strat3 | Col strat4 | Col strat 5 | Row Minimum | Maximin |
| Row strategy 1 | 10 | -12 | 34 | 75 | 67 | -12 | -12 |
| Row strategy 2 | 38 | 57 | 96 | 28 | -33 | -33 | |
| Column Maximum | 38 | 57 | 96 | 75 | 67 | | |
| Minimax | 38 | | | | | | |
| -12 <= value <= 38 | | | | | | | |

Expected Values for Row

Gambar berikut menampilkan perhitungan (perkalian) nilai yang diharapkan untuk masing-masing strategi pemain 1. Karena pemain 1 harus menggunakan kedua strategi, nilai-nilai yang diharapkan adalah sama dan sesuai dengan nilai yang diharapkan dari permainan.

| Example Solution | | | | | | |
|------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|--------------------------|
| | Col mix 1 * cell payoff | Col mix 2 * cell payoff | Col mix 3 * cell payoff | Col mix 4 * cell payoff | Col mix 5 * cell payoff | Expected Value (row sum) |
| Column's Optimal Mix | 0 | 5917 | 0 | 0 | 4083 | |
| Row strategy 1 | 0 | -7.1006 | 0 | 0 | 27.355 | 20.2544 |
| Row strategy 2 | 0 | 33.7278 | 0 | 0 | -13.4734 | 20.2544 |
| Value of game (to row) | | | | | | 20.2544 |

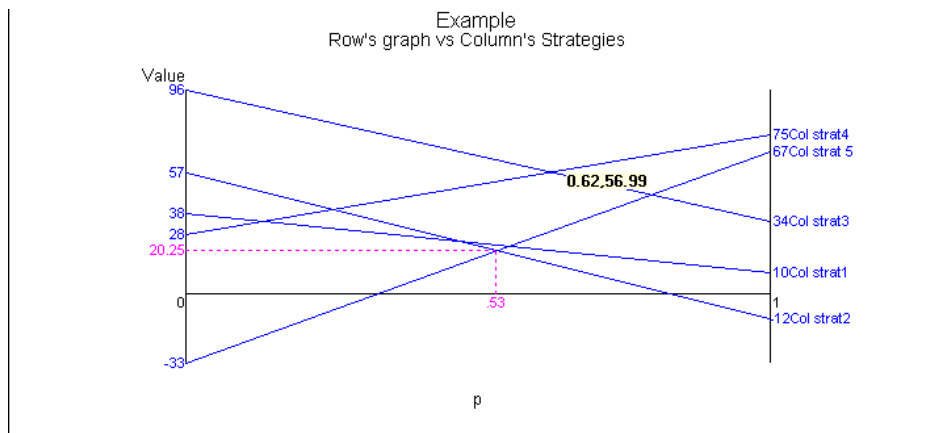
Expected Values for Column

Demikian pula, jika pemain 2 memainkan strategi 2 atau 5, pemain 2 akan mencapai nilai permainan. Namun, jika pemain 2 memilih strategi 1, 3, atau 4, maka ia akan

membayar lebih dari nilai permainan seperti yang ditunjukkan oleh nilai yang diharapkan pada gambar berikut.

| Column's Expected Values | | | | | | |
|--------------------------|-----------------|------------|------------|------------|------------|-------------|
| Example Solution | | | | | | |
| | Optimal Row Mix | Col strat1 | Col strat2 | Col strat3 | Col strat4 | Col strat 5 |
| Row 1 mix * cell payoff | .5325 | 5.3254 | -6.3905 | 18.1065 | 39.9408 | 35.6805 |
| Row 2 mix * cell payoff | .4675 | 17.7633 | 26.645 | 44.8757 | 13.0688 | -15.426 |
| Expected Value (Col sum) | | 23.0888 | 20.2544 | 62.9823 | 53.0296 | 20.2544 |
| Value of game (to row) | 20.2544 | | | | | |

Grafik hanya akan ditampilkan jika salah satu atau kedua pemain memiliki paling banyak dua strategi.



BAB XI WAITING LINES

Waiting Lines disebut juga model antrian, memungkinkan untuk penentuan biaya rata-rata operasi sistem antrian, di mana biaya adalah jumlah dari biaya tenaga kerja dan biaya menunggu dibebankan kepada sistem waktu (jumlah dalam sistem) atau waktu tunggu (nomor menunggu).

Kerangka untuk antrian tergantung pada model tertentu yang digunakan. Ada sembilan model diuraikan sini, dan masing-masing model dapat digunakan dengan atau tanpa biaya. Secara umum, data yang diperlukan akan bervariasi mengikuti perubahan model yang dipilih di awal.

Dari sembilan model yang tersedia, beberapa model adalah kasus khusus dari model lainnya. Secara khusus, semua model server tunggal adalah kasus khusus dari model multi-server yang bersesuaian. Model-model tersebut dijelaskan seperti dibawah ini. Semua model mengasumsikan proses kedatangan Poisson.

$M/M/1$ — waktu layanan eksponensial, 1 server (model server tunggal)

$M/D/1$ — waktu layanan konstan, 1 server (model layanan konstan)

$M/G/1$ — waktu layanan umum, 1 server

$M/E_k/1$ — waktu layanan Erlang-k, 1 server

$M/M/s$ — waktu layanan eksponensial, 1 atau lebih server (model multiple server)

$M/M/1$ dengan antrian terbatas (atau sistem terbatas)

$M/M/s$ dengan antrian terbatas (atau sistem terbatas)

$M/M/1$ dengan populasi terbatas

$M/M/s$ dengan populasi terbatas

Parameter M pertama dalam notasi mengacu pada proses kedatangan. M singkatan dari Memorylessness, yang berarti proses kedatangan Poisson. Parameter M kedua mengacu pada proses pelayanan. M singkatan dari memoryless, yang berarti bahwa waktu layanan mengikuti distribusi eksponensial. D singkatan dari deterministik, yang digunakan ketika waktu layanan yang konstan (selalu sama). G singkatan dari general (umum), dan E_k singkatan distribusi Erlang-k. Parameter ketiga adalah

jumlah server (juga disebut saluran). Perhatikan bahwa s dapat diatur menjadi 1 pada model $M/M/s$ untuk menyelesaikan model $M/M/1$ model.

Contoh tampilan data.

| Cost analysis <input checked="" type="radio"/> No costs <input type="radio"/> Use Costs | | Time unit (arrival, service rate) hours | Instruction Enter the value for the arrival rate(λ). This must be a strictly positive value. |
|---|-------|--|---|
| sample | | | |
| Parameter | Value | | |
| MM/1 (exponential service) | | | |
| Arrival rate(λ) | 26 | | |
| Service rate(μ) | 30 | | |
| Number of servers | 1 | | |

- **Arrival rate (λ).** Setiap sistem antrian harus memiliki tingkat kedatangan pelanggan. Jumlah ini adalah *rate*, yang berarti bahwa satuan waktu (jam, hari, dll) dikaitkan dengan tingkat kedatangan. Hal ini penting karena unit waktu harus sesuai dengan satuan waktu dari parameter berikutnya.
- **Service rate (μ).** Jumlah yang akan dimasukkan adalah rate di mana setiap server memproses pelanggan. Perhatikan bahwa ini adalah tingkat yang artinya adalah waktu pelayanan dan satuan waktu dari tingkat ini harus sesuai dengan satuan waktu dari tingkat kedatangan.
- **Number of servers.** Nilai standar dan minimum untuk number of servers adalah 1
- **Time unit,** digunakan untuk (1) mengingatkan Anda bahwa tingkat kedatangan dan laju pelayanan keduanya harus didasarkan pada satuan waktu yang sama. (2) jika Anda memilih jam, tampilan output akan menampilkan "menit" dan "detik." Jika tidak, tampilan output akan menampilkan "60 * kali menjawab."

A. Model $M/M/1$

Tingkat kedatangan pelanggan adalah 26 orang per jam berdasarkan proses kedatangan Poisson. Ada satu server yang melayani pelanggan dalam waktu rata-rata 2 menit berdasarkan distribusi eksponensial.

Layar output untuk sistem ini berikut. Perhatikan bahwa tingkat kedatangan dimasukkan 26, seperti yang diberikan dalam contoh. Waktu pelayanan 2 menit harus dikonversi ke tingkat 30 per jam.

| sample solution | | | | | |
|-----------------------------|-------|---------------------------------------|--------|---------|---------|
| Parameter | Value | Parameter | Value | Minutes | Seconds |
| M/M/1 (exponential service) | | Average server utilization | .8667 | | |
| Arrival rate(λ) | 26 | Average number in the queue(L_q) | 5.6333 | | |
| Service rate(μ) | 30 | Average number in the system(L_s) | 6.5 | | |
| Number of servers | 1 | Average time in the queue(W_q) | .2167 | 13 | 780 |
| | | Average time in the system(W_s) | .25 | 15 | 900 |

Average server utilization. Rata-rata persentase waktu tiap server dalam keadaan sibuk. Dalam contoh, server sibuk adalah 87 persen (waktu dalam satuan jam).

Average number in the queue (line). Rata-rata jumlah pelanggan di dalam system sedang menunggu layanan. Dalam contoh, rata-rata ada 5,63 pelanggan sedang menunggu layanan

Average number of customers in the system. Rata-rata jumlah pelanggan baik yang sedang antri atau sedang dilayani. Dalam contoh ini, rata-rata ada 6,5 pelanggan dalam system.

Average time in the queue (line). Rata-rata waktu tunggu pelanggan sebelum dilayani. Satuan waktu adalah sama dengan tingkat kedatangan dan pelayanan. Dalam contoh ini 0,2167 jam.

Average time in the system. Rata-rata waktu yang dihabiskan pelanggan untuk menunggu dan dilayani. Dalam contoh ini 0,25 jam.

Sering kali kita ingin mengubah waktu dari jam ke menit atau dari menit ke detik. Waktu rata-rata akan dikalikan dengan 60 atau 3.600, dan jawaban akan menunjukkan di samping rata-rata asli. Angka-angka di sana mengungkapkan waktu yang sama tetapi dengan unit menit, karena kali asli berada di jam.

Tabel probabilitas menunjukkan peluang (persentase waktu) k pelanggan dalam sistem, probabilitas kumulatif dari k atau lebih sedikit pelanggan yang sedang berada di dalam system. Seperti contoh, peluang 3 pelanggan yang berada di dalam system 0.0868, peluang lebih sedikit atau sama dengan 3 pelanggan yang berada dalam system 0.4358. Peluang tepat lebih besar dari 3 pelanggan yang berada dalam sistem 0.5642.

| sample solution | | | |
|-----------------|-----------------------|------------------------|----------------------|
| k | Prob (num in sys = k) | Prob (num in sys <= k) | Prob (num in sys >k) |
| 0 | .1333 | .1333 | .8667 |
| 1 | .1156 | .2489 | .7511 |
| 2 | .1001 | .349 | .651 |
| 3 | .0868 | .4358 | .5642 |
| 4 | .0752 | .5111 | .4889 |
| 5 | .0652 | .5762 | .4238 |
| 6 | .0565 | .6327 | .3673 |
| 7 | .049 | .6817 | .3183 |
| 8 | .0424 | .7242 | .2758 |
| 9 | .0368 | .7609 | .2391 |
| 10 | .0319 | .7928 | .2072 |
| 11 | .0276 | .8204 | .1796 |
| 12 | .0239 | .8444 | .1556 |
| 13 | .0207 | .8651 | .1349 |
| 14 | .018 | .8831 | .1169 |
| 15 | .0156 | .8987 | .1013 |
| 16 | .0135 | .9122 | .0878 |
| 17 | .0117 | .9239 | .0761 |
| 18 | .0101 | .9341 | .0659 |
| 19 | .0088 | .9428 | .0572 |
| 20 | .0076 | .9505 | .0495 |

B. Model $M/D/1$

Data berikut sama dengan contoh sebelumnya tetapi model telah diubah.

| M/D/1 Example Solution | | | | | |
|--------------------------------|-------|---------------------------------------|--------|---------|---------|
| Parameter | Value | Parameter | Value | Minutes | Seconds |
| M/D/1 (constant service times) | | Average server utilization | .8667 | | |
| Arrival rate(λ) | 26 | Average number in the queue(L_q) | 2.8167 | | |
| Service rate(μ) | 30 | Average number in the system(L_s) | 3.6833 | | |
| Number of servers | 1 | Average time in the queue(W_q) | .1083 | 6.5 | 390 |
| | | Average time in the system(W_s) | .1417 | 8.5 | 510 |

Format dari output masih sama. Akan tetapi karena model telah diubah, beberapa hasil perhitungan juga berubah. Jumlah pelanggan dalam antrian 2.8167 orang, bukan 5.63 orang seperti dalam sistem $M/M/1$. (Jumlah pelanggan dan waktu antrian dalam system $M/D/1$ selalu setengah dari sistem $M/M/1$). Probabilitas tidak tersedia karena waktu layanan tidak mengikuti dist peluang eksponensial.

C. Model $M/G/1$

Dalam model ini, waktu layanan boleh mengikuti sembarang distribusi. Input tidak hanya rata-rata tingkat layanan, tetapi juga standar deviasi waktu pelayanan. Layar berikut berisi semua informasi untuk contoh ini. Perhatikan bahwa ada satu baris tambahan untuk input. Output adalah sama. Dalam contoh, rata-rata pelanggan masih 30 orang per jam, seperti sebelumnya, tetapi standar deviasi waktu pelayanan adalah 0,05 jam atau 3 menit.

| M/G/1 Example Solution | | | | | |
|-------------------------------|-------|---------------------------------------|---------|---------|---------|
| Parameter | Value | Parameter | Value | Minutes | Seconds |
| M/G/1 (general service times) | | Average server utilization | .8667 | | |
| Arrival rate(λ) | 26 | Average number in the queue(L_q) | 9.1542 | | |
| Service rate(μ) | 30 | Average number in the system(L_s) | 10.0208 | | |
| Number of servers | 1 | Average time in the queue(W_q) | .3521 | 21.125 | 1,267.5 |
| Standard deviation | .05 | Average time in the system(W_s) | .3854 | 23.125 | 1,387.5 |

Sebuah standar deviasi dari 1/tingkat layanan, menghasilkan distribusi eksponensial. Misalnya, karena tingkat layanan adalah 30, jika waktu layanan standar deviasi adalah $1/30 = 0,03333$, model memiliki distribusi waktu layanan eksponensial.

| Exponential Service Times Expressed as an M/G/1 Queue Solution | | | | | |
|--|-------|---------------------------------------|--------|------------|-----------------|
| Parameter | Value | Parameter | Value | Value * 60 | Value * 60 * 60 |
| M/G/1 (general service times) | | Average server utilization | .8667 | | |
| Arrival rate(λ) | 26 | Average number in the queue(L_q) | 5.6333 | | |
| Service rate(μ) | 30 | Average number in the system(L_s) | 6.5 | | |
| Number of servers | 1 | Average time in the queue(W_q) | .2167 | 13 | 779.9993 |
| Standard deviation | .0333 | Average time in the system(W_s) | .25 | 15 | 899.9993 |

Waktu layanan eksponensial ditampilkan di layar sebelumnya. Perhatikan bahwa jawaban adalah identik dengan yang di contoh 1 kecuali untuk pembulatan (digunakan 0,0333 dari pada $1/30$).

Standar deviasi bernilai 0 akan menghasilkan waktu pelayanan konstan model ($M/D/1$). Hal ini ditampilkan pada hasil berikut. Bandingkan hasilnya dengan Contoh 2, yang ditampilkan model $M/D/1$.

| Waiting Lines Results | | | | | |
|---|-------|----------------------------------|--------|------------|-----------------|
| Constant Service Times Expressed as an M/G/1 Queue Solution | | | | | |
| Parameter | Value | Parameter | Value | Value * 60 | Value * 60 * 60 |
| M/G/1 (general service times) | | Average server utilization | .8667 | | |
| Arrival rate(lambda) | 26 | Average number in the queue(Lq) | 2.8167 | | |
| Service rate(mu) | 30 | Average number in the system(Ls) | 3.6833 | | |
| Number of servers | 1 | Average time in the queue(Wq) | .1083 | 6.5 | 390 |
| Standard deviation | 0 | Average time in the system(Ws) | .1417 | 8.5 | 510 |

D. Model $M/E_k/1$

Distribusi waktu layanan lain adalah distribusi Erlang- k . Layar berikut menunjukkan model $M/E_k/1$ dan solusinya. Satu-satunya perbedaan di input adalah bahwa nilai k harus diberikan (bukan tanpa nilai atau standar deviasi).

| Waiting Lines Results | | | | | |
|---------------------------------|-------|----------------------------------|--------|---------|---------|
| Erlang service times solution | | | | | |
| Parameter | Value | Parameter | Value | Minutes | Seconds |
| M/Ek/1 (Erlang-k service times) | | Average server utilization | .8667 | | |
| Arrival rate(lambda) | 26 | Average number in the queue(Lq) | 3.7556 | | |
| Service rate(mu) | 30 | Average number in the system(Ls) | 4.6222 | | |
| Number of servers | 1 | Average time in the queue(Wq) | .1444 | 8.6667 | 520 |
| k for Erlang-k | 3 | Average time in the system(Ws) | .1778 | 10.6667 | 640 |

Ketika k sama dengan 1, seperti layar berikut, distribusi layanan adalah distribusi eksponensial. Bandingkan hasil dengan contoh pertama.

| Waiting Lines Results | | | | | |
|---|-------|----------------------------------|--------|---------|---------|
| Exponential service times expressed as Erlang distribution solution | | | | | |
| Parameter | Value | Parameter | Value | Minutes | Seconds |
| M/Ek/1 (Erlang-k service times) | | Average server utilization | .8667 | | |
| Arrival rate(lambda) | 26 | Average number in the queue(Lq) | 5.6333 | | |
| Service rate(mu) | 30 | Average number in the system(Ls) | 6.5 | | |
| Number of servers | 1 | Average time in the queue(Wq) | .2167 | 13 | 780 |
| k for Erlang-k | 1 | Average time in the system(Ws) | .25 | 15 | 900 |

E. Antrian $M/M/s$

Pertanyaan yang paling dasar dalam antrian adalah apa yang akan terjadi jika jumlah server meningkat. Pada layar berikut output untuk situasi aslinya kecuali dengan dua server. Waktu tunggu menjadi 0,0077 jam daripada 0,217 jam dalam

deskripsi asli. Untuk memeriksa contoh asli, Anda dapat menggunakan model $M/M/s$ dan masukkan 1 server.

| Multiple servers solution | | | | | |
|---------------------------|-------|---------------------------------------|-------|---------|----------|
| Parameter | Value | Parameter | Value | Minutes | Seconds |
| M/M/s | | Average server utilization | .4333 | | |
| Arrival rate(λ) | 26 | Average number in the queue(L_q) | .2004 | | |
| Service rate(μ) | 30 | Average number in the system(L_s) | 1.067 | | |
| Number of servers | 2 | Average time in the queue(W_q) | .0077 | .4624 | 27.7428 |
| | | Average time in the system(W_s) | .041 | 2.4624 | 147.7428 |

F. System dengan antrian terbatas $M/M/1$

Dalam sistem ini, jumlah ruang tunggu terbatas. Contoh yang khas adalah sistem telepon. Saluran tambahan untuk model ini adalah ukuran sistem maksimum yang diijinkan. Pada contoh berikut paling banyak dua pelanggan dapat masuk ke dalam sistem. Ini berarti bahwa tidak lebih dari satu pelanggan bisa menunggu ketikapelanggan yang lain sedang dilayani. (Ini adalah telepon satu jalur dengan panggilan tunggu.) Jika ada dua server, itu berarti bahwa tidak ada yang bisa menunggu. Hati-hati ketika mempertimbangkan ukuran sistem versus ukuran ruang tunggu.

Catatan: Model ini disebut antrian terbatas dengan ukuran sistem, bukan ukuran antrian yang dimasukkan ke dalam program.

Karena ukuran sistem terbatas, bisa jadi pelanggan akan tiba di sistem tetapi diblokir untuk memasukinya. Oleh karena itu, sebagai contoh, tingkat kedatangan efektif didefinisikan sebagai jumlah aktual pelanggan yang memasuki toko daripada jumlah pelanggan yang tiba di toko. Selain itu, output menampilkan persentase waktu (probabilitas) bahwa sistem penuh.

| Waiting Lines Results | | | | | |
|-----------------------------------|-------|---------------------------------------|---------|---------|----------|
| Finite system size model solution | | | | | |
| Parameter | Value | Parameter | Value | Minutes | Seconds |
| M/M/1 with a Finite System Size | | Average server utilization | .618 | | |
| Arrival rate(λ) | 26 | Average number in the queue(L_q) | .2869 | | |
| Service rate(μ) | 30 | Average number in the system(L_s) | .9049 | | |
| Number of servers | 1 | Average time in the queue(W_q) | .0155 | .9286 | 55.7143 |
| Maximum system size | 2 | Average time in the system(W_s) | .0488 | 2.9286 | 175.7143 |
| | | Effective Arrival Rate | 18.5399 | | |
| | | Probability that system is full | .2869 | | |

Dari contoh ini, hanya 71 persen dari pelanggan yang nampak memasuki sistem, pelanggan datang pada tingkat 26 orang per jam tetapi tingkat kedatangan efektif adalah 18,5399 per jam. Probabilitas ditampilkan, 28.69 persen dari waktu sistem penuh ($k = 2$), artinya, 28,69 persen dari waktu ketika panggilan telepon dibuat menerima sinyal sibuk.

| Table of Probabilities | | | |
|-----------------------------------|-----------------------|----------------------------|-------------------------|
| Finite system size model solution | | | |
| k | Prob (num in sys = k) | Prob (num in sys \leq k) | Prob (num in sys $>$ k) |
| 0 | .382 | .382 | .618 |
| 1 | .3311 | .7131 | .2869 |
| 2 | .2869 | 1 | 0 |

G. System dengan populasi terbatas $M/M/1$

Biasanya diasumsikan bahwa populasi tak terbatas. Tabel di bawah ini menunjukkan sebuah populasi sebanyak 13 pelanggan potensial, masing-masing tiba pada tingkat 2 kali per jam (untuk tingkat kedatangan potensial bersih 26, seperti pada contoh sebelumnya). Ini adalah tingkat kedatangan ketika mereka tidak dalam sistem. Namun, dari output dapat dilihat bahwa rata-rata setiap pelanggan datang 0,088 jam. Tingkat kedatangan efektif adalah 1 kedatangan per jam dikali jumlah rata-rata dari 13 yang tidak dalam sistem. Dalam contoh ini, tingkat kedatangan efektif hanya 22,1 pelanggan per jam (bukan potensi 26 kedatangan per jam). Jika pelayanan lebih baik, maka pelanggan tersebut bisa tiba lebih sering.

| Waiting Lines Results | | | | | |
|----------------------------------|-------|----------------------------------|---------|---------|----------|
| Finite Population Model Solution | | | | | |
| Parameter | Value | Parameter | Value | Minutes | Seconds |
| M/M/1 with a Finite Population | | Average server utilization | .7368 | | |
| Arrival rate PER CUSTOMER | 2 | Average number in the queue(Lq) | 1.2115 | | |
| Service rate(mu) | 30 | Average number in the system(Ls) | 1.9483 | | |
| Number of servers | 1 | Average time in the queue(Wq) | .0548 | 3.2888 | 197.3262 |
| Population size | 13 | Average time in the system(Ws) | .0881 | 5.2888 | 317.3262 |
| | | Effective Arrival Rate | 22.1034 | | |
| | | Probability that customer waits | .6904 | | |

Catatan: Dalam model ini, tingkat kedatangan yang akan dimasukkan ke dalam program adalah tingkat kedatangan untuk nasabah individu. Dalam banyak buku teks, waktu antara kedatangan diberikan. Kali ini harus dikonversi ke tingkat kedatangan. Sebagai contoh, mungkin bahwa masing-masing 5 pelanggan muncul rata-rata setiap 30 menit. Ini harus diubah menjadi tingkat $60/30 = 2$ per jam (per pelanggan). Program ini secara otomatis akan menyesuaikan jumlah pelanggan. Perhatikan bahwa kita masukkan angka 2 sebagai tingkat kedatangan. Hal ini menggoda untuk memasukkan $2 * 5 = 10$, tetapi ini tidak benar!

H. Antrian dengan M/M/s

Berikut adalah contoh dengan biaya. Biaya pelanggan dapat diberikan kepada pelanggan yang menghabiskan waktu dalam sistem atau waktu menunggu. Untuk setiap jam pelanggan menunggu dibayar biaya \$ 2. Ini menghasilkan total biaya sebesar \$ $2 * 6,5$ pelanggan dalam sistem ditambah biaya tenaga kerja \$ 4/jam untuk biaya sistem total \$ 17 per jam (baris paling bawah dalam tabel). Atau, biaya mungkin diberikan untuk waktu menunggu pelanggan daripada waktu yang dihabiskan dalam sistem. Dalam hal ini, ada 5,633 pelanggan menunggu rata-rata dikalikan dengan \$ 2 untuk subtotal dari \$ 11,27 yang \$ 4 biaya server ditambahkan menghasilkan \$ 15,27, seperti yang ditampilkan di baris kedua dari bawah.

| sample solution | | | | | |
|---------------------------|-------|---------------------------------------|---------|---------|----------|
| Parameter | Value | Parameter | Value | Minutes | Seconds |
| M/M/s | | Average server utilization | .8667 | | |
| Arrival rate(λ) | 26 | Average number in the queue(L_q) | 5.6333 | | |
| Service rate(μ) | 30 | Average number in the system(L_s) | 6.5 | | |
| Number of servers | 1 | Average time in the queue(W_q) | .2167 | 13 | 780 |
| Server cost \$/time | 4 | Average time in the system(W_s) | .25 | 15 | 900.0001 |
| Waiting cost \$/time | 2 | Cost (Labor + # waiting*wait cost) | 15.2667 | | |
| | | Cost (Labor + # in system*wait cost) | 17 | | |

| sample solution | | |
|-------------------|-----------------------------|----------------------------|
| Number of servers | Total cost based on waiting | Total cost based on system |
| 1 | 15.2667 | 17 |
| 2 | 8.4007 | 10.1341 |
| 3 | 12.0518 | 13.7851 |
| 4 | 16.007 | 17.7403 |
| 5 | 20.0009 | 21.7342 |

Tugas Latihan

- Manajer sebuah restoran yang cukup sukses akhir-akhir ini merasa prihatin dengan panjangnya antrian. Beberapa pelanggannya telah mengadu tentang waktu menunggu yang berlebihan, oleh karena itu manajer khawatir suatu saat akan kehilangan pelanggannya. Analisis dengan teori antrian diketahui, tingkat kedatangan rata-rata langganan selama periode puncak adalah 50 orang per jam. System pelayanan satu per satu dengan waktu rata-rata 1 orang 1 menit.
Pertanyaan:
 - Tingkat kegunaan bagian pelayanan restoran
 - Jumlah rata-rata dalam antrian
 - Jumlah rata-rata dalam system
 - Waktu menunggu rata-rata dalam antrian
 - Waktu menunggu rata-rata dalam system
- Sama seperti kasus 1, hanya dalam pelayanan di restoran dalam satu kali pelayanan dapat dilayani 2 orang sekaligus.

3. Sebuah SPBU memiliki satu mesin pompa yang dapat melayani rata-rata 25 mobil/jam. Jika rata-rata kedatangan mobil perjamnya adalah 20 kendaraan, maka hitunglah:
- a. Tingkat kegunaan bagian pelayanan pom bensin tersebut!
 - b. Jumlah rata-rata kendaraan dalam antrian!
 - c. Jumlah rata-rata kendaraan dalam system!
 - d. Waktu rata-rata yang dibutuhkan kendaraan dalam antrian!
 - e. Waktu rata-rata yang dibutuhkan kendaraan dalam system!

DAFTAR PUSTAKA

Sumber buku.

Howard J. Weiss. 2010. *POM-QM for Windows version 4*. New Jersey: Pearson Prentice Hall.

Siswanto. 2007. *Operations Research Jilid 1*, Jakarta: Erlangga.

Andi Wijaya. 2011. *Pengantar Riset Operasi*. Jakarta: Mitra Wacana Media.

Dwi Hayu Agustini & Yus Endra Rahmadi. 2004. *Riset Operasional Konsep-Konsep Dasar*. Jakarta: Rineka Cipta.

Johanes Supranto. 2006. *Riset Operasi*. Jakarta: Universitas Indonesia.

Sumber internet.

Z. Hartawan. 2013. *Praktikum Riset Operasional Dengan Software POM-QM for Windows ver. 3*. Bengkulu.

Harsuko Riniwati. 2014. *Buku Panduan Praktikum Operation Research*. Malang.

Nuhfil Hanani & Rosihan Anwar. *Metode Kuantitatif*.