

Modul Praktikum Simulasi Komputer 2022

Tim Penyusun
Brav Deva Bernandhi, ST., MT
Ali Wedo Sarjono, ST., M.Kom
Muchamad Maknun, ST



ProModel



extendsim

Arena[®] Simulation Software



Laboratorium Decision Support System
Program Studi Teknik Industri
Fakultas Teknologi Industri
Universitas Islam Sultan Agung

DAFTAR ISI

HALAMAN JUDUL.....	i
DAFTAR ISI.....	ii
MODUL 1 SIMULASI KOMPUTER.....	4
1.1 Definisi Simulasi Komputer.....	4
1.2 Kelebihan dan Kekurangan Metode Simulasi.....	4
1.3 Penyebab Kegagalan Simulasi.....	5
1.4 Langkah-langkah Melakukan Studi Simulasi.....	5
MODUL II ANTRIAN.....	9
2.1 Teori Antrian.....	9
2.2 Konsep - Konsep Dasar Antrian.....	9
2.2.1 Tujuan.....	9
2.2.2 Unsur - Unsur Dasar Model Antrian.....	10
2.2.3 Elemen – Elemen Pokok Dalam Sistem Antrian.....	11
2.2.4 Model - Model Antrian.....	12
2.3 Biaya Model Antrian.....	14
2.3.1 Model - Model Keputusan Antrian.....	14
MODUL III TEKNIK PENGAMBILAN DATA.....	16
3.1 Pendeskripsian Sistem Nyata.....	16
3.1.1 Layout Sistem Produksi Atau Jasa.....	16
3.1.2 Tata Urutan Proses.....	16
3.1.3 Jumlah Server dan Entitas.....	16
3.1.4 Aktivitas dan Event.....	17
3.2 Model Atrian.....	18
3.3 Pengumpulan Data Pengamatan.....	18
3.4 Rekapitulasi Data Pengamatan.....	19
MODUL IV PROMODEL.....	19
4.1 Pengertian Promodel.....	20
4.2 Elemen Dasar Promodel.....	20

4.3	Fitting Data Into Distribution.....	24
3.3.1	Fitting Data Dengan Stat Fit	24
3.3.2	Fitting Data Secara Manual	25
4.4	Pembuatan Model	28
4.5	Study Kasus Software Promodel.....	32
	MODUL V EXTEND.....	37
5.1	<i>Extend</i>	37
5.2	Bagian – Bagian <i>Extend</i>	37
5.2.1	<i>DE.LIX</i>	37
5.2.2	<i>GENERIC.LIX</i>	38
5.2.3	<i>STATS.LIX</i>	38
5.2.4	<i>MFG.LIX</i>	39
	MODUL VI SOFTWARE ARENA	40
6.1	<i>Activity Cycle Diagram</i>	40
6.2	Definisi ARENA	42
6.3	<i>ARENA Home Screen</i>	43
6.4	Pengenalan <i>Software ARENA</i>	44
	MODUL VII REPLIKASI, VERIFIKASI, DAN VALIDASI	48
7.1	Perhitungan Replikasi	48
7.2	Verifikasi.....	48
7.2.1	Teknik Verifikasi	49
7.3	Validasi	49
7.3.1	Validasi	49
	MODUL VIII EVALUASI PERFORMANSI DAN RANCANGAN MODEL ALTERNATIF (PROMODEL)	51
	DAFTAR PUSTAKA	66

MODUL 1

SIMULASI KOMPUTER

1.1 Definisi Simulasi Komputer

Jerry Banks mendefinisikan simulasi sebagai tiruan atau imitasi bekerjanya suatu proses atau sistem nyata (Banks, 1998). Simulasi komputer adalah komputasi yang memodelkan perilaku sebuah sistem (nyata maupun tidak) dalam suatu rentang waktu. Simulasi diskrit event adalah model yang mempresentasikan suatu sistem dan beroperasi dalam suatu rentang waktu dengan perubahan variabel status terjadi pada titik-titik waktu terpisah. Titik-titik waktu adalah titik-titik waktu terjadinya event yang merupakan kejadian sesaat (*instantaneous occurrence*) yang mengubah status sistem (Law. A. M, 2000).

Simulasi digunakan untuk menggambarkan perilaku sebuah sistem nyata ketika proses pembentukan model matematis sulit dilakukan karena sistem nyata yang kompleks.

1.2 Kelebihan dan Kekurangan Metode Simulasi

Simulasi didefinisikan sebagai suatu model sistem dimana komponennya dipresentasikan oleh proses-proses aritmatika dan logika yang dijalankan pada komputer untuk memperkirakan sifat-sifat dinamis sistem tersebut.

Kelebihan utama dari metode simulasi adalah sebagai berikut (Law. A. M, 2000):

1. Model yang telah dibangun dapat digunakan berulang-ulang untuk menganalisis model atau kebijakan yang baru.
2. Kebanyakan sistem yang nyata yang kompleks dengan elemen-elemen stokastik yang ada tidak dapat dijelaskan dengan model matematik yang dapat dievaluasi secara analitis sehingga simulasi seringkali merupakan satu-satunya cara pemecahan yang mungkin.
3. Simulasi memungkinkan seseorang untuk mengestimasi performansi dari sistem yang ditinjau dalam kondisi yang diinginkan.
4. Alternatif-alternatif dari rancangan sistem dapat dibandingkan dengan simulasi untuk melihat mana yang terbaik menurut persyaratan yang diinginkan.
5. Dalam simulasi kita dapat menjaga kondisi penelitian dengan lebih baik

dibanding bila dilakukan pada sistemnya langsung.

6. Simulasi memungkinkan kita untuk melakukan studi terhadap sebuah sistem dengan jangka waktu yang lama dalam waktu yang singkat.

Kekurangan metode simulasi adalah sebagai berikut :

1. Setiap kali kita menjalankan model simulasi stokastik akan menghasilkan estimasi dari karakteristik model sesungguhnya untuk satu set input parameter tertentu sehingga model tersebut harus dijalankan beberapa kali untuk setiap set input parameter.
2. Model simulasi seringkali mahal dan membutuhkan waktu yang lama untuk dibuat.

1.3 Penyebab Kegagalan Simulasi

Ada beberapa hal yang harus diperhatikan yang dapat membuat simulasi yang dilakukan tidak sesuai dengan sistem yang ditinjau (Law. A. M, 2000) :

1. Kegagalan dan menentukan tujuan dilakukannya simulasi.
2. Model yang kurang detail.
3. Menganggap bahwa simulasi hanya bertumpu pada program komputer.
4. Kegagalan dalam mengolah data secara statistik.
5. Menggunakan *software* simulasi yang *error*.
6. Kesalahan dalam menggunakan animasi.
7. Kegagalan dalam menentukan kerandoman dalam sistem nyata.
8. Menggunakan distribusi yang umum (normal atau *uniform*) sebagai input simulasi.
9. Menganalisis output data dari hasil sekali menjalankan program dengan formula statistik yang diasumsikan independen.
10. Hanya melakukan sekali replikasi untuk suatu sistem tertentu dan melakukannya sebagai hasil data yang valid.
11. Membandingkan sistem alternatif hanya dengan sekali replikasi.
12. Menggunakan ukuran performansi yang salah.

