

BAB II

LINEAR GOAL PROGRAMMING

2.1 Dasar Teori

Linear goal programming merupakan pengembangan *linear programming*. *Goal programming* diperkenalkan oleh Charnes dan Cooper. *Linear goal programming* merupakan modifikasi atau variasi khusus dari *linear goal programming* yang memiliki banyak tujuan. *Linear goal programming* bertujuan untuk meminimumkan penyimpangan-penyimpangan dari tujuan-tujuan tertentu dengan mempertimbangkan hierarki prioritas. Model *linear goal programming* banyak diterapkan dalam beberapa situasi pengambilan keputusan, seperti bidang *finance*, produksi, *marketing*, dan akuntansi. Unsur-unsur *linear goal programming* adalah fungsi tujuan, kendala-kendala, kendala *non-negative*, dan kendala *structural* (Widya, 2014).

2.2 Tujuan Praktikum

Adapun tujuan dari praktikum mengenai *linear goal programming* antara lain sebagai berikut:

1. Praktikan mampu untuk menyelesaikan persoalan *linear goal programming* dengan komputerisasi.
2. Praktikan mampu untuk menganalisis hasil (*output*) dari persoalan *linear goal programming*.

2.3 Penyelesaian *Linear Goal Programming*

Berikut adalah cara penyelesaian masalah *linear goal programming* dengan komputerisasi:

Contoh kasus:

The Fabulous merupakan sebuah perusahaan percetakan terbesar di Bandung. Perusahaan ini setiap harinya mencetak 3 kategori produk utama, yaitu: majalah,

pamflet, dan brosur. Untuk mencetak produk majalah memakan waktu selama 4 jam proses desain, 3 jam proses pencetakan, dan 2 jam proses pengemasan. Sedangkan untuk mencetak produk pamflet memakan waktu selama 2 jam proses desain, 1 jam proses pencetakan, dan 1 jam proses pengemasan. Untuk mencetak produk brosur memakan waktu selama 2 jam proses desain, 2 jam proses pencetakan, dan 1 jam proses pengemasan. Untuk total waktu produksi pada setiap minggunya dikerjakan selama 64 jam untuk proses desain, 52 jam untuk proses pencetakan, dan 15 jam untuk proses pengemasan.

The Fabulous menentukan 4 prioritas dalam menunjang kinerja dan keuntungan yang diinginkan yaitu:

Prioritas 1 : Meminimkan ketidaktercapaian total produksi sebesar 100 unit perminggu
(*Goal 1*)

Prioritas 2 : Meminimkan ketidaktercapaian hasil produksi kurang dari 35 unit majalah perminggu (*Goal 2*)

Prioritas 3 : Meminimkan ketidaktercapaian hasil produksi kurang dari 40 unit pamflet perminggu (*Goal 3*)

Prioritas 4 : Meminimkan ketidaktercapaian hasil produksi kurang dari 42 unit brosur perminggu (*Goal 4*)

Dibawah ini merupakan formulasi dari model GP untuk menentukan jumlah dari 3 kategori produk utama dari *The Fabulous*, yaitu majalah, pamflet, dan brosur yang harus diproduksi.

Tabel 2.1 Permasalahan *linear goal programming*

Proses Pencetakan	Majalah	Pamflet	Brosur	Kemampuan (Jam/Minggu)
Desain (Jam)	4	2	2	64
Pencetakan (Jam)	3	1	2	52
Pengemasan (Jam)	2	1	1	15

Penyelesaian:

Penyelesaian pada persoalan masalah *linear goal programming* di atas dapat diselesaikan dengan menggunakan metode grafik yang dilakukan secara komputerisasi

menggunakan *software* LINGO 18.0. Hasil dari perhitungan *The Fabulous* tersebut dengan menggunakan *software* LINGO dapat dijabarkan sebagai berikut:

1. Membuka *file* LINGO 18.0 dengan cara klik dua kali pada *icon* LINGO,
2. Layar akan muncul *untitled* baru yang siap untuk tempat menyetikkan formulasi,
3. Berdasarkan persoalan di atas, maka terlebih dahulu ditentukan variabel keputusan yang kemudian diterjemahkan ke dalam model matematis seperti berikut:

Variabel keputusan:

X1 : jumlah produk majalah yang harus diproduksi (unit)

X2 : jumlah produk pamflet yang harus diproduksi (unit)

X3 : jumlah produk brosur yang harus diproduksi (unit)

Tujuan :

Prioritas 1 : Meminimkan ketidaktercapaian total produksi 100 unit perminggu
(*Goal 1*)

Prioritas 2 : Meminimkan ketidaktercapaian produksi kurang dari 35 unit majalah perminggu (*Goal 2*)

Prioritas 3 : Meminimkan ketidaktercapaian produksi kurang dari 40 unit pamflet perminggu (*Goal 3*)

Prioritas 4 : Meminimkan ketidaktercapaian produksi kurang dari 42 unit brosur perminggu (*Goal 4*)

Formulasi ke dalam model matematis:

Fungsi tujuan: $\text{Min } P_1d_1^- + P_2d_2^- + P_3d_3^- + P_4d_4^-$

Batasan:

$$4X_1 + 2X_2 + 2X_3 \leq 64$$

$$3X_1 + X_2 + 2X_3 \leq 52$$

$$2X_1 + X_2 + X_3 \leq 15$$

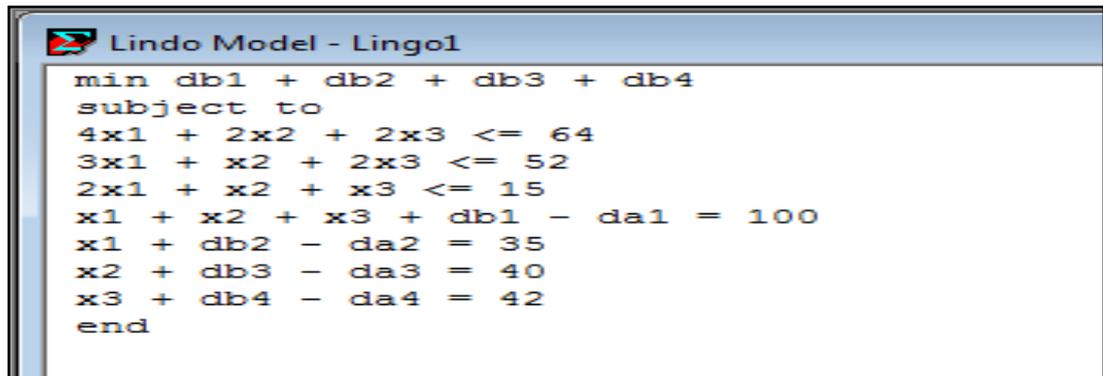
$$X_1 + X_2 + X_3 + d_1^- - d_1^+ = 100$$

$$X_1 + d_2^- - d_2^+ = 35$$

$$X_2 + d_3^- - d_3^+ = 40$$

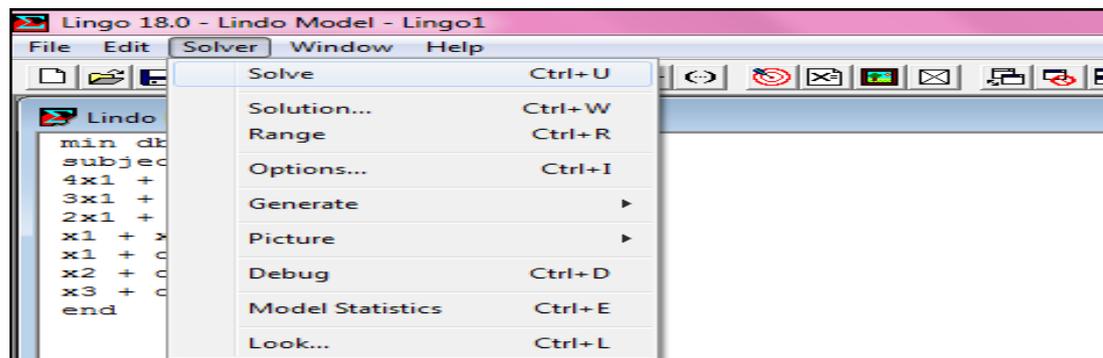
$$X_3 + d_4^- - d_4^+ = 42$$

4. Ketikkan formulasi model matematis yang telah dibuat di atas pada layar LINGO seperti pada gambar berikut.



```
Lindo Model - Lingo1
min db1 + db2 + db3 + db4
subject to
4x1 + 2x2 + 2x3 <= 64
3x1 + x2 + 2x3 <= 52
2x1 + x2 + x3 <= 15
x1 + x2 + x3 + db1 - da1 = 100
x1 + db2 - da2 = 35
x2 + db3 - da3 = 40
x3 + db4 - da4 = 42
end
```

5. Klik *Solver* kemudian pilih *Solve* pada *toolbar* untuk melihat solusi optimal dari perumusan masalah tersebut.



6. Maka akan muncul kotak dialog *Solution Report - Lingo1* dari permasalahan di atas seperti pada gambar berikut.

```

Global optimal solution found.
Objective value:                187.0000
Infeasibilities:                0.000000
Total solver iterations:        8
Elapsed runtime seconds:        0.21

Model Class:                    LE|

Total variables:                11
Nonlinear variables:            0
Integer variables:              0

Total constraints:              8
Nonlinear constraints:          0

Total nonzeros:                27
Nonlinear nonzeros:            0

```

Variable	Value	Reduced Cost
DB1	85.00000	0.000000
DB2	35.00000	0.000000
DB3	40.00000	0.000000
DB4	27.00000	0.000000
X1	0.000000	2.000000
X2	0.000000	0.000000
X3	15.00000	0.000000
DA1	0.000000	1.000000
DA2	0.000000	1.000000
DA3	0.000000	1.000000
DA4	0.000000	1.000000

Row	Slack or Surplus	Dual Price
1	187.0000	-1.000000
2	34.00000	0.000000
3	22.00000	0.000000
4	0.000000	2.000000
5	0.000000	-1.000000
6	0.000000	-1.000000
7	0.000000	-1.000000
8	0.000000	-1.000000

7. Setelah diperoleh *output* di atas, maka selanjutnya dilakukan analisis *output* atau interpretasi hasil.

Note:

Pada *software* LINGO, kekurangan yang dimiliki yaitu tidak dapat menampilkan grafik (hanya jika persoalan terdiri dari 2 variabel).

2.4 Analisis *Output* atau Interpretasi Hasil *Goal Programming*

Analisis interpretasi hasil atau *output* dari *software* LINGO 18.0 yang digunakan untuk penyelesaian optimal permasalahan di atas dijelaskan sebagai berikut:

1. *Objective Function Value*

Pada bagian *Global Optimal Solution Found* terdapat *objective value* dapat dilihat bahwa hasil perhitungannya adalah sebesar 187 dengan fungsi tujuan $\text{Min } P_1 d_1^- +$

$P_2d_2^- + P_3d_3^- + P_4d_4^-$. *Infeasibilities* sebesar 0 dan *total solver iteration* sebanyak 8 iterasi.

2. Value

Variabel keputusan pada *output* LINGO ditandai dengan label variabel. Permasalahan *The Fabulous* tersebut memiliki tiga variabel dengan label X1, X2, dan X3. X1 adalah produksi majalah, X2 adalah produksi pamflet dan X3 adalah produksi brosur. Bahwa jumlah produk majalah yang harus diproduksi (X1) sebanyak 0 unit, jumlah produk pamflet yang harus diproduksi (X2) sebanyak 0 unit, dan jumlah produk brosur yang harus diproduksi (X3) sebanyak 15 unit. DB1 optimalnya mampu mencapai 85 unit produksi untuk *Goal 1*, DB2 sudah optimal karena mampu menghasilkan 35 unit untuk *Goal 2*, DB3 sudah optimal karena mampu menghasilkan 40 unit untuk *Goal 3*, dan DB4 optimalnya mampu mencapai 27 unit untuk *Goal 4*.

3. Reduced cost

Reduced cost memberikan informasi mengenai sampai sejauh mana nilai C_j harus diturunkan agar nilai variabel keputusan menjadi positif. Nilai *reduced cost* untuk produk majalah (X1) bernilai 2 yang berarti terjadi pengurangan sebesar 2 unit. Untuk produk pamflet (X2) dan produk brosur (X3) *reduced cost*nya bernilai 0 yang berarti untuk variabel X2 dan X3 tidak terjadi pengurangan unit.

4. Slack or Surplus

Nilai *slack or surplus* pada *solution report* menunjukkan hasil pada *row 1* bernilai 187, *row 2* bernilai 34, dan *row 3* bernilai 22. Ketiganya sebagai kendala aktif atau variabel *non basis*. Kemudian pada *row* selanjutnya yaitu *row 3, 4, 5, 6, 7, dan 8* bernilai 0 sebagai kendala *non aktif* atau variabel *basis*.

5. Dual Price

Dual price menginformasikan perubahan pada nilai fungsi tujuan bila nilai ruas kanan berubah satu unit. Nilai *dual price* persoalan di atas untuk *row 1,5,6,7, dan 8* bernilai sebesar -1 yang apabila menghasilkan 1 unit akan mengurangi nilai dari fungsi tujuannya sebesar 1, pada *row 2 dan row 3* bernilai 0 yang apabila menghasilkan 1 unit maka tidak akan menambah nilai dari fungsi tujuannya.. Pada *row 4* bernilai sebesar 2 yang apabila menghasilkan 1 unit maka akan menambah nilai dari fungsi tujuannya sebesar 2.

BAB IV

SISTEM ANTRIAN (*QUEUING THEORY*)

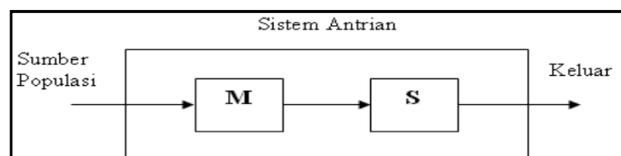
4.1 Dasar Teori

Teori antrian adalah teori yang menyangkut studi matematis dari antrian-antrian atau baris-baris penungguan (Dimiyati, 1992). Pelopor penyusunan teori antrian adalah seorang ahli matematika dari Denmark, Agner Kramp Erlang (1878-1929). Penulisan model antrian pada umumnya mengikuti notasi Kendall-Lee bentuk $a/b/c/d/e$. Suatu antrian adalah garis tunggu dari pelanggan yang memerlukan layanan dari satu atau lebih pelayanan. Studi matematika dari garis tunggu ini disebut teori antrian.

Tujuan dasar model-model antrian adalah untuk meminimumkan total biaya langsung penyediaan fasilitas pelayanan dan biaya tidak langsung karena para individu harus menunggu untuk dilayani.

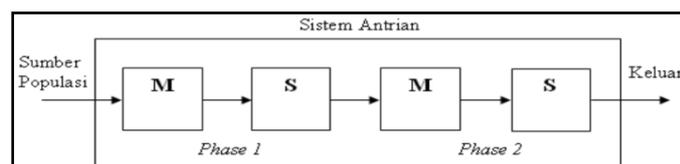
Model struktur antrian secara umum adalah sebagai berikut:

1. *Single channel-single phase*, menunjukkan bahwa dalam struktur hanya ada satu jalur untuk memasuki sistem pelayanan atau ada satu fasilitas pelayanan.



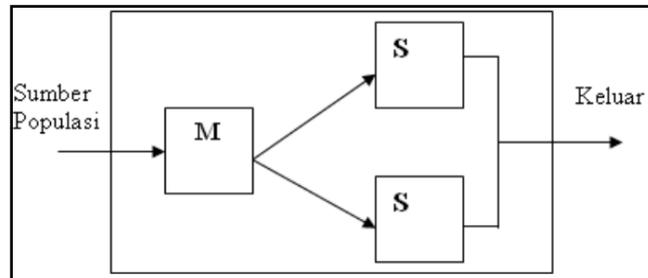
Gambar 4.1 Struktur Antrian *Single Channel-Single Phase*

2. *Single channel-multiphase*, menunjukkan bahwa ada dua atau lebih pelayanan yang dilaksanakan secara berurutan.



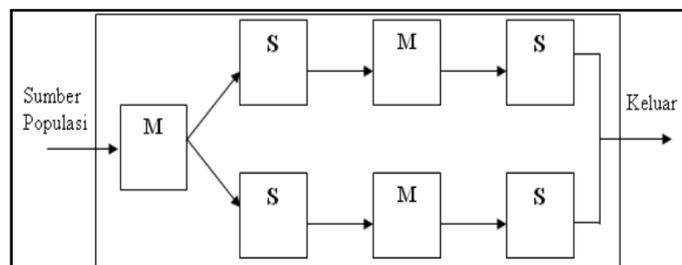
Gambar 4.2 Struktur Antrian *Single Channel-Multiphase*

3. *Multichannel-single phase*, menunjukkan bahwa terdapat dua atau lebih fasilitas pelayanan dialiri oleh antrian tunggal.



Gambar 4.3 Struktur Antrian *Multichannel-single phase*

4. *Multichannel-multiphase*, menunjukkan bahwa terdapat dua atau lebih fasilitas pelayanan dan pelayanan yang dilaksanakan.



Gambar 4.4 Struktur Antrian *Multichannel-multiphase*

4.2 Tujuan Praktikum

Adapun tujuan dari praktikum mengenai sistem antrian antara lain sebagai berikut:

1. Praktikan mampu untuk menyelesaikan persoalan sistem antrian dengan komputerisasi.
2. Praktikan mampu untuk menganalisis hasil (*output*) dari persoalan sistem antrian.

4.3 Penyelesaian Teori Antrian

Berikut adalah cara penyelesaian masalah sistem antrian dengan komputerisasi: Contoh kasus:

Kedai Kopiria yang berlokasi di jalan Ahmad Yani memiliki 1 unit *server* antrian untuk pemesanan kopi dan makanan ringan. Dalam sebuah penelitian di kedai ini, didapatkan waktu rata-rata untuk melayani pelanggan adalah 30 pelanggan/jam. Pelanggan yang datang rata-rata adalah 23 pelanggan/jam. Gambar antrian di kedai Kopiria dapat dilihat pada Gambar 4.1. Tentukan:

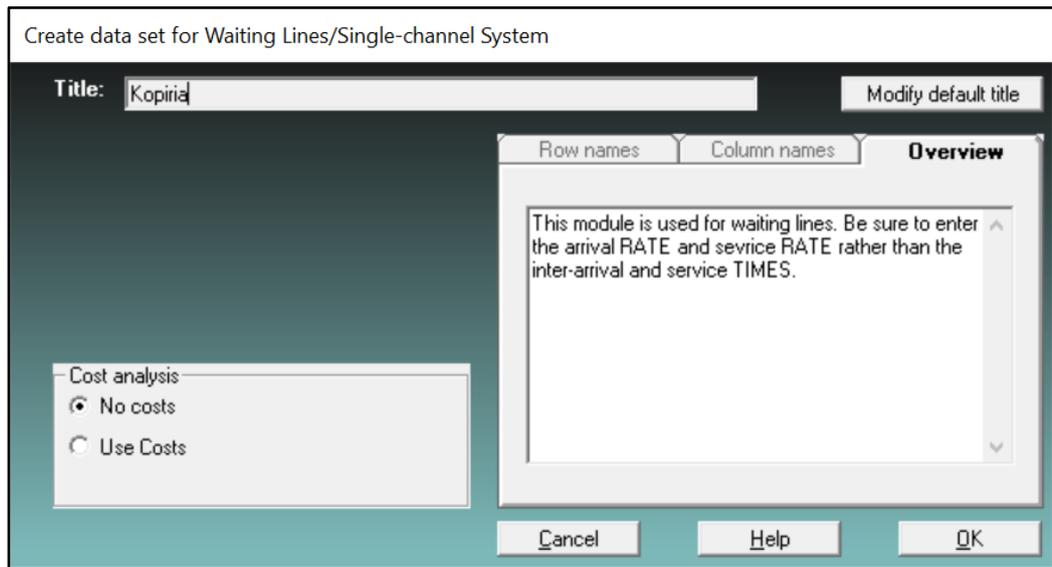
1. Tingkat rata-rata kesibukan sistem,
2. Rata-rata jumlah pelanggan dalam garis tunggu,
3. Jumlah rata-rata pelanggan di dalam sistem,
4. Waktu rata-rata pelanggan di garis tunggu, dan
5. Waktu rata-rata pelanggan berada dalam sistem.

Permasalahan teori antrian di atas akan diselesaikan secara komputerisasi dengan *software* POM-QM *For Windows*. Di bawah ini adalah langkah-langkah perhitungan untuk permasalahan yang ada di atas dengan menggunakan *software* POM-QM *For Windows*.

1. *Single Channel- Single Phase*

Langkah-langkah untuk menyelesaikan persoalan teori antrian dengan *software* POM-QM *For Windows* adalah sebagai berikut:

- a. Klik dua kali pada *icon* POM-QM *For Windows* untuk membuka aplikasi POM-QM *For Windows*.
- b. Klik *Module*, kemudian pilih *waiting line* pada *toolbar*.
- c. Layar kosong akan muncul, kemudian klik *File*. Lalu Klik *New*, kemudian pilih *Single Channel- Single Phase* pada *toolbar*.
- d. Kotak dialog *Create data set for Waiting Lines/Single-channel System* akan muncul. Ketik judul “Kopiria” pada *Title* dan pilih *no cost* pada *Cost Analysis* seperti yang dapat dilihat pada Gambar 4.3.



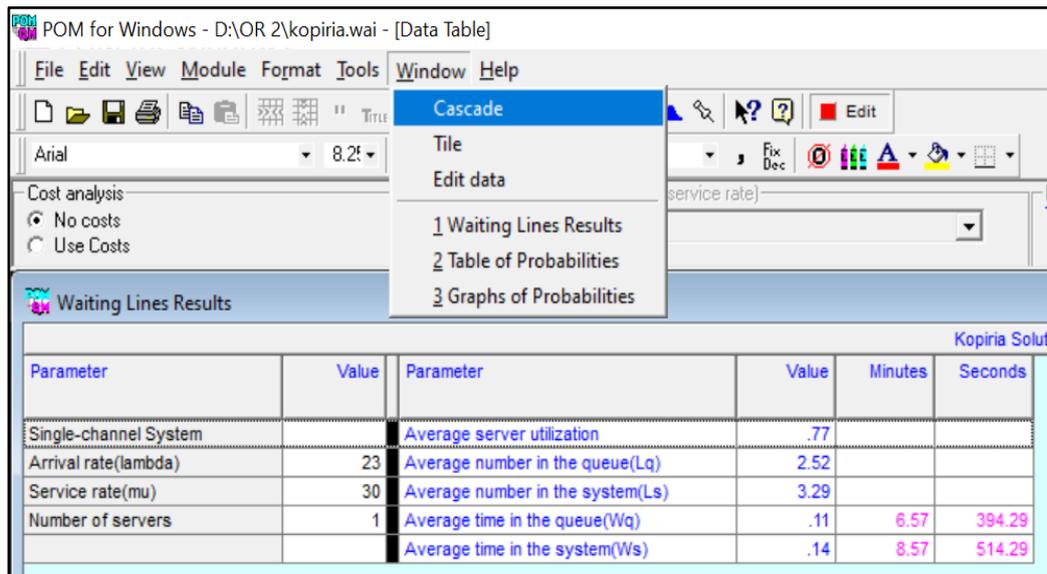
Gambar 4.3 Kotak dialog *Create data set for Waiting Lines*

- e. Klik OK, akan muncul *Data Table* yang akan diisi sesuai dengan data yang ada pada permasalahan di atas yaitu: *arrival rate (lambda)* = 23, *service rate (mu)* = 30, dan *number of service* = 1, seperti yang dapat dilihat pada Gambar 4.4.

Parameter	Value
Single-channel System	
Arrival rate(lambda)	23
Service rate(mu)	30
Number of servers	1

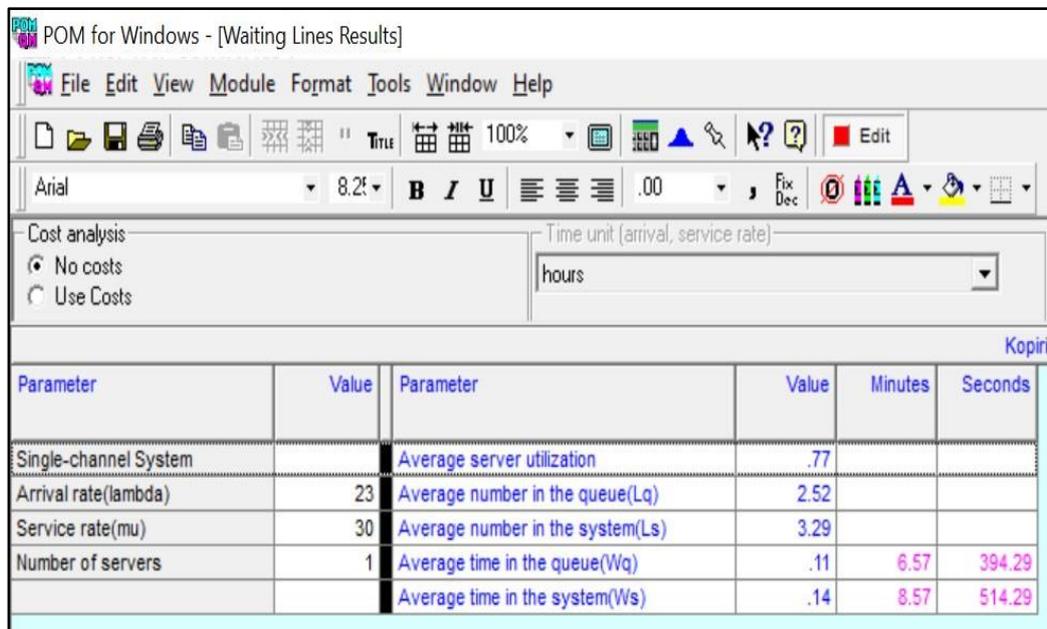
Gambar 4.4 *Data Table*

- f. Klik *Solve* pada *toolbar*, lalu akan muncul kotak dialog *output* pada *toolbar windows* yang akan keluar seperti pada Gambar 4.5.



Gambar 4.5 Kotak dialog *output*

- g. Untuk melihat hasil yang akan diperoleh maka pilih *Waiting Lines Result* sehingga menghasilkan *output* seperti pada Gambar 4.6.



Gambar 4.6 *Waiting lines result*

- h. Untuk melihat hasil tabel probabilitas maka pada *toolbar Window* pilih *Table of Probabilities* sehingga menghasilkan *output* seperti pada Gambar 4.7.

POM for Windows - D:\OR 2\kopiria.wai - [Table of Probabilities]

File Edit View Module Format Tools Window Help

Arial 8.25 B I U 100%

Cost analysis
 No costs
 Use Costs

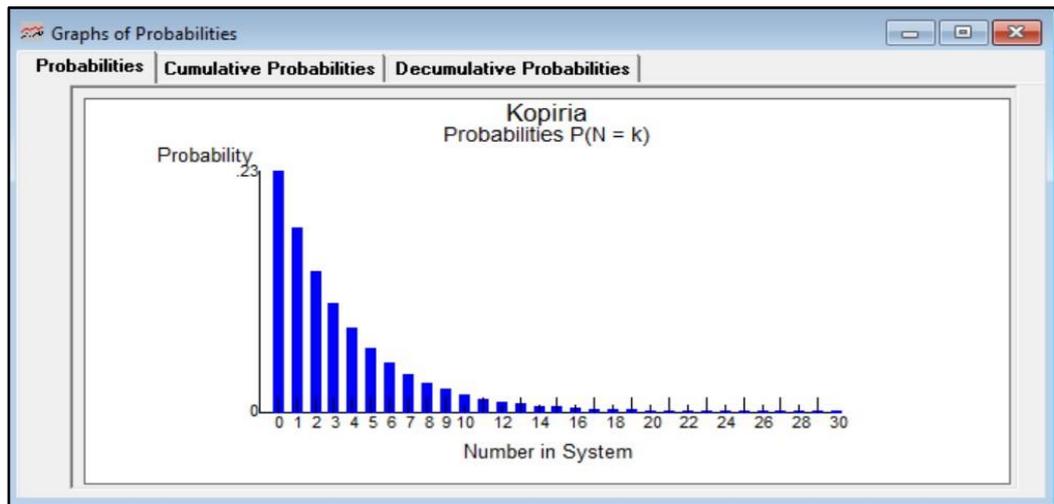
Time unit (arrival, service rate)
 hours

Kopiria Solution

k	Prob (num in sys = k)	Prob (num in sys <= k)	Prob (num in sys >k)
0	.23	.23	.77
1	.18	.41	.59
2	.14	.55	.45
3	.11	.65	.35
4	.08	.74	.26
5	.06	.8	.2
6	.05	.84	.16
7	.04	.88	.12
8	.03	.91	.09
9	.02	.93	.07
10	.02	.95	.05
11	.01	.96	.04
12	0	.97	.03
13	0	.98	.02
14	0	.98	.02
15	0	.99	.01
16	0	.99	.01
17	0	1	0
18	0	1	0
19	0	1	0
20	0	1	0
21	0	1	0
22	0	1	0
23	0	1	0
24	0	1	0
25	0	1	0
26	0	1	0
27	0	1	0
28	0	1	0
29	0	1	0
30	0	1	0

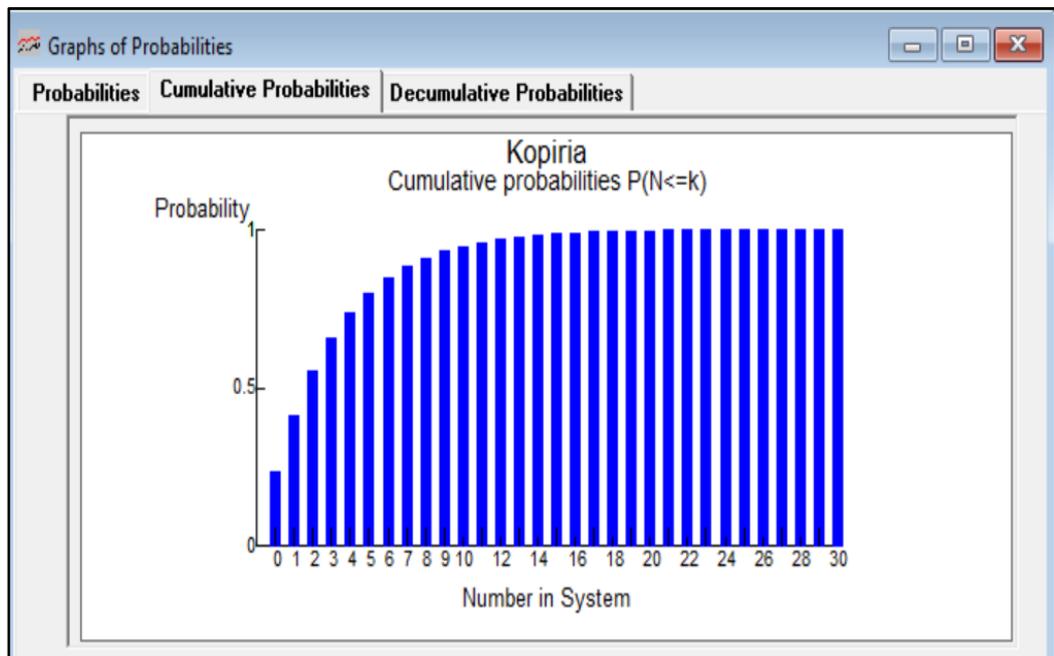
Gambar 4.7 Table of probabilities

- i. Untuk melihat hasil grafik probabilitas maka pada *toolbar Window* pilih *Graphs of Probabilities* sehingga menghasilkan *output* seperti pada Gambar 4.8.



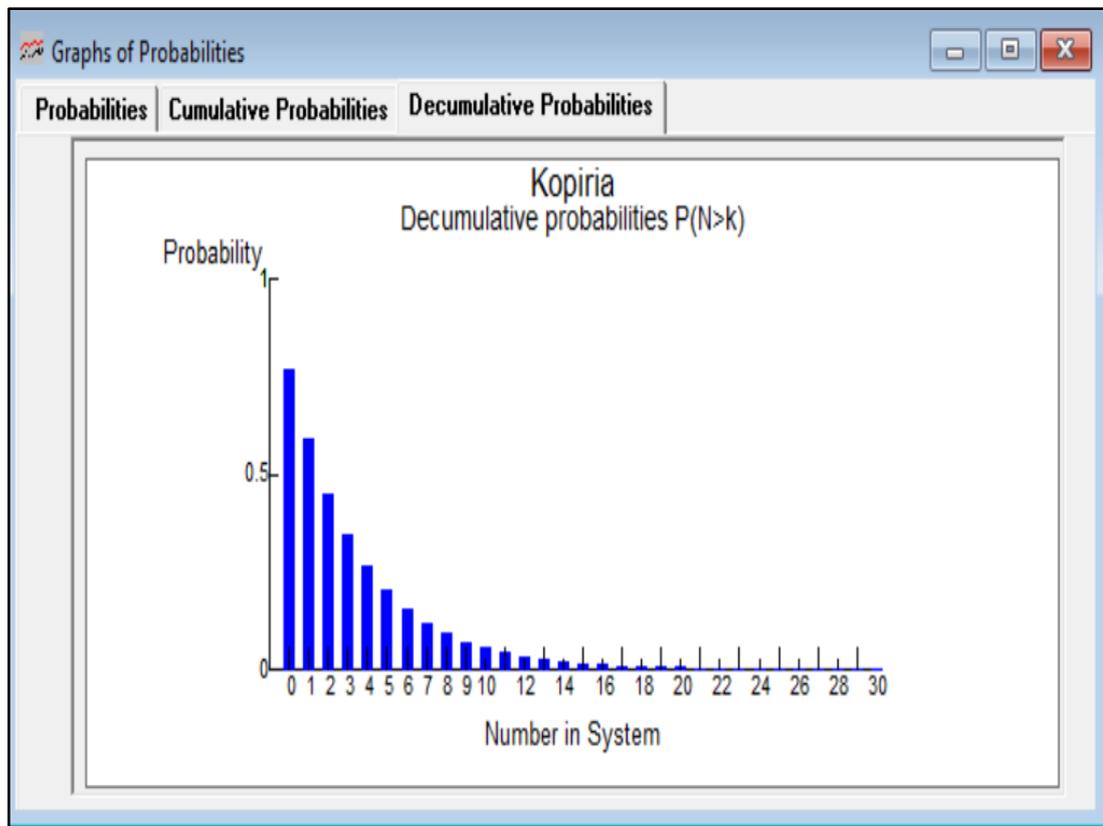
Gambar 4.8 *Graph of probabilities*

- j. Klik *Cumulative Probabilities*, maka akan keluar kotak dialog *Graph of Cumulative Probabilities* pada *toolbar window* yang akan keluar seperti pada Gambar 4.9.



Gambar 4.9 *Graph of Cumulative Probabilities*

- k. Klik *Decumulative Probabilities*, maka akan keluar kotak dialog *Graph of Decumulative Probabilities* pada *toolbar window* yang akan keluar seperti pada Gambar 4.10.



Gambar 4.10 *Graph of Decumulative Probabilities*

1. Setelah *output* didapatkan, maka berikutnya dilakukan analisis *output* atau interpretasi hasil.

4.4 Analisis *Output* atau Interpretasi Hasil Sistem Antrian

Penyelesaian masalah teori antrian menggunakan *software POM-QM For Windows* akan mengeluarkan *output* berupa hasil dari penyelesaian masalah. Berikut merupakan analisis dari *output* teori antrian:

1. Single Channel- Single Phase

Berikut merupakan analisis *output* teori antrian dari permasalahan di kedai Kopiria yang merupakan permasalahan *Single Channel- Single Phase*:

a. Waiting Line Result

- 1) *Average server utilitation*

Output dari hasil perhitungan mengenai permasalahan di atas menggunakan *software POM-QM For Windows* menunjukkan nilai *average server* yaitu 0,77 yang menunjukkan bahwa tingkat kesibukan sistem dalam antrian kedai Kopiria adalah 0,77 atau 77%.

2) *Average number in the queue (Lq)*

Output dari hasil perhitungan mengenai permasalahan di atas menggunakan *software POM-QM For Windows* menunjukkan nilai *average number in the queue (Lq)* yaitu 2,52 yang menunjukkan bahwa rata-rata jumlah pelanggan dalam garis tunggu atau antrian kedai Kopiria adalah 2,52 orang/jam.

3) *Average number in the system (Ls)*

Output dari hasil perhitungan mengenai permasalahan di atas menggunakan *software POM-QM For Windows* menunjukkan nilai *average number in the system (Ls)* yaitu 3,29 yang menunjukkan bahwa jumlah rata-rata pelanggan di dalam sistem adalah 3,29 orang/jam.

4) *Average time in the queue (Wq)*

Output dari hasil perhitungan mengenai permasalahan di atas menggunakan *software POM-QM For Windows* menunjukkan nilai *average time in the queue (Wq)* yaitu 6,57 yang menunjukkan bahwa waktu rata-rata pelanggan di garis tunggu adalah 6,57 menit atau 394,29 detik.

5) *Average time in the system (Ws)*

Output dari hasil perhitungan mengenai permasalahan di atas menggunakan *software POM-QM For Windows* menunjukkan nilai *average time in the system (Ws)* yaitu 8,57 yang menunjukkan bahwa waktu rata-rata pelanggan berada dalam sistem adalah 8,57 menit atau 514,29 detik.

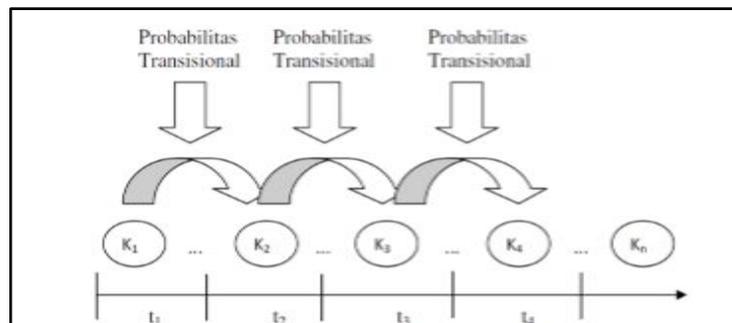
BAB III

MARKOV CHAIN

3.1 Dasar Teori

Chain Rantai Markov (Markov Chains) adalah suatu teknik matematika yang biasa digunakan untuk melakukan pemodelan (*modelling*) bermacam-macam sistem dan proses bisnis. Teknik ini dapat digunakan untuk memperkirakan perubahan-perubahan di waktu yang akan datang dalam variabel-variabel dinamis atas dasar perubahan-perubahan dari variabel-variabel dinamis tersebut di waktu yang lalu. Teknik ini dapat digunakan juga untuk menganalisis kejadian-kejadian pada waktu-waktu mendatang secara matematis (Andry, 2015).

Gambaran mengenai rantai Markov ini kemudian dituangkan dalam Gambar 2.1 Peristiwa Dalam Rantai Markov, dimana gerakan-gerakan dari beberapa variabel di masa yang akan datang bisa diprediksi berdasarkan gerakan-gerakan variabel tersebut pada masa lalu. K_4 dipengaruhi oleh kejadian K_3 , K_3 dipengaruhi oleh kejadian K_2 dan demikian seterusnya dimana perubahan ini terjadi karena peranan probabilitas transisi (*transition probability*) (Andry, 2015).



Gambar 3.1 Peristiwa Dalam Rantai Markov

Rantai markov ini dikenalkan oleh Andrei A. Markov, ahli matematika dari Rusia yang lahir tahun 1856 (Ching, W., dan Ng, M. 2006). Analisis markov menghasilkan suatu informasi probabilistik yang dapat digunakan untuk membantu pembuatan keputusan,

jadi analisis ini bukan suatu teknik optimisasi melainkan suatu teknik deskriptif. Analisis markov merupakan suatu bentuk khusus dari model probabilistik yang lebih umum dinamakan dengan *Stochastic process*, yaitu proses perubahan perubahan probabilistik yang terjadi terus –menerus (Putra, 2018).

Konsep dasar analisis markov adalah *state* dari sistem atau *state* transisi, sifat dari proses ini adalah apabila diketahui proses berada dalam suatu keadaan tertentu, maka peluang berkembangnya proses di masa mendatang hanya tergantung pada keadaan saat ini dan tidak tergantung pada keadaan sebelumnya, atau dengan kata lain rantai Markov adalah rangkaian proses kejadian dimana peluang bersyarat kejadian yang akan datang tergantung pada kejadian sekarang (Putra, 2018).

Menurut Basjaraduin (2016), beberapa syarat yang harus dipenuhi pada analisis Markov adalah sebagai berikut:

1. Jumlah probabilitas transisi keadaan awal dari sistem sama dengan 1,
2. Probabilitas tersebut berlaku untuk semua partisipan dalam sistem,
3. Probabilitas transisi konstan sepanjang waktu, dan
4. Kondisi merupakan kondisi yang independen sepanjang waktu.

3.2 Tujuan Praktikum

Adapun tujuan dari praktikum mengenai markov chain antara lain sebagai berikut:

1. Praktikan mampu untuk menyelesaikan persoalan *markov chain* dengan komputerisasi, dan
2. Praktikan mampu untuk menganalisis hasil (*output*) dari persoalan *markov chain* dengan komputerisasi.

3.3 Penyelesaian Rantai Markov

Penyelesaian pada persoalan masalah *markov chain* dapat diselesaikan dengan secara komputerisasi menggunakan *software* POM-QM For Windows dan WinQSB.

Terdapat 3 tempat Warung makan yang terdapat di Jalan Pramuka antara lain, Restu, Jogsol, dan kartini. Jumlah seluruh pelanggan dari ke 3 tempat warung makan tersebut totalnya adalah 400. Berikut adalah hasil dari penelitian ke 3 warung makan tersebut.

Tabel 3.1 Jumlah Pelanggan Warung Makan

Fotocopy	Banyaknya pelanggan	
	Bulan pertama	Bulan Kedua
Restu	50	100
Jogsol	150	120
Kartini	200	180
Jumlah	400	400

Dari data diatas dapat dilihat adanya perubahan dari bulan pertama ke bulan kedua. Pada Restu bertambah 50, Jogsol berkurang 30 dan Kartini berkurang 20. Berikut merupakan data detail perubahan tersebut.

Tabel 3.2 Detail Perubahan Jumlah Pelanggan

Bulan Pertama	Bulan kedua			Jumlah
	Restu	Jogsol	Kartini	
Restu	25	15	10	50
Jogsol	30	75	45	150
Kartini	50	30	120	200
Jumlah	100	120	180	400

1. Buatlah tabel probabilitas transisi dan matriks transisi!
2. Hitunglah presentase untuk ketiga warung tersebut dalam 3 periode mendatang?
3. Jika harga rata-rata :

Restu = Rp 12000

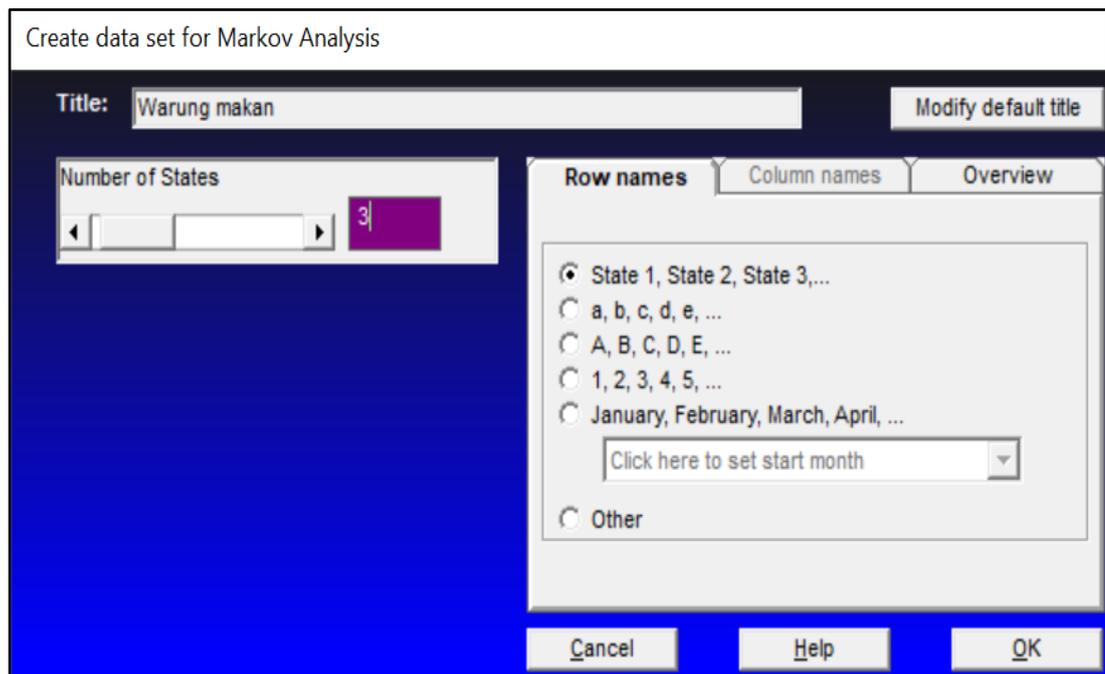
Jogsol = Rp 15000

Kartini = Rp 10000

1. *Software* POM-QM

Penyelesaian pada persoala masalah *markov* pada fotocopy di atas dapat diselesaikan secara komputerisasi menggunakan *software* POM-QM *For Windows*. Berikut dijabarkan prosedur untuk menyelesaikan permasalahan sistem antrian dengan menggunakan *software* POM-QM *For Windows*.

- a. Pertama, buka aplikasi POM-QM *For Windows* dengan cara klik dua kali pada *icon* POM-QM *For Windows*.
- b. Selanjutnya, pada layar akan memunculkan layar baru, kemudian klik *Module* kemudian *Markov Analysis* pada *toolbar*.
- c. Setelah itu akan muncul layar kosong, kemudian klik *File* selanjutnya klik *New* lalu *Multichannel System* pada *toolbar*.
- d. Kemudian muncul kotak dialog *Create data set for Markov Analysis*, kemudian pada kotak dialog *title* masukkan seperti pada Gambar 3.1.
Masukan pada *title* judul dengan nama Warung makan.
Isi *Number of States* sebanyak 3.
Pilih salah satu pilihan pada *Row names*, pilih *state 1, state 2, state 3,...*
Detail tampilan pengisian kotak dialog *Create data set for Markov Analysis* adalah seperti pada Gambar 3.1



Gambar 3.1 Kotak Dialog

- e. Pada kotak dialog diklik *OK*, tampilan *Data Table* diisi sesuai dengan *number of transition* dengan 2 untuk mengetahui data 2 periode setelahnya, dan model matematis yaitu dimasukkan nilai probabilitas dari keempat restoran. Berikut adalah model matematis yang dimasukkan seperti pada Gambar 3.2,

Number of transitions		Instruction		
2		Use the scroll bar or the text box to enter the number of transitions.		
Warung makan				
	Initial	Restu	Jogsol	Kartini
Restu	50	.5	.3	.2
Jogsol	150	.2	.5	.3
Kartini	200	.25	.15	0.6

Gambar 3.2 Formulasi Model Matematis *POM QM*

- f. Klik *Solve* pada *toolbar*, maka akan muncul kotak dialog *output* pada *toolbar window* yang akan keluar hasilnya seperti Gambar 3.3.

Warung makan Steady state transition matrix			
	Restu	Jogsol	Kartini
Restu	.3132	.3031	.3839
Jogsol	.3132	.3031	.3839
Kartini	.3132	.3031	.3839
Ending number (given initial)	125.2631	121.2224	153.5483
Steady State probability	.3132	.3031	.3839

Gambar 3.3 Output Software *POM QM*

1. Diklik *Window* kemudian diklik *Multiplications* untuk memperoleh hasil seperti pada Gambar 3.4,

Warung makan solution			
	Restu	Jogsol	Kartini
End of Period 1			
Restu	.5	.3	.2
Jogsol	.2	.5	.3
Kartini	.25	.15	.6
End Number (given initial)	105	120	175
End of Period 2			
Restu	.36	.33	.31
Jogsol	.275	.355	.37
Kartini	.305	.24	.455
End Number (given initial)	120.25	117.75	162.0

Gambar 3.4 *Multiplications*

- g. Selanjutnya untuk lebih memperjelas hasil *output* bisa ditambahkan dengan pilih *State Analysis*. *Outputnya State* tersebut akan muncul seperti pada Gambar 3.5.

Warung makan solution		
State	Type	Class number
Restu	Recurrent	1
Jogsol	Recurrent	1
Kartini	Recurrent	1

Gambar 3.5 *State Analysis*

- h. Setelah diperoleh hasil seperti *output* dari hasil formulasi dan grafik seperti pada penjelasan diatas, maka selanjutnya di lakukan analisis atau interprestasi hasil.

2. Software WinQSB

Berikut dijabarkan prosedur untuk menyelesaikan permasalahan sistem antrian dengan menggunakan *software WinQSB*.

- a. Klik start lalu membuka *WinQSB* dan pilih *Markov Process*
- b. Klik *File* lalu pilih *new problem*
- c. Input sesuai persoalan yang akan diselesaikan:

Problem Title : Warung Makan

Number of States : 3

Tampilannya bisa dilihat pada Gambar 3.5.



Gambar 3.5 *MKP Problem Specification*

- d. Klik *Edit* lalu pilih *state name* untuk mengganti nama variable, kemudian ubah variable sesuai dengan yang ditentukan :

State 1 : Restu

State 2 : Jogsol

State 3 : Kartini



Gambar 3.6 *State Names*

e. Masukkan data sesuai persoalan :

From \ To	Restu	Jogsol	Kartini
Restu	0.5	0.3	0.2
Jogsol	0.2	0.5	0.3
Kartini	0.25	0.15	0.6
Initial Prob.	50	150	200
State Cost	12000	15000	10000

Gambar 3.7 *Input Data*

f. Klik *solve and analyze* lalu pilih *Solve steady state*

09-20-2021	State Name	State Probability	Recurrence Time
1	Restu	0.3131	3.1935
2	Jogsol	0.3030	3.3000
3	Kartini	0.3838	2.6053
	Expected	Cost/Return =	12141.4100

Gambar 3.8 *Steady State*

g. Untuk mengetahui probabilitas pada periode tertentu, klik *solve and analyze* lalu pilih *Markov Process Step*. Untuk mengetahui periode 1 bulan yang akan datang, ketik angka 1 pada *the number of time periods from initial* lalu klik OK.

State	Initial State Probability	Resulted State Probability
Restu	0.125000	0.262500
Jogsol	0.375000	0.300000
Kartini	0.500000	0.437500

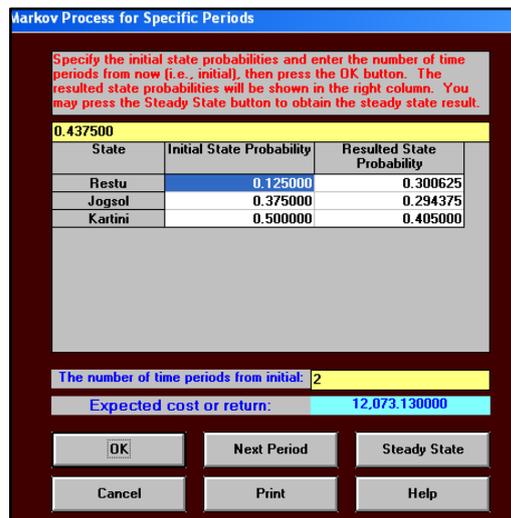
The number of time periods from initial: 1

Expected cost or return: 12025

Buttons: OK, Next Period, Steady State, Cancel, Print, Help

Gambar 3.10 *Markov Process Step Periode 1*

- h. Untuk mengetahui periode 2 bulan yang akan datang, ketik angka 2 pada *the number of time periods from initial* lalu klik OK. Maka akan muncul *Resulted State Probability* dan *Expected Cost or Return*.



Gambar 3.11 *Markov Process Step Periode 2*

3.4 Analisis Output *Markov Chain*

1. *Software POM-QM*

Berdasarkan langkah-langkah komputerisasi menggunakan *software POM-QM ForWindows*, dapat dilakukan analisa *output* penyelesaian permasalahan rantai markov (*markov chain*) sesuai studi kasus di atas. Analisis interpretasi hasil dari permasalahan warung makan berdasarkan *output* pada *POM-QM for windows* adalah sebagai berikut:

a. *Markov analysis results*

Menunjukkan angka yang berbeda pada kolom, hal ini menunjukkan bahwa perhitungan *markov chain* pada studi kasus ini merupakan jenis bukan *steady state* atau tidak seimbang dan terjadi tidak ada keseimbangan pada *market share* dengan nilai untuk ketiga warung makan yaitu 0,3132 pada warung restu, 0,3031 pada warung jogsol, dan 0,3839 pada warung kartini. Terdapat *ending number* yang berarti bahwa jumlah pelanggan pada dua bulan berikutnya yaitu restu sebanyak $125,2631 \approx 125$ pelanggan, jogsol $121,2224 \approx 121$ pelanggan dan kartini $153,5483 \approx 153$ pelanggan.

b. *Multiplications*

Menunjukkan probabilitas pengunjung pada setiap periodenya dan menunjukkan jumlah pengunjung pada setiap *state* akhir periode. Pada bulan selanjutnya yaitu bulan ketiga, warung makan restu sebanyak 105 pelanggan, warung makan jogsol sebanyak 120 pelanggan, dan warung makan kartini sebanyak 175 pelanggan. Pada bulan selanjutnya yaitu bulan keempat, warung makan restu sebanyak $120,25 \approx 120$ pelanggan, warung makan jogsol sebanyak $117,75 \approx 117$ pelanggan, dan warung makan kartini sebanyak 162 pelanggan.

c. *Output state analysis*

Terdapat kolom *type* dengan isi *recurrent* dengan *class of number* untuk setiap *statenya* yaitu 1, hal ini menunjukkan bahwa tipe perhitungan dari permasalahan tersebut adalah berulang. Maksud dari berulang ini adalah bahwa perhitungan dilakukan secara berulang sesuai dengan nilai *steady state* yaitu 0,3132, 0,301, dan 0,3839.

2. *Software WinQSB*

Penyelesaian masalah *markov chain* menggunakan *software* akan mengeluarkan *output* berupa hasil dari penyelesaian masalah. Analisis interpretasi hasil dengan menggunakan Win-QSB adalah sebagai berikut:

a. *State analyze*

Untuk Restu 0,3131 atau 31,31% dengan kondisi pelanggan akan berpaling atau berpindah ke merk lain setelah $3,1395 \approx 3$ bulan Untuk Jogsol 0,3030 atau 30,30% dengan kondisi pelanggan akan berpaling atau berpindah ke merk lain setelah $3,3000 \approx 3$ bulan. Untuk Kartini 0,3838 atau 38,38% dengan kondisi pelanggan akan berpaling atau berpindah ke merk lain setelah $2,6053 \approx 3$ bulan.

b. *Markov process step*

Di dalam *Markov process step* menjelaskan probabilitas pendapatan yang diperoleh oleh objek selama periode mendatang dan juga menjelaskan *expected cost or Return*.

1) Periode 1

Probabilitas pendapatan untuk ke 3 warung pada satu bulan mendatang yaitu untuk restu 0,262500 atau 26,25%, jogsol 0,3 atau 30%, dan kartini 0,437500

atau 43,75%. Dengan *expected cost or return* periode ketiga yaitu Rp12.025,000, dan

2) Periode 2

Probabilitas pendapatan untuk ke 3 warung pada dua bulan mendatang yaitu untuk restu 0,300625 atau 30,06%, jogsol 0,294375 atau 29,43%, dan kartini 0,405000 atau 40,50%. Dengan *expected cost or return* periode kedua yaitu Rp12.073,130.

BAB V

TEORI PERMAINAN (*GAME THEORY*)

5.1 Dasar Teori

Teori permainan (*game theory*) merupakan ilmu pengetahuan yang berhubungan dengan pembuatan keputusan ketika ada dua pihak atau lebih berada dalam kondisi persaingan atau konflik. Pihak-pihak yang bersaing ini disumsikan bersifat rasional dan cerdas, artinya masing-masing pihak akan melakukan strategi tindakan yang rasional untuk memenangkan persaingan itu, dan masing-masing pihak juga mengetahui strategi pihak lawannya, selanjutnya pihak ini disebut pemain. Menurut J Von Neumann dan O. Morgenstern (1944) dalam hal teori permainan, tidak dibiarkan pemain lain diam, tetapi melakukan aksi-reaksi atas apa yang terjadi dalam interaksi. Pemahaman tentang teori permainan:

1. *One-shot*, contohnya jika dua orang bertemu dan tahu bahwa mereka hanya akan sekali saja bertemu, terjadi kasus dimana mereka tidak menjaga reputasi mereka. Mereka berbuat buruk karena tahu bahwa tindakan mereka tidak akan menimbulkan tindakan lain di kemudian hari.
2. *Repeated, finite*; terjadi misalkan ketika orang tahu bahwa 10 hari lagi dia akan mati. Maka segala ekspektasi tersebut mengubah *behavior*-nya untuk berpikir mengenai hari akhir ke-10. Ini membuat orang menjadi *desperate* dan bertindak di luar dugaan atau disebut juga efek akhir.
3. *Repeated, infinite*: terjadi dimana seseorang menjaga reputasinya karena ia yakin akan berinteraksi dengan pihak lain *in the near future and more*.

5.1.1 Ketentuan Umum

Dapat dijelaskan beberapa ketentuan dasar yang terpenting dalam teori permainan yaitu:

1. Setiap pemain bermain rasional, dengan asumsi memiliki intelegensi yang sama, dan tujuan sama, yaitu memaksimalkan *payoff*, dengan kriteria *maximin* dan *minimax*.

2. Terdiri dari 2 pemain, keuntungan bagi salah satu pemain merupakan kerugian bagi pemain lain.
3. Tabel yang disusun menunjukkan keuntungan pemain baris, dan kerugian pemain kolom.
4. Permainan dikatakan adil jika hasil akhir menghasilkan nilai nol (0), tidak ada yang menang/kalah.
5. Tujuan dari teori permainan ini adalah mengidentifikasi strategi yang paling optimal

5.1.2 Strategi

Berikut beberapa strategi yang digunakan dalam *game theory* adalah sebagai berikut:

1. Strategi Murni

Penyelesaian dilakukan dengan menggunakan konsep *maximin* untuk pemain baris dan *minimax* untuk pemain kolom. *Maximin* adalah memaksimalkan keuntungan minimum sedangkan *Minimax* meminimalkan kerugian maksimum, dalam strategi ini pemain akan menggunakan satu strategi tunggal untuk mendapat hasil optimal *saddle point* yang sama

2. Strategi Campuran

Strategi ini dilakukan bila strategi murni belum memberi penyelesaian optimal. Sehingga perlu dilakukan tindak lanjut untuk mendapat titik optimal, dengan usaha mendapatkan *saddle point* yang sama.

5.2 Tujuan Praktikum

Adapun tujuan dari praktikum mengenai sistem antrian antara lain sebagai berikut:

1. Praktikan mampu untuk menyelesaikan persoalan teori permainan dengan komputerisasi.
2. Praktikan mampu untuk menganalisis hasil (*output*) dari persoalan teori permainan

5.3 Penyelesaian Teori Permainan

Berikut adalah cara penyelesaian masalah teori permainan dengan komputerisasi:

Contoh kasus

1. Dua minimarket yaitu Ceriwis dan Danjer bersaing untuk mendapatkan hasil yang

optimal yaitu dengan mendapatkan keuntungan yang maksimum dan meminimalkan kerugian. Produk yang dijual pada kedua minimarket tersebut relatif sama. Kedua minimarket ini ingin menentukan strategi yang dapat menghasilkan keuntungan yang optimal. Hal yang dilakukan Ceriwis dalam menarik minat konsumen untuk membeli produk yaitu dengan menggunakan 3 strategi dan Danjer juga menggunakan 3 strategi dalam menarik konsumen. Berikut strategi dari masing-masing minimarket dapat dilihat sebagai berikut:

Ceriwis \ Danjer	Harga Bundle (S1)	Harga Diskon (S2)	Harga Normal (S3)
Harga Bundle (S1)	4	6	5
Harga Diskon (S2)	5	7	8
Harga Normal (S3)	3	8	3

2. Karena terjadi perkembangan di pasar dan minat konsumen yang terus bertambah, maka minimarket Danjer yang pada sebelumnya hanya melakukan strategi dilihat dari harga yang diberikan kembali melakukan strategi ulang dengan menambahkan satu variable strategi yakni Harga Murah yang disesuaikan dengan harga barang dari *Supplier*. Strategi ini dilihat dari persentase keuntungan yang telah diakumulasi dari berbagai faktor yang ada pada strategi sebelumnya yakni harga murah, harga diskon dan harga normal. Berikut ini merupakan tabel dari strategi dari masing-masing minimarket.

Ceriwis \ Danjer	Harga Bundle (S1)	Harga Diskon (S2)	Harga Normal (S3)	Harga Murah (S4)
Harga Bundle (S1)	2	5	9	4
Harga Diskon (S2)	1	2	2	7
Harga Normal (S3)	7	1	4	2

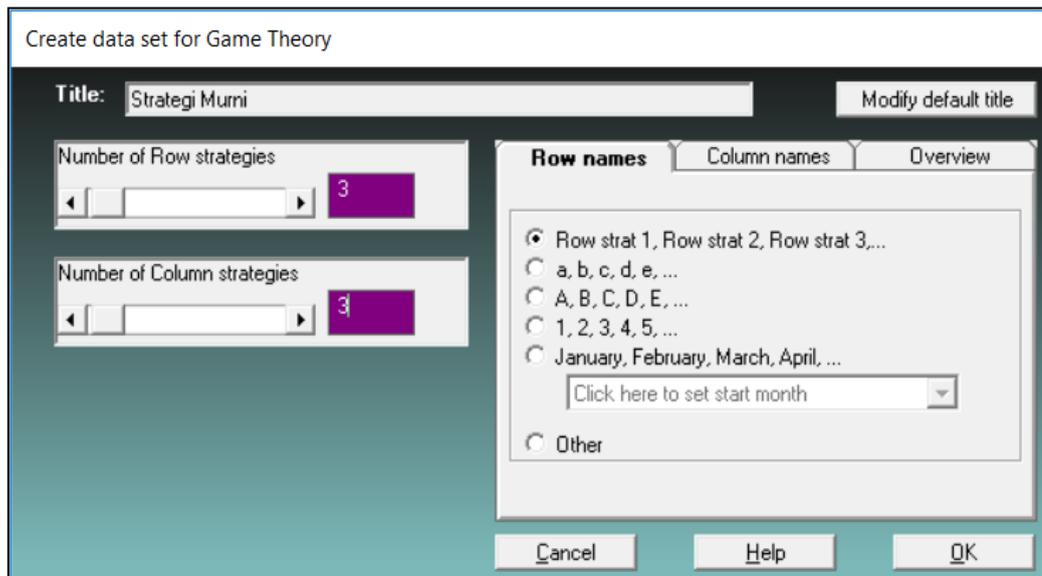
5.3.1 Software POM-QM

1. Strategi Murni

Langkah untuk mencari *saddle point* atau titik pelana untuk kedua minimarket dapat pada studi kasus no. 1 dapat dilihat sebagai berikut:

- a. Membuka *file File POM-QM For Windows* dengan cara klik *Start > All Program > Folder POM-QM For Windows 3 > POM-QM For Windows 3*,
- b. Pada layar akan muncul layar baru, kemudian klik *Module > game theory* pada *Toolbar*,
- c. Setelah itu akan muncul layar kosong, kemudian klik *File > New*,

- d. Kemudian muncul kotak dialog *creat data set for game theory*, kemudian pada kotak dialog title masukkan Pada *Title* masukkan judul misalkan “Strategi Murni”,
- e. Selanjutnya ketik angka 3 pada kolom *number of row strategies* dan kolom *number of column strategies*, lalu klik OK, seperti yang terlihat pada gambar berikut:



- f. Data-data strategi pada kolom *row strat 1, 2, dan 3* dan kolom *row strat 1, 2, dan 3* diisi. Berikut ini kolom data pada setiap strategi:

Strategi Murni			
	Harga Bundle (S1)	Harga Diskon (S2)	Harga Normal (S3)
Harga Bundle (S1)	4	6	5
Harga Diskon (S2)	5	7	8
Harga Normal (S3)	3	8	3

- g. Kemudian klik *solve*, dan akan diperoleh *output* seperti di bawah ini:

	Harga Bundle (S1)	Harga Diskon (S2)	Harga Normal (S3)	Row Mix
Harga Bundle (S1)	4	6	5	0
Harga Diskon (S2)	5	7	8	1
Harga Normal (S3)	3	8	3	0
Column Mix-->	1	0	0	
Value of game (to row)	5			

	Harga Bundle (S1)	Harga Diskon (S2)	Harga Normal (S3)	Row Minimum	Maximin
Harga Bundle (S1)	4	6	5	4	
Harga Diskon (S2)	5	7	8	5	5
Harga Normal (S3)	3	8	3	3	
Column Maximum	5	8	8		
Minimax	5				
Value=5					

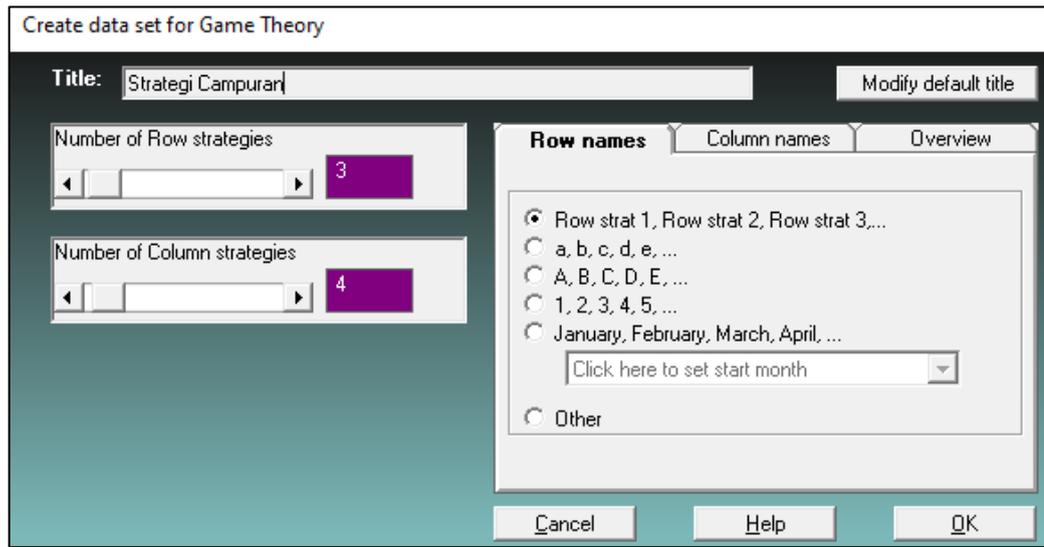
h. Langkah terakhir melakukan analisis terhadap nilai *saddle point* yang didapatkan berdasarkan *output* pada POM-QM For Windows.

2. Strategi Campuran

Langkah untuk mencari nilai *payoff* untuk kedua minimarket dapat pada studi kasus no. 2 dapat dilihat sebagai berikut:

- Membuka file POM-QM For Windows dengan cara klik *Start > All Program > Folder POM-QM For Windows 3 > POM-QM For Windows 3*.
- Pada layar akan muncul layar baru, kemudian klik *Module > game theory* pada *toolbar*.
- Setelah itu akan muncul layar kosong, kemudian klik *File > New*

- d. Kemudian muncul kotak dialog *creat data set for game theory*, kemudian pada kotak dialog title masukkan Pada *Title* masukkan judul misalkan “Strategi Campuran”
- e. Selanjutnya ketik angka 3 pada kolom *number of row strategies* dan angka 4 pada *number of column strategies*, lalu klik OK, seperti yang terlihat pada gambar berikut:



- f. Langkah berikutnya adalah memasukkan data-data strategi pada kolom *row strat* 1, 2, dan 3 dan kolom *col strat* 1, 2, 3 dan 4. Berikut ini kolom data pada setiap strategi:

	Harga Bundle (S1)	Harga Diskon (S2)	Harga Normal (S3)	Harga Murah(S4)
Harga Bundle (S1)	2	5	9	4
Harga Diskon (S2)	1	2	2	7
Harga Normal (S3)	7	1	4	2

- g. Kemudian klik *solve*, dan akan diperoleh *output* seperti di bawah ini:

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

Strategi Campuran Solution					
	Harga Bundle (S1)	Harga Diskon (S2)	Harga Normal (S3)	Harga Murah(S4)	Row Mix
Harga Bundle (S1)	2	5	9	4	,62
Harga Diskon (S2)	1	2	2	7	,06
Harga Normal (S3)	7	1	4	2	,32
Column Mix-->	,36	,26	0	,38	
Value of game (to row)	3,54				

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

Strategi Campuran Solution						
	Harga Bundle (S1)	Harga Diskon (S2)	Harga Normal (S3)	Harga Murah(S4)	Row Minimum	Maximin
Harga Bundle (S1)	2	5	9	4	2	2
Harga Diskon (S2)	1	2	2	7	1	
Harga Normal (S3)	7	1	4	2	1	
Column Maximum	7	5	9	7		
Minimax		5				
$2 \leq \text{value} \leq 5$						

h. Langkah terakhir melakukan analisis terhadap nilai *payoff* yang didapatkan berdasarkan *output* pada POM-QM.

5.3.2 Software WinQSB

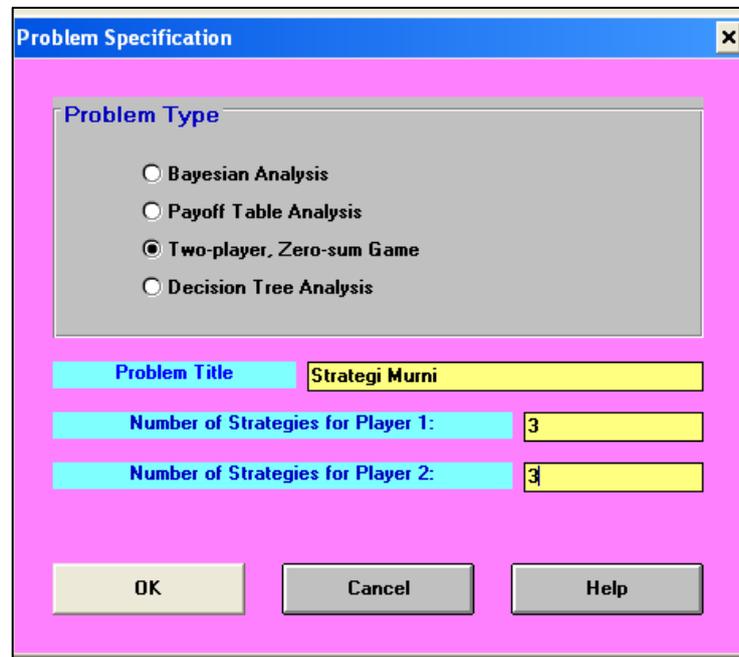
1. Strategi Murni

Langkah untuk mencari *saddle point* atau titik pelana untuk kedua minimarket dapat pada studi kasus no. 1 dapat dilihat sebagai berikut:

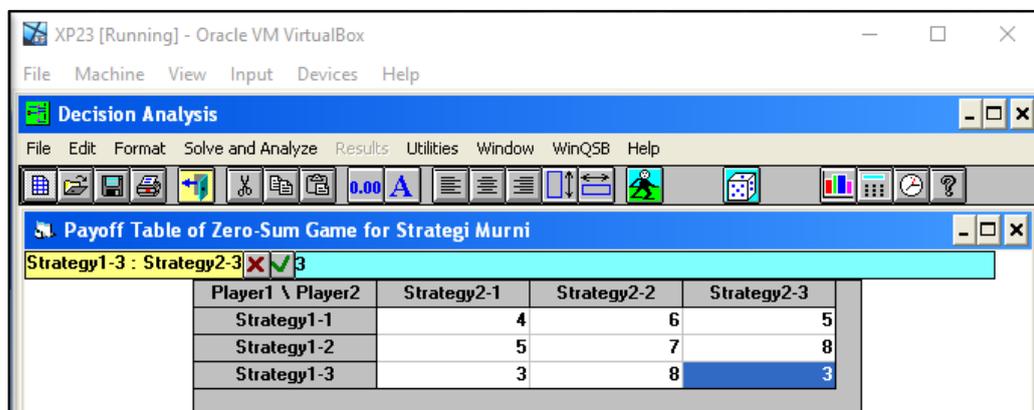
- a. Membuka *software WinQSB* dengan cara klik *Start > All Program > Folder WinQSB > Decision Analysis*,
- b. Pada layar akan muncul layar baru, kemudian klik *New Problem > Two*

Player, Zero-sum Game pada problem type,

- c. Setelah itu akan muncul Problem Title, masukkan judul misalkan “Strategi Murni”,
- d. Kemudian melakukan input *Number of strategies for player 1* sebanyak 3, dan *Number of strategies for player 2* sebanyak 3, kemudian klik “Ok”,



- e. Langkah berikutnya adalah memasukkan data-data strategi pada kolom *strategy 1-1*, 1-2, 1-3 dan baris *strategy 2-1*, 2-2, dan 2-3. Berikut ini kolom data pada setiap strategi,



- f. Kemudian klik *solve and analyze*, dan akan diperoleh *output* seperti di bawah ini:

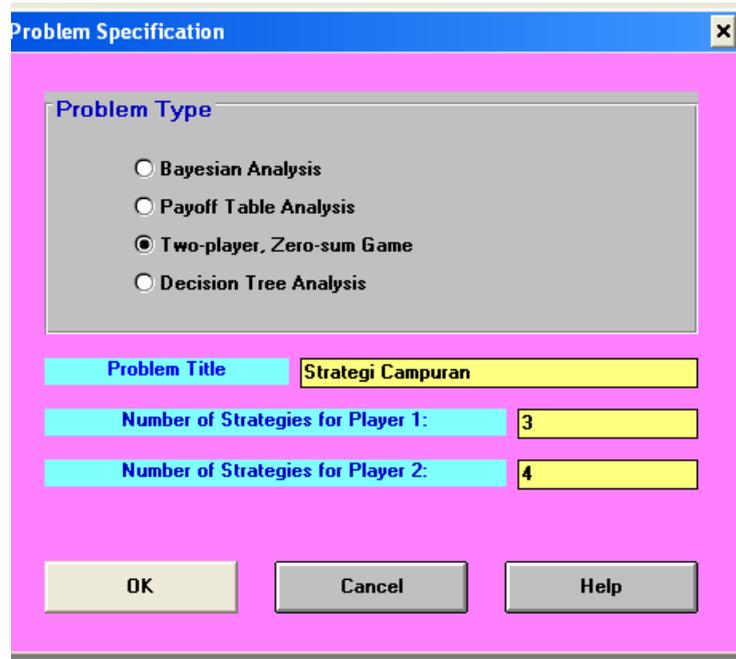
11-12-2021	Player	Strategy	Dominance	Elimination Sequence
1	1	Strategy1-1	Dominated by Strategy1-2	1
2	1	Strategy1-2	Not Dominated	
3	1	Strategy1-3	Dominated by Strategy1-2	4
4	2	Strategy2-1	Not Dominated	
5	2	Strategy2-2	Dominated by Strategy2-1	2
6	2	Strategy2-3	Dominated by Strategy2-1	3
***	Saddle	Point	(Equilibrium)	is Achieved!!
	The Best	Pure	Strategy for Player 1:	Strategy1-2
	The Best	Pure	Strategy for Player 2:	Strategy2-1
	Stable	Payoff	for Player 1 =	5
	Player 1	is	Winning!!!	

g. Langkah terakhir melakukan analisis terhadap nilai *payoff* yang didapatkan Berdasarkan *output* pada *WinQSB*.

2. Strategi Campuran

Langkah untuk mencari *saddle point* atau titik pelana untuk kedua minimarket dapat pada studi kasus no. 2 dapat dilihat sebagai berikut:

- a. Membuka *software WinQSB* dengan cara klik *Start > All Program > Folder WinQSB > Decision Analysis*,
- b. Pada layar akan muncul layar baru, kemudian klik *New Problem > Two-Player, Zero-sum Game* pada problem type,
- c. Setelah itu akan muncul Problem Title, masukkan judul misalkan “Strategi Campuran”,
- d. Kemudian melakukan input *Number of strategies for player 1* sebanyak 3, dan *Number of strategies for player 2* sebanyak 4, kemudian klik “Ok”



- e. Langkah berikutnya adalah memasukkan data-data strategi pada kolom *strategy* 1-1, 1-2, 1-3 dan baris *strategy* 2-1, 2-2, 2-3, dan 2-4. Berikut ini kolom data pada setiap strategi,

Player1 \ Player2	Strategy2-1	Strategy2-2	Strategy2-3	Strategy2-4
Strategy1-1	2	5	9	4
Strategy1-2	1	2	2	7
Strategy1-3	7	1	4	2

- f. Kemudian klik *solve and analyze*, dan akan diperoleh *output* seperti di bawah ini:

11-12-2021	Player	Strategy	Dominance	Elimination Sequence
1	1	Strategy1-1	Not Dominated	
2	1	Strategy1-2	Not Dominated	
3	1	Strategy1-3	Not Dominated	
4	2	Strategy2-1	Not Dominated	
5	2	Strategy2-2	Not Dominated	
6	2	Strategy2-3	Dominated by Strategy2-2	
7	2	Strategy2-4	Not Dominated	
	Player	Strategy	Optimal Probability	
1	1	Strategy1-1	0.62	
2	1	Strategy1-2	0.06	
3	1	Strategy1-3	0.32	
1	2	Strategy2-1	0.36	
2	2	Strategy2-2	0.26	
3	2	Strategy2-3	0	
4	2	Strategy2-4	0.38	
	Expected	Payoff	for Player 1 =	3.54

g. Langkah terakhir melakukan analisis terhadap nilai *payoff* yang didapatkan Berdasarkan *output* pada *WinQSB*.

5.4 Analisis Game Theory

1. Software POM-QM

Analisa *output* berdasarkan penyelesaian masalah teori permainan (*Game Theory*) dengan menggunakan *software* POM-QM *For Windows* akan menghasilkan *output* berupa hasil dari penyelesaian masalah. Berikut ini adalah analisis interpretasi hasil atau *output* dari permasalahan strategi murni dan permasalahan strategi campuran pada teori permainan dengan menggunakan *software* POM-QM *For Windows*.

a. Strategi Murni

Berikut adalah analisis interpretasi hasil dari permasalahan strategi murni dari minimarket Ceriwis dan Danjer untuk mencari *saddle point* berdasarkan *output* yang diperoleh dari *software* POM-QM *For Windows*.

1) Game Theory result

Berdasarkan *output* pada *Game Theory result* didapatkan hasil untuk nilai *saddle point* sebesar 5. Hal ini dapat diartikan bila minimarket Ceriwis menginginkan keuntungan lebih besar, namun hanya akan mendapat maksimal 5 dan pada

minimarket Danjer meskipun menginginkan kerugian sekecil mungkin, akan tetap mengalami kerugian maksimum sebesar 5.

2) *Maximin dan minimax*

Berdasarkan pada hasil output *maximin* dan *minimax*, diperoleh nilai *maximin* sebesar 5 dan nilai *minimax* sebesar 5. Oleh karena itu, dapat dikatakan bahwa strategi yang optimal untuk digunakan pada minimarket Ceriwis adalah strategi Harga Diskon (karena baris ini terdapat nilai *maximin*) dan strategi yang optimal untuk minimarket Danjer adalah strategi Harga Bundle juga (karena pada strategi ini terdapat nilai *minimax*).

3) *Result in list form*

Nilai *result in list form* merupakan persentase dari tiap strategi yang ada di tiap minimarket, untuk Ceriwis pada strategi harga bundle sebesar 0%, strategi harga diskon sebesar 100%, dan strategi harga normal sebesar 0%. Pada Danjer untuk pada strategi harga bundle sebesar 100%, strategi harga diskon sebesar 0%, dan strategi harga normal sebesar 0%.

b. Strategi Campuran

Berikut adalah analisis interpretasi hasil dari permasalahan strategi campuran dari minimarket Ceriwis dan Danjer untuk mencari *saddle point* berdasarkan *output* yang diperoleh dari *software POM-QM For Windows*.

1) *Game Theory result*

Berdasarkan *output* pada *Game Theory result* didapatkan hasil untuk nilai *pay off* sebesar 3,54. Hal ini dapat diartikan bila minimarket Ceriwis menginginkan keuntungan lebih besar, namun hanya akan mendapat maksimal 3,54 dan pada minimarket Danjer meskipun menginginkan kerugian sekecil mungkin, akan tetap mengalami kerugian maksimum sebesar 3,54.

2) *Maximin dan minimax*

Berdasarkan pada output *maximin* dan *minimax* diperoleh nilai *maximin* sebesar 2 dan nilai *minimax* sebesar 5. Sehingga value yang didapatkan adalah nilai yang antara 2 dan 5 dengan menggunakan strategi harga bundle pada *maximin* dan harga diskon pada *minimax*.

3) *Result in list form*

Nilai *result in list form* merupakan persentase dari tiap strategi yang ada di tiap minimarket, untuk minimarket Ceriwis untuk pada strategi harga bundle sebesar 62%, harga diskon sebesar 6%, dan pada strategi harga normal sebesar 32%. Danjer

pada strategi harga bundle sebesar 36%, strategi harga diskon sebesar 26%, strategi harga normal 0% dan strategi harga murah sebesar 38%.

2. *Software WinQSB*

Analisa *output* berdasarkan penyelesaian masalah teori permainan (*Game Theory*) dengan menggunakan *software WinQSB* akan menghasilkan *output* berupa hasil dari penyelesaian masalah. Berikut ini adalah analisis interpretasi hasil atau *output* dari permasalahan strategi murni dan permasalahan strategi campuran pada teori permainan dengan menggunakan *software WinQSB*.

a. Strategi Murni

Berikut adalah analisis interpretasi hasil dari permasalahan strategi murni dari minimarket Ceriwis dan Danjer untuk mencari *saddle point* berdasarkan *output* yang diperoleh dari *software WinQSB*. Berdasarkan output pada WinQSB didapatkan output Saddle Point Equilibrium atau kesetimbangan. Pada soal pertama ini permasalahan yang ada merupakan pengolahan strategi murni. Pada Zero-sum Game didapatkan bahwa nilai payoff dari permasalahan minimarket Ceriwis adalah sebesar 5. Strategi murni terbaik pada player 1 menggunakan Strategi 1-2 dengan strategi Harga Diskon dengan mendominasi strategi 1-1 dan 1-3 atau strategi Harga Bundle dan Harga Normal kemudian pada Strategi murni terbaik pada player 2 menggunakan Strategi 2-1 dengan strategi Harga Bundle dengan mendominasi strategi 2-2 dan 2-3 atau strategi Harga Diskon dan Harga Normal. Player 1 atau Ceriwis mengalami keuntungan maksimum sebanyak 5.

b. Strategi Campuran

Berikut adalah analisis interpretasi hasil dari permasalahan strategi murni dari minimarket Ceriwis dan Danjer untuk mencari *saddle point* berdasarkan *output* yang diperoleh dari *software POM-QM For Windows*. Berdasarkan output pada WinQSB didapatkan output Saddle Point optimal probability atau peluang yang paling optimal, sehingga strategi yang memiliki peluang terbesar memiliki kesempatan dalam mengantisipasi kerugian dan meningkatkan keuntungan pada masing-masing minimarket. Pada soal kedua ini permasalahan yang ada merupakan pengolahan strategi campuran. Pada Zero-sum Game didapatkan bahwa nilai payoff dari permasalahan minimarket Ceriwis adalah sebesar 3,54. Strategi campuran terbaik pada player 1 terdapat Strategi 1-1 dengan strategi Harga Bundle dengan peluang 0,62; pada strategi 1-2 dengan strategi Harga Diskon dengan peluang 0,06; dan pada strategi 1-3 dengan

Harga Normal dengan peluang 0,32. Sehingga strategi yang tertinggi yaitu Harga Bundle dengan peluang tertinggi yang akan digunakan. Strategi campuran terbaik pada player 2 terdapat Strategi 2-1 dengan strategi Harga Bundle dengan peluang 0,36; pada strategi 2-2 dengan strategi Harga Diskon dengan peluang 0,26; pada strategi 2-3 dengan Harga Normal dengan peluang 0, pada strategi 2-4 dengan Harga Murah dengan peluang 0,38 . Sehingga strategi yang tertinggi yaitu Harga Murah dengan peluang tertinggi yang akan digunakan. Player 1 atau Ceriwis mengalami keuntungan maksimum sebanyak 3,54.