1.4 Langkah-langkah Melakukan Studi Simulasi

Langkah-langkah yang harus dilakukan dalam melakukan studi simulasi

adalah sebagai berikut :

1. Formulasi Masalah dan Perencanaan Studi

Studi diawali dengan pernyataan jelas tentang pokok masalah dan tujuan penelitian yang ingin dicapai. Setelah itu pelaksana studi direncanakan dengan mempertimbangkan keterbatasan sarana dan prasarana yang tersedia. Selain itu juga ditentukan :

- Model yang digunakan
- Kriteria performansi yang akan dipakai
- Kerangka konfigurasi sistem yang akan ditinjau

2. Pengumpulan Data dan Perancangan Model

Pengumpulan data dan informasi dari sistem yang ditinjau diperlukan untuk mengetahui bagaimana sistem bekerja dan menentukan distribusi peluang bagi proses random yang digunakan dalam model. Kekurangan data akan mengurangi keakuratan model dan sebaliknya data yang terlalu kecil akan membutuhkan biaya besar dan waktu pengumpulan yang lama.

3. Validasi Model

Tahap ini dilakukan dengan melakukan pengecekan asumsi-asumsi yang ditetapkan dalam pembuatan model serta melibatkan ahli yang mengenal sistem dengan baik.

4. Penyusunan Program Komputer dan Verifikasi

Pemilihan perangkat lunak yang akan digunakan dalam simulasi mempunyai pengaruh yang besar terhadap kesuksesan penelitian, yaitu dalam hal keakuratan model, validitas model dan waktu eksekusi, dan waktu penyelesaian penelitian secara keseluruhan. Beberapa teknik untuk melakukan verifikasi program antara lain :

- Melakukan pelacakan jalannya program (*trace*)
- Pengembangan program dalam bentuk sub program atau modular
- Menggunakan *interactive debugger*
- Meneliti kelayakan hasil program
- Menggunakan estimasi

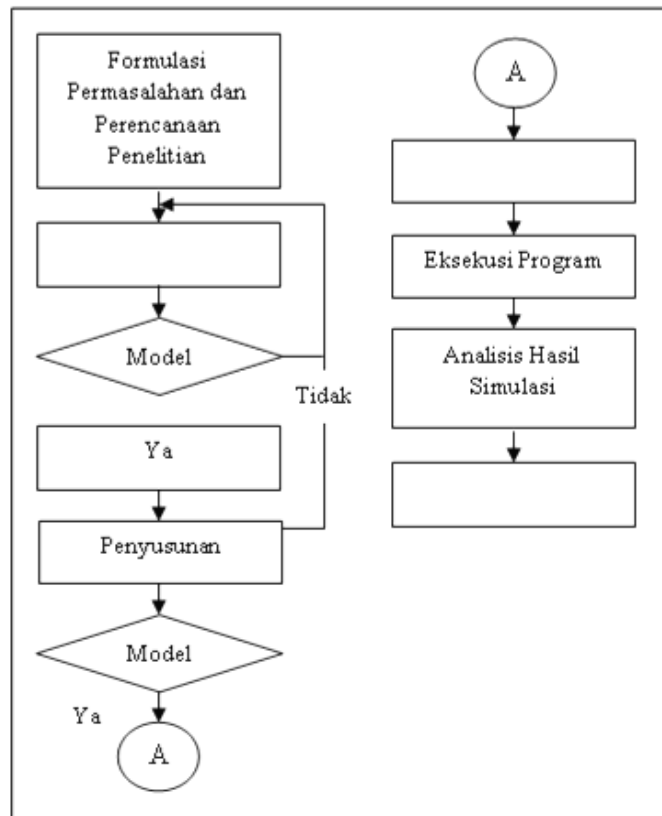
5. Uji coba program

Uji coba program dilakukan untuk keperluan validasi pada tahap berikutnya.

6. Validasi Program

Hasil uji coba program diteliti kembali untuk mengetahui apakah ada kesalahan dalam program atau model yang digunakan, cara yang dapat dilakukan antara lain :

- Menguji sensitivitas output model terhadap perubahan input.
- Membandingkan output simulasi dengan performansi sistem di masa lalu (data historis). Jika hasil simulasi dengan data historis tidak berbeda secara signifikan, maka dapat dikatakan model simulasi sudah valid.



Gambar 1.1 Metodologi Perancangan Model Simulasi

7. Perancangan Eksperimen

Pada tahap ini diputuskan perancangan sistem seperti apa yang akan disimulasikan dari beberapa alternatif yang mungkin ada. Untuk setiap rancangan sistem yang akan disimulasikan perlu ditentukan hal-hal sebagai berikut :

- Panjang replikasi dalam setiap eksekusi program
 - Jumlah replikasi
 - Inisiasi program
 - Panjang periode *transient*
8. Eksekusi Program
- Eksekusi program dilakukan sesuai dengan perancangan eksperimen yang telah dibuat.
9. Analisis Output Simulasi
- Data output simulasi digunakan untuk mengestimasi kriteria performansi sistem yang diteliti. Hasil estimasi ini kemudian digunakan untuk menjawab tujuan studi.
10. Dokumentasi, Presentasi dan Implementasi
- Dokumentasi yang baik diperlukan karena tidak jarang model simulasi yang telah dibuat akan dipakai untuk lebih dari satu aplikasi. Akhirnya hasil dari studi simulasi perlu di implementasikan, untuk itu kredibilitas model simulasi yang dibangun harus tinggi agar dapat digunakan secara nyata.

MODUL II

ANTRIAN

2.1 Teori Antrian

Teori antrian diciptakan dan mulai dikembangkan oleh seorang ahli matematika Denmark yang bernama A.K. ERLANG.

Fenomena menunggu adalah hasil langsung dari keacakan dalam operasi sarana pelayanan (Taha, 1997). Secara umum kedatangan pelanggan tidak diketahui sebelumnya, karena jika dapat diketahui maka pengoperasian sarana tersebut dapat dijadwalkan sedemikian rupa sehingga akan sepenuhnya menghilangkan keharusan untuk menunggu.

Walaupun teori antrian sendiri tidak langsung memecahkan persoalan, teori antrian menyumbangkan informasi penting yang diperlukan untuk membuat suatu pengambilan keputusan seperti dengan cara memprediksi beberapa karakteristik dari antrian seperti waktu antar kedatangan dan waktu pelayanan. Apabila kedua waktu ini dapat diketahui secara pasti maka akan sangat mudah dalam melakukan penjadwalan fasilitas pelayanan. Akan tetapi waktu antar kedatangan dan waktu pelayanan ini cenderung tidak pasti, sehingga menimbulkan dua masalah yang saling berkaitan, pertama apabila fasilitas pelayanan terlalu banyak akan menurunkan tingkat utilisasi fasilitas pelayanan tersebut yang berarti meningkatkan waktu menganggur dari fasilitas tersebut sehingga merugikan investasi yang telah dilakukan. Kedua, jika fasilitas pelayanan kurang maka akan terjadi antrian yang lama yang nantinya akan menimbulkan biaya sosial seperti kenyamanan pengguna yang terganggu.

2.2 Konsep - Konsep Dasar Antrian

2.2.1 Tujuan

Tujuan dasar model antrian adalah untuk meminimumkan total biaya, yaitu biaya langsung penyediaan fasilitas pelayanan dan biaya tidak langsung karena pelanggan harus menunggu untuk dilayani, bila suatu sistem mempunyai fasilitas pelayanan lebih dari jumlah optimal, ini berarti membutuhkan investasi modal yang berlebihan, tetapi bila jumlahnya kurang dari optimal hasilnya adalah tertundanya pelayanan (Subagyo dkk,1983).

Situasi keputusan sering kali timbul dimana unit satuan yang datang untuk memperoleh pelayanan (orang, barang) harus menunggu sebelum memperoleh pelayanan yang diinginkan. Apabila aturan yang mengatur kedatangan (arrival), unit penerima pelayanan, waktu pelayanan (Service Time) dan urutan kedatangan satuan penerimaan pelayanan (spp) diketahui (mungkin berdasarkan pengalaman atau hasil penelitian), maka sifat atau ciri – ciri dalam situasi antrian dapat dipelajari dengan menggunakan peralatan matematika dengan mudah.

Tujuan mempelajari teori antrian adalah untuk menentukan beberapa karakteristik yang menjadi ukuran performansi dari sistem pelayanan yang dipelajari.

1. Berapa lama obyek yang harus dilayani tersebut menunggu sampai dilayani.
2. Persentase waktu mengganggu dari fasilitas pelayanan.

Dalam hal ini semakin lama waktu menunggu yang dialami oleh obyek yang membutuhkan pelayanan, akan semakin kecil kemungkinan fasilitas mengganggu, begitu juga sebaliknya. Ukuran performansi yang telah diperoleh selanjutnya biasa digunakan untuk memilih tingkat pelayanan yang optimal diantara situasi yang bertentangan tersebut.

2.2.2 Unsur - Unsur Dasar Model Antrian

Walaupun pola kedatangan dan kepergian adalah faktor – faktor yang penting dalam analisis antrian, terdapat juga faktor- faktor penting lain dalam pengembangan model model antrian (Taha, 1997) :

1. Faktor pertama adalah cara memilih pelanggan dari antrian untuk memulai pelayanan, biasanya disebut peraturan pelayanan.
2. Faktor kedua berkaitan dengan rancangan sarana tersebut dari pelaksanaan pelayanan. Sarana pelayanan lebih dari satu pelayan dan menawarkan pelayanan yang sama sehingga dikatakan memiliki pelayanan sejajar. Sarana pelayanan yang dapat dilalui pelanggan sebelum pelayanan diselesaikan. Situasi yang dihasilkan biasanya dikenal sebagai antrian serial (tandem queue). Rancangan paling umum dari sebuah sarana pelayanan mencakup baik stasiun pengolahan serial atau paralel. Ini menghasilkan antrian jaringan (net work queue).
3. Faktor ketiga berkaitan dengan ukuran antrian yang diijinkan.

4. Faktor keempat berkaitan dengan sifat sumber yang meminta pelayanan (kedatangan pelanggan). Sumber pemanggilan (calling source) dapat menghasilkan sejumlah terbatas pelanggan atau secara teoritis sejumlah tak terbatas pelanggan.

2.2.3 Elemen – Elemen Pokok Dalam Sistem Antrian

Elemen – elemen dasar dari model antrian tergantung pada faktor – faktor sebagai berikut [Dubagyo dkk, 1985] :

1. Sumber Masukan (*Input*)

Sumber masukan dari suatu sistem antrian dapat terdiri atas suatu populasi orang, barang, komponen atau kertas kerja yang datang pada sistem yang dilayani. Bila populasi relatif besar sering di anggap bahwa hal itu merupakan besaran tak terbatas. Anggapan ini adalah hampir umum karena perumusan sumber masukan yang tak terbatas lebih sederhana dari pada sumber yang terbatas. Suatu populasi dinyatakan besar bila populasi tersebut besar dibanding dengan kapasitas pelayanan.

2. Pola Kedatangan

Pola kedatangan yang biasa terjadi pada suatu sistem antrian diantaranya adalah pola kedatangan teratur, pola kedatangan random (acak), pola kedatangan yang dipengaruhi aspek lain dan sebagainya. Besarnya selang waktu suatu kedatangan dengan kedatangan berikutnya disebut selang waktu kedatangan.

3. Kepanjangan Antrian

Banyak sistem antrian dapat menampung jumlah individu-individu yang relatif besar, tetapi ada beberapa sistem yang mempunyai kapasitas terbatas. Bila kapasitas antrian menjadi faktor pembatas jumlahnya individu yang dapat dilayani dalam sistem secara nyata, berarti sistem mempunyai kepanjangan antrian terbatas dan model antrian terbatas untuk menganalisa sistem tersebut

4. Disiplin Antrian

Disiplin antrian adalah aturan dalam mana para pelanggan dilayani. Aturan ini dapat didasarkan pada yang pertama masuk, pertama keluar (PMPK, kalam bahasa inggris *FIFO*) (yakni, pelayanan menurut urutan kedatangan),

yang terakhir masuk pertama keluar (PMPK, dalam bahasa Inggris disingkat *FIFO*) (contohnya, pelanggan yang datang paling akhir mendapat pelayanan yang berikutnya), secara acak, atau berdasarkan prioritas. *Service in Random Order* (SIRO) artinya panggilan didasarkan pada peluang random, tidak mempertimbangkan siapa dulu yang datang. *Priority Service* (PS) artinya prioritas pelayanan diberikan pada mereka yang mempunyai prioritas yang lebih tinggi dibandingkan dengan mereka yang mempunyai prioritas lebih rendah, meskipun yang terakhir ini kemungkinan sudah lebih dahulu tiba dalam garis tunggu.

5. Mekanisme Pelayanan

Pola pelayanan biasanya dicirikan oleh waktu pelayanan (*service time*), yaitu waktu yang dibutuhkan seorang pelayan untuk melayani seorang pelanggan. Waktu pelayanan ini dapat bersifat deterministik, atau berupa suatu variabel acak yang distribusi probabilitasnya dianggap telah diketahui. Besaran ini dapat tergantung jumlah pelanggan yang telah berada di dalam fasilitas pelayanan, atau tidak bergantung pada keadaannya. Bila tidak disebutkan secara khusus, maka anggapan dasarnya adalah bahwa satu pelayan saja dapat melayani secara tuntas urusan seorang pelanggan.

6. Keluar (*Exit*)

Sesudah individu selesai dilayani, dia keluar (*exit*) dari sistem. Sesudah keluar, dia mungkin bergabung dengan populasi asal dan mempunyai probabilitas yang sama untuk memasuki sistem kembali, atau dia mungkin bergabung dengan populasi lain yang mempunyai probabilitas yang lebih kecil dalam hal kebutuhan pelayanan tersebut kembali.

2.2.4 Model - Model Antrian

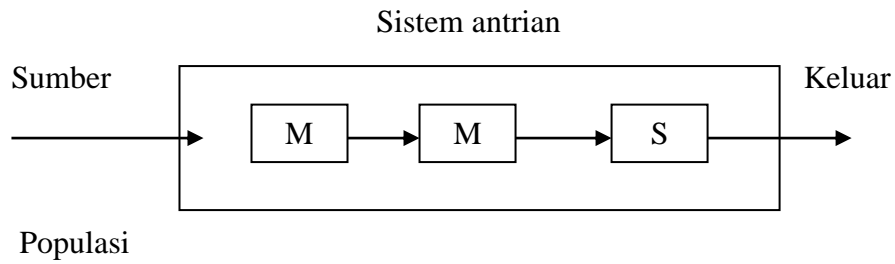
Ada empat tipe model antrian, yaitu (Subagyo, dkk, 1985) :

1. *Single Chanel - Single Phase*

Sistem antrian ini terdiri dari satu fasilitas pelayanan, dimana objek yang masuk pada sistem ini akan dilayani oleh fasilitas tunggal.

M = antrian

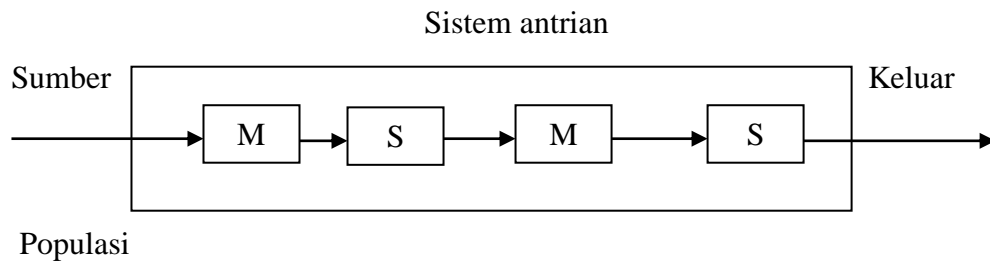
S = fasilitas pelayanan (*server*)



Gambar 2.1 *Single Chanel - Single Phase*

2. *Single Chanel - Multi Phase*

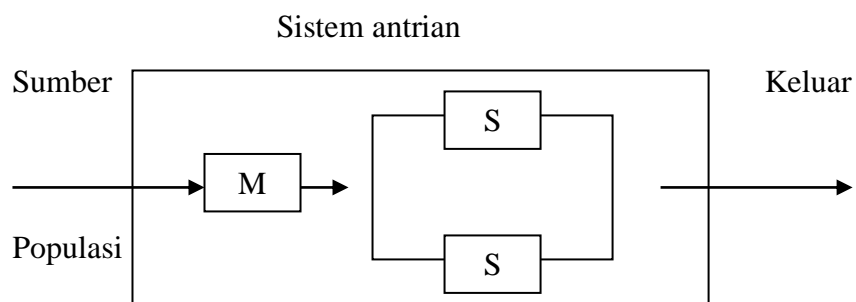
Sistem antrian ini terdiri dari multi fasilitas pelayanan yang dilaksanakna secara berurutan.



Gambar 2.2 *Single Chanel - Multi Phase*

3. *Multi Chanel - Single Phase*

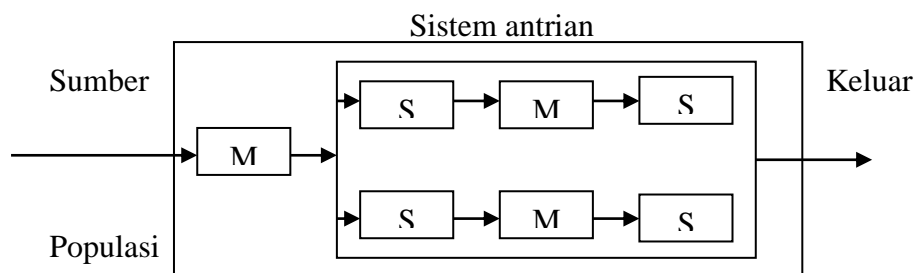
Sistem antrian ini terdiri dari multi (lebih dari satu fasilitas pelayanan) dimana objek yang masuk dalam sistem akan dilayani oleh fasilitas yang sedang menganggur.



Gambar 2.3 *Multi Chanel - Single Phase*

4. *Multi Chanel - Multi Phase*

Bentuk ini mempunyai beberapa fasilitas pelayanan pada setiap tahap dengan beberapa stasiun pelayanan.



Gambar 2.4 *Multi Chanel - Multi Phase*

2.3 Biaya Model Antrian

Tujuan biaya model antrian adalah menghitung tingkat pelayanan (kecepatan pada pelayanan atau jumlah stasiun pelayanan) yang diimbangi dengan biaya yang saling bertentangan, sebagai berikut :

1. Biaya stasiun pelayanan

Biaya stasiun pelayanan diasumsikan dengan biaya operasi fasilitas pelayanan yang mencakup biaya untuk investasi, biaya pemasangan, maupun biaya – biaya variabel (gaji pegawai atau operator, pengeluaran tambahan untuk pemeliharaan).

2. Biaya menunggu dari pelanggan

Biaya ini timbul sebagai akibat adanya pelanggan menunggu dan biaya ini merupakan kerugian yang diderita oleh perusahaan. Biaya menunggu (*cost waiting*) yang dapat diperhitungkan sebagai berikut :

- a. Biaya menganggurnya karyawan
- b. Kehilangan pelanggan
- c. Tingkat persediaan
- d. Kemacetan sistem

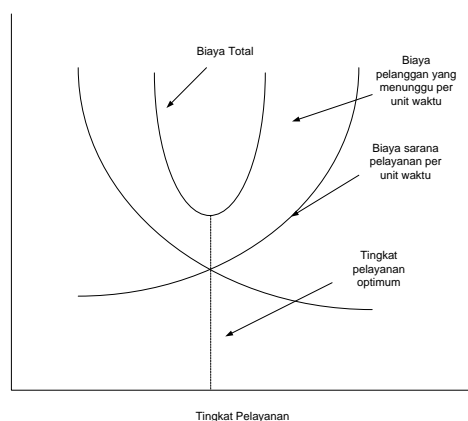
Penambahan tingkat pelayanan akan mengurangi waktu menunggu pelanggan dan sebaliknya bila biaya pengoperasian fasilitas meingkat karena peningkatan tingkat pelayanan, maka biaya menunggu akan menurun.

2.3.1 Model - Model Keputusan Antrian

Pengambilan keputusan menyangkut antrian berkaitan dengan peningkatan hasil karya sistem melalui penggunaan model keputusan yang sesuai. Model ini dibangun dengan menggunakan sifat operasi yang cocok, pada akhirnya menetapkan parameter optimim, parameter yang mana mencakup laju pelayanan, jumlah pelayanan atau panjang antrian maksimum yang diperkenankan.

Optimasi parameter dapat dilihat dari bermacam-macam cara tergantung pada keinginan pengambil keputusan. Pandangan yang paling umum didasarkan pada keputusan yang meminimumkan jumlah pelayanan dan antrian persatuan biaya waktu (Siagan, 1987).

Sifat dari beberapa situasi antrian mencakup penggunaan model-model keputusan biaya. Khususnya, biaya menunggu paling sulit ditentukan. Untuk mempelajari hal ini situasi antrian digolongkan didalam tiga kategori besar berikut ini:



Gambar 2.5 Model Keputusan Biaya dalam Model Antrian

Model yang ideal adalah kalau kita menentukan taksiran terpercaya dari parameter biaya yang diperlukan. Kadang-kadang sukar bahkan tidak mungkin menaksir parameter biaya terutama yang menyangkut waktu tunggu. Oleh karena itu kita harus mencari kriteria optimalisasi yang lain seperti apa yang dinamakan model tingkat aspirasi atau *aspirationlevel* yang digunakan bila model biaya tidak dapat lagi digunakan dalam praktek (Siagian, 1987).

MODUL III

TEKNIK PENGAMBILAN DATA

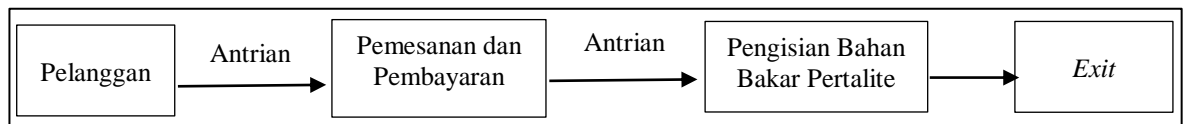
3.1 Pendeskripsian Sistem Nyata

Dalam melakukan simulasi komputer langkah awal yang dilakukan adalah melakukan pengambilan data. Data yang diambil dari salah satu sistem tertentu yang nantinya akan dibuat model untuk mensimulasikan ke dalam sebuah software. Pendeskripsian sistem nyata ini berguna untuk menentukan model antrian yang digunakan serta mempermudah dalam proses pengambilan waktu pada setiap layanan.

3.1.1 Layout Sistem Produksi Atau Jasa

Pembuatan layout sistem produksi atau jasa digunakan untuk menggambarkan keadaan sistem nyata dari proses awal sampai akhir.

Berikut ini merupakan contoh layout sistem nyata pada Pengisian Peralite di SPBU Self Service Srandol Semarang :



Gambar 3.1 Model *Layout* Awal SPBU *Self Service* Srandol

3.1.2 Tata Urutan Proses

Merupakan penjelasan mengenai urutan proses suatu pelayanan sistem produksi atau jasa dari awal proses pelayanan sampai akhir proses pelayanan.

Beikut merupakan contoh tata urutan proses pada Pengisian Peralite di SPBU Self Service Srandol Semarang : Pelanggan datang dan mengantri untuk pemesanan dan pembayaran bahan bakar peralite, kemudian setelah selesai melakukan pemesanan dan pembayaran pelanggan mengantri untuk pengisian bahan bakar peralite.

3.1.3 Jumlah Server dan Entitas

Berikut merupakan contoh jumlah server dalam sistem antrian Pengisian Peralite di SPBU Self Service Semarang:

Tabel 3.1 Jumlah server dalam sistem jasa

No	Keterangan	Jumlah <i>Server</i>
1	Pemesanan dan pembayaran	1
2	Pengisian bahan bakar	1

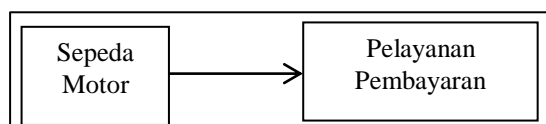
Entities merupakan proses urutan pelayanan pengisian bahan bakar pertalite yang mengikuti arus dari pelayanan pertama ke pelayanan kedua. Adapun yang menjadi entitas dalam Pengisian Peralite di SPBU Self Service Srandol Semarang ini adalah pelanggan bersepeda motor.

3.1.4 Aktivitas dan Event

Aktivitas adalah kegiatan yang dilakukan untuk memperoleh tujuan *event*. Sedangkan *event* adalah urutan kejadian suatu proses. Berikut merupakan contoh aktivitas dan event pada Pengisian Bahan Bakar Peralite di SPBU Self Service Srandol Semarang :

1) Aktivitas Pelayanan Pemesanan dan Pembayaran

Pada tahap ini dibagi menjadi setiap proses sebagai berikut :

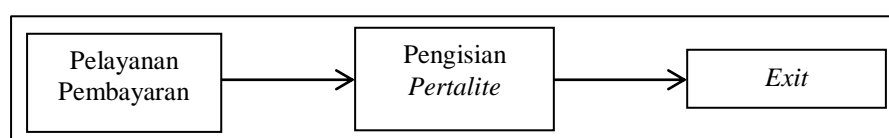


Gambar 3.2 *Layout* Pelayanan Pemesanan dan Pembayaran

Proses Awal : Kedatangan sepeda motor di pelayanan pemesanan dan pembayaran yang terdiri dari satu *server*.

Proses Akhir : Setelah dilayani di pelayanan pemesanan dan pembayaran, kemudian menuju pengisian bahan bakar pertalite dengan *self service*.

2) Aktivitas Pengisian BBM *Pertalite*



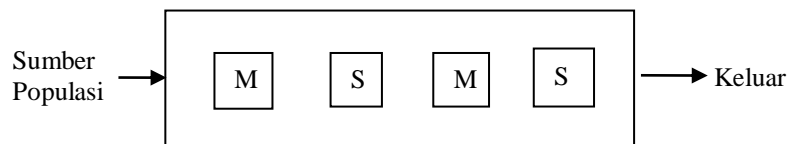
Gambar 3.3 Aktivitas Pengisian BBM

Proses Awal : Kedatangan sepeda motor di pelayanan pembayaran yang terdiri dari satu *server*.
Proses Akhir : Setelah dilayani di pelayanan pembayaran, kemudian menuju pengisian bbm dengan *self service* kemudian menuju *exit*.

3.2 Model Atrian

Terdapat empat model antrian yaitu single channel single phase, single channel multi phase, multi channel single phase, dan multi channel multi phase. Seperti pada modul 2 yang sudah dijelaskan secara rinci.

Berikut merupakan contoh model antrian yang terdapat pada Pengisian Pertalite di SPBU Self Service Sronдол Semarang, dimana menggunakan single channel multi phase :



M = Antrian

S = Fasilitas Pelayanan (*Server*)

3.3 Pengumpulan Data Pengamatan

Berikut merupakan contoh pengumpulan data pada Pengisian Pertalite di SPBU Self Service Sronдол Semarang :

Tabel 3.2 Data Pengamatan Awal

No.	Waktu Kedatangan Pengunjung	Waktu Mulai Dilayani Pada Server 1 di Phase 1	Waktu Selesai Dilayani Pada Server 1 di Phase 1	Waktu Proses (Detik)	Waktu Mulai Dilayani Pada Server 1 di Phase 2	Waktu Selesai Dilayani Pada Server 1 di Phase 2	Waktu Proses (Detik)
		Pembayaran			Pengisian Peralite		
1	17.01.27	17.06.02	17.06.17	15	17.06.34	17.06.56	22
2	17.02.29	17.06.12	17.06.29	17	17.07.19	17.07.52	33
3	17.03.20	17.06.12	17.06.45	33	17.08.14	17.08.44	30
4	17.04.27	17.08.13	17.08.48	35	17.09.02	17.09.30	28
5	17.05.01	17.09.04	17.09.16	12	17.09.53	17.10.26	33

3.4 Rekapitulasi Data Pengamatan

Terdapat rumus perhitungan yang dapat digunakan untuk melakukan rekapitulasi :

1. Waktu Antar Kedatangan = Waktu kedatangan saat ini – waktu kedatangan sebelumnya
2. Waktu Tunggu = Waktu pelayanan – waktu kedatangan
3. Waktu Pelayanan = Waktu selesai dilayani proses – waktu mulai dilayani proses
4. Waktu Dalam Sistem = Waktu antar kedatangan + waktu tunggu + waktu pelayanan

Berikut merupakan rekapitulasi data pengamatan pada Pengisian Peralite di SPBU Self Service Srandol Semarang :

Tabel 3.3 Rekapitulasi Data Pengamatan Awal

No:	Waktu Antar Kedatangan	Waktu Tunggu	Waktu Pelayanan 1	Waktu Tunggu	Waktu Pelayanan 2	Waktu Dalam Sistem
			Kasir		Pelayanan	
1	0	275	15	17	22	329
2	62	223	17	50	33	323
3	51	172	33	89	30	324
4	67	226	35	14	28	303
5	34	243	12	37	33	325

MODUL IV PROMODEL

4.1 Pengertian Promodel

Menurut Harrel (2000:66) ProModel merupakan *software* simulasi yang dirancang untuk memodelkan sistem dengan proses *discrete-event*. Dalam ProModel, terdapat *entities* (*item* yang diproses), *locations* (tempat terjadinya proses), *resources* (sumber daya yang digunakan untuk memproses dan memindahkan entitas), dan *paths* (jalan dan jalur yang dapat dilalui entitas dan *resource*).

4.2 Elemen Dasar Promodel

1. *Locations*

Location mewakili tempat pada sistem yang akan dilewati oleh entitas ataupun untuk tempat terjadinya aktivitas maupun pengambilan keputusan. Data-data yang diperlukan dalam untuk mendefinisikan *location*, adalah:

- a. *Icon* : merupakan *graphic icon* yang digunakan untuk merepresentasikan suatu *location* tertentu. Untuk merubah grafik *location* dilakukan dengan cara menggunakan *tools* pada *location graphic window*.
- b. *Name* : merupakan nama dari setiap *location* dengan panjang maksimal 80 karakter.
- c. *Capacity* : kapasitas dari suatu *location* merujuk pada jumlah entitas yang dapat ditahan atau diproses pada suatu *location* pada suatu waktu. Kapasitas maksimal *location* adalah 999999.
- d. *Unit* : jumlah dari *unit* yang ada pada suatu *location*. Jumlah maksimal *unit* pada suatu *location* adalah 999.
- e. *Downtimes* : untuk mendefinisikan *downtimes* yang terjadi pada suatu *location* termasuk waktu *setup* mesin.
- f. *Stats* : level dari detail statistik yang harus dikumpulkan untuk *location* tertentu. Terdapat tiga pilihan, yaitu *none*, *basic*, dan *time series*. *None* artinya tidak ada statistik yang dikumpulkan. *Basic* berarti hanya utilisasi dan rata – rata pada suatu *location* yang dapat dikumpulkan. *Time series* berarti mengumpulkan statistik dasar dan waktu terjadinya suatu keadaan pada suatu *location* dari waktu ke waktu.

g. *Rules* : menunjukkan bagaimana *location* dipilih dari kedatangan entitas berikutnya.

h. *Notes* : untuk menuliskan catatan apapun tentang *location*.

Contoh penggunaan *location*, misalnya untuk memodelkan *location* manufaktur, permesinan di manufaktur, dan pergudangan.

2. *Entities*

Entities adalah apapun yang diproses dalam suatu model. Data-data yang diperlukan dalam untuk mendefinisikan entitas:

a. *Icon* : merupakan *icon graphic* yang digunakan untuk mewakili suatu entitas pada animasi saat model dijalankan.

b. *Name* : merupakan nama untuk masing –masing entitas.

c. *Speed* : mendefinisikan kecepatan dari suatu entitas dalam proses, biasanya hanya digunakan untuk entitas yang dapat bergerak sendiri atau entitas manusia. Pada saat membuat *entity* baru ada nilai tetap 150 fpm yang ditetapkan oleh ProModel.

d. *Stats* : level dari detail statistik yang harus dikumpulkan untuk *location* tertentu. Terdapat tiga pilihan, yaitu *none*, *basic*, dan *time series*. *None* artinya tidak ada statistik yang dikumpulkan. *Basic* berarti hanya utilisasi dan rata – rata pada suatu *location* yang dapat dikumpulkan. *Time series* berarti mengumpulkan statistik dasar dan waktu terjadinya suatu keadaan pada suatu *location* dari waktu ke waktu.

e. *Notes* : untuk menuliskan catatan apapun tentang *location*.

Contoh *entities*, yaitu dokumen pada bank, pelanggan pada restoran, maupun barang barang pada proses manufaktur. *Entities* dapat memiliki attribut dengan sifat tertentu.

3. *Path network*

Path network adalah jalur yang dilalui oleh *resource* maupun entitas. Data-data yang diperlukan dalam untuk mendefinisikan *path network*:

- a. *Graphic* : merupakan fungsi yang menampilkan pendefinisian dari warna *path networks*.
- b. *Name* : nama yang mengidentifikasi suatu *path network*.
- c. *Type* : merupakan tipe *path network* dimana terdapat dua tipe yang dapat dipilih yaitu *passing* dan *non passing*. *Passing* dapat dilewati entitas maupun *resource*. *Non passing* tidak dapat dilewati oleh entitas maupun *resource*.
- d. T/S : merupakan fungsi waktu atau (kecepatan dan jarak) adalah dasar untuk mengukur pergerakan sepanjang *network*.
- e. *Paths* : jumlah dari *segment path* pada *network*.
- f. *Interfaces* : jumlah dari *location-interface* pada *path network* dimana entitas akan diangkat maupun diturunkan pada *location* tertentu oleh *resource*.
- g. *Mapping* : jumlah masukan dari *mapping edit table* dimana *user* dapat memetakan tujuan dari *network* tertentu.
- h. *Nodes* : merupakan titik yang dibuat secara otomatis ketika mendefinisikan *path segments*.

4. **Resource**

Resource adalah orang peralatan ataupun barang – barang yang digunakan untuk melakukan beberapa fungsi tertentu, seperti pemindahan entitas membantu pelaksanaan kinerja fungsi tertentu ataupun melakukan *maintenance* pada suatu *location*. Data-data yang diperlukan dalam untuk mendefinisikan *resource* adalah *icon*, *name*, *downtimes*, *stats*, *specs*, *search*, *logic*, *pts*, *notes*, dan *units*. Contoh *resource*, misalnya operator yang menjalankan mesin pemotong pada pabrik peleburan besi.

- a. *Icon* : merupakan *graphic icon* yang digunakan untuk merepresentasikan suatu *resource* tertentu.
- b. *Name* : merupakan nama dari *resource*.
- c. *Units* : merupakan jumlah dari *unit* yang diwakili oleh *resource* tertentu. *Resource* dapat berjumlah 0 sampai 999.
- d. *Downtimes* : untuk mendefinisikan nilai *downtimes* pada *resource* tertentu.
- e. *Stats* : level dari detail statistik yang harus dikumpulkan untuk *location* tertentu. Terdapat tiga pilihan, yaitu *none*, *summary*, dan *by units*. *None*

artinya tidak ada statistik yang dikumpulkan. *Summary* berarti rata – rata utilitas dan aktivitas sesuai waktu yang dikumpulkan untuk semua *unit resource*. *By units* berarti statistik hanya dikumpulkan untuk masing – masing individual *resource*.

- f. *Spec* : untuk menugaskan *path network* tertentu atau menentukan kecepatan *resource* dan waktu pengambilan serta waktu simpan *resource*.
- g. *Search* : digunakan untuk mendefinisikan pemilihan pekerjaan dan pemberhentian *resource* maupun digunakan untuk pencarian *path network* tertentu yang telah digunakan untuk *resource*.
- h. *Logic*: untuk mendefinisikan jumlah nodes yang telah didefinisikan pada *path network* tertentu maupun *location* masuk dan *location* keluar nodes tertentu.
- i. *Pts*: digunakan untuk mendefinisikan posisi *resource* pada *path network*.
- j. *Notes* : untuk menuliskan catatan apapun tentang *resource*.

5. ***Processing***

Processing mendefinisikan *routing* dari entitas yang melalui sistem dan operasi yang terjadi pada setiap *location* yang dimasuki entitas. Dalam mendefinisikan *processing*, terdapat empat *editing windows* yaitu *process edit table*, *routing edit table*, *tools windows*, dan *layout window*. *Process edit table* digunakan untuk mendefinisikan logika operasi pada setiap tipe *entity* dalam setiap *location* yang terdapat di dalam sistem. Untuk mempermudah pembuatan model sebaiknya logika proses diurutkan sesuai dengan entitas atau *location*. *Routing edit table* mendefinisikan *output* dari setiap proses yang didefinisikan oleh proses *edit table*, walaupun tidak semua proses *record* memerlukan *routing*. Ketika pada *process edit table* ada penentuan perpindahan entitas ke *location* tertentu, maka *routing edit table* akan mencari entitas yang berkaitan pada *location* yang dituju. Ketika telah didefinisikan pada proses *edit table* ke *location* tujuan tidak ditemukan *routing edit table* atau pendefinisian entitas pada *location* tersebut maka akan terjadi *error*. Pada *processing* juga terdapat fungsi prioritas yang dapat digunakan ketika terdapat satu *downstream location* yang tersedia dan dua atau lebih entitas dari *upstream* berkompetisi untuk masuk ke *downstream*.

6. ***Arrivals***

Arrival mendefinisikan waktu dimana entitas masuk pada sistem. Data-data yang diperlukan dalam *dialog box* untuk mendefinisikan *arrival* :

- a. *Entity* : nama dari entitas yang datang.
- b. *Location* : nama *location* dimana entitas akan datang.
- c. *Quantity each* : jumlah dari entitas yang harus datang pada suatu jarak waktu tertentu. Harus diisi dengan bilangan tertentu antara 1 sampai 999999 kecuali untuk atribut dan fungsi sistem *non general*.
- d. *First time* : untuk memvariasikan secara dinamis waktu kedatangan pertama dari entitas pada model yang telah dibuat, serta dapat dibuat penjadwalan kedatangan pada interval tertentu.
- e. *Occurences* : jumlah waktu persimulasi dimana ProModel harus memunculkan arrival (1-999999). Ketika arrival dibentuk dalam suatu siklus maka jumlah *occurences* merupakan jumlah waktu perulangan siklus.
- f. *Frequency* : waktu antar kedatangan, dimana nilai berapapun dapat dimasukkan kecuali untuk atribut dan fungsi sistem *non general*. Fungsi *frequency* akan dievaluasi selama simulasi dijalankan dan akan berubah ketika hasil dari simulasi berubah.
- g. *Logic* : untuk mendefinisikan logika pilihan kedatangan tertentu, terdiri dari satu atau lebih pernyataan tertentu yang akan dieksekusi pada saat entitas datang pada kedatangan tertentu.
- h. *Disable*: dapat diganti menjadi pilihan *yes* atau *no* ketika *user* ingin menonaktifkan *arrival* secara temporer tanpa harus menghapus *arrival* tersebut.

4.3 Fitting Data Into Distribution

3.3.1 Fitting Data Dengan Stat Fit

Setelah melakukan pengujian kecukupan data dan data dianggap cukup, maka data-data tersebut kemudian dibuat distribusinya menggunakan *tools*

Stat:Fit yang terdapat pada *software* ProModel dengan langkah-langkah sebagai berikut.

- a. Menjalankan *software* ProModel
- b. Pilih *Tools* pada *toolbar*, pilih Stat::Fit
- c. Masukkan data pengamatan yang telah dilakukan pada data tabel.
- d. Klik Fit, kemudian Auto::Fit, pilih *Continuous Distribution* lalu *Unbounded* kemudian OK.
- e. Hasil akan ditampilkan berupa *automatic filling*. Untuk penggunaan distribusi pada simulasi, pilih distribusi *Acceptance: do not reject* dengan rank tertinggi.

3.3.2 Fitting Data Secara Manual

1. Melakukan Pendugaan Distribusi

Untuk mengetahui apakah data pengamatan mengikuti pola data tertentu maka perlu diketahui nilai *index of fit* (r) dari masing – masing distribusi normal, lognormal, eksponensial, weibull. *Index of fit* yang terpilih adalah dengan nilai terbesar. *Index of fit* didapatkan dengan rumus sebagai berikut :

$$F(t_i) = \frac{i - 0,3}{n + 0,4}$$

Dimana : i = Data waktu ke- t

$n = r$ = Jumlah data yang terjadi untuk data lengkap

$n = N$ = Jumlah data yang diamati untuk data sensor

Perhitungan untuk tiap distribusi adalah :

- Distribusi Eksponensial

$$r_{eksponensial} = \frac{n \sum_{i=1}^n x_i y_i - (\sum_{i=1}^n x_i)(\sum_{i=1}^n y_i)}{\sqrt{[n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2][n \sum_{i=1}^n y_i^2 - (\sum_{i=1}^n y_i)^2]}}$$

Dimana :

$$x_i = t_i$$

$$y_i = \ln\left(\frac{1}{1 - F(t_i)}\right)$$

- Distribusi Weibull

$$r_{weibull} = \frac{n \sum_{i=1}^n x_i y_i - (\sum_{i=1}^n x_i)(\sum_{i=1}^n y_i)}{\sqrt{[n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2][n \sum_{i=1}^n y_i^2 - (\sum_{i=1}^n y_i)^2]}}$$

Dimana :

$$x_i = \ln t_i$$

$$y_i = \ln \left[\ln \left(\frac{1}{1 - F(t_i)} \right) \right]$$

- Distribusi Normal

$$r_{normal} = \frac{n \sum_{i=1}^n x_i y_i - (\sum_{i=1}^n x_i)(\sum_{i=1}^n y_i)}{\sqrt{[n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2][n \sum_{i=1}^n y_i^2 - (\sum_{i=1}^n y_i)^2]}}$$

Dimana :

$$x_i = t_i$$

$$y_i = z = \Phi^{-1}[F(t_i)] = \frac{t_i - \mu}{\sigma} \rightarrow \text{diperoleh dari tabel } Standardized Normal$$

Probabilities ($\Phi(z)$)

- Distribusi Lognormal

$$r_{lognormal} = \frac{n \sum_{i=1}^n x_i y_i - (\sum_{i=1}^n x_i)(\sum_{i=1}^n y_i)}{\sqrt{[n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2][n \sum_{i=1}^n y_i^2 - (\sum_{i=1}^n y_i)^2]}}$$

Dimana :

$$x_i = \ln(t_i)$$

$$y_i = z = \Phi^{-1}[F(t_i)] = \frac{t_i}{s} \ln t \rightarrow \text{diperoleh dari tabel } Standardized Normal$$

Probabilities ($\Phi(z)$)

Kemudian pilih dengan nilai r terbesar.

2. Uji Kecocokan Distribusi (*Goodness of Fit Test*)

Setelah diketahui *index of fit* dan diketahui distribusi yang digunakan, langkah selanjutnya adalah melakukan uji kecocokan distribusi dikarenakan nilai r pada *index of fit* hanya merupakan suatu nilai yang menunjukkan bahwa suatu data mengikuti distribusi tertentu. Untuk itu perlu dilakukan uji *goodness of fit test* untuk mengetahui secara signifikan bahwa data waktu kerusakan mengikuti suatu distribusi yang tepat. Uji kecocokan distribusi ini dilakukan dengan cara manual dan bantuan *software Minitab*.

- **Uji Bartlett untuk distribusi Eksponensial**

Hipotesis yang digunakan untuk uji ini adalah :

H_0 : Data berdistribusi Eksponensial

H_1 : Data tidak berdistribusi Eksponensial

Uji statistiknya :

$$B = \frac{2r \left[\ln \left(\frac{\sum_{i=1}^r t_i}{r} \right) - \left(\frac{\sum_{i=1}^r \ln t_i}{r} \right) \right]}{1 + \frac{(r+1)}{6r}}$$

Dimana : r = jumlah kedatangan

t_i = data kedatangan ke-i

B = nilai uji statistik untuk *Bartlett's Test*

H_0 diterima apabila nilai B jatuh dalam wilayah kritis :

$$X_{1-\frac{\alpha}{2}, r-1}^2 < B < X_{\frac{\alpha}{2}, r-1}^2$$

- **Uji Mann untuk distribusi Weibull**

Hipotesis yang digunakan untuk uji ini adalah :

H_0 : Data berdistribusi Weibull

H_1 : Data tidak berdistribusi Weibull

Uji statistiknya :

$$M = \frac{k_1 \sum_{i=k_1+1}^{r-1} \left(\frac{\ln t_{i+1} - \ln t_i}{M_i} \right)}{k_2 \sum_{i=1}^{k_1} \left(\frac{\ln t_{i+1} - \ln t_i}{M_i} \right)}$$

$$k_1 = \frac{r}{2} \text{ dan } k_2 = \frac{r-1}{2}$$

Dimana : $M_i = Z_{i+1} - Z_i$

$$Z_i = \ln \left[-\ln \left(1 - \frac{i - 0.5}{n + 0.25} \right) \right]$$

Keterangan : M = nilai uji statistik untuk *Mann's Test*

t_i = data waktu kedatangan ke-i

t_{i+1} = data waktu kedatangan ke-(i+1)

r = n : jumlah unit yang diamati

M_i = nilai pendekatan *Mann* untuk data ke-i

$M_{\alpha, K1, k2}$ = nilai M_{tabel} untuk distribusi weibull → lihat

distribusi F, dengan $v_1=k_1$ dan $v_2=k_2$

H_0 diterima jika M_{hitung} jatuh pada wilayah kritis : $M_{\text{hitung}} <$

$M_{\text{tabel}(\alpha, K1, k2)}$

- **Uji Kolmogorov-Smirnov untuk distribusi Normal dan Lognormal**

Hipotesis yang digunakan untuk uji ini adalah :

H_0 : Data berdistribusi Normal/Lognormal

H_1 : Data tidak berdistribusi Normal/Lognormal

Uji statistiknya : $D_n = \max \{D_1, D_2\}$

$$\text{Dimana : } D_1 = \max_{1 \leq i \leq n} \left\{ \Phi \left(\frac{t_i - \bar{t}}{s} \right) - \frac{i-1}{n} \right\}$$

$$D_2 = \max_{1 \leq i \leq n} \left\{ \frac{i}{n} - \Phi \left(\frac{t_i - \bar{t}}{s} \right) \right\}$$

$$\bar{t} = \sum_{i=1}^n \frac{\ln t_i}{n} \text{ dan } s^2 = \frac{\sum_{i=1}^n \ln(t_i - \bar{t})^2}{n-1}$$

Keterangan : t_i = data waktu antar kedatangan ke- i

\bar{t} = rata – rata data waktu antar kedatangan

s = standar deviasi

n = banyaknya data kerusakan

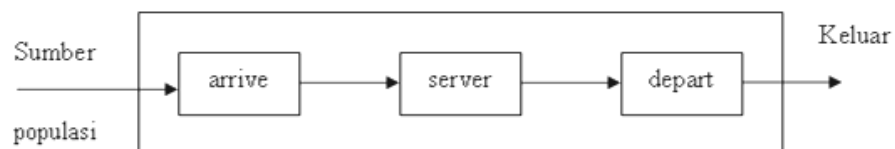
Bila $D_n < D_{\text{crit}}$ maka terima H_0 , dan bila sebaliknya maka terima H_1 .

Nilai D_{crit} diperoleh dari tabel *critical value for thr Kolmogorov-Smirnov test for normality*.

4.4 Pembuatan Model

Model merupakan bentuk sederhana dari sebuah sistem. Sistem disini bisa berupa sistem dalam pelayanan umum atau proses manufaktur. Untuk bisa mensimulasikan dalam program promodel maka kita membuat model yang merupakan gambaran sistem tersebut, secara sederhana model dapat diilustrasikan terdiri dari 3 komponen yaitu kedatangan (*arrive*), pelayanan (*service*), dan keluaran (*depart*).

Sistem Antrian



Gambar 4.1. Model Sistem Antrian

Konsumen datang kemudian dilayani dan keluar, pelayanan dalam hal ini dapat dimisalkan berupa pembelian tiket kereta, pembayaran di bank, pembelian barang

di supermarket, dan lain-lain. Server juga dapat berupa mesin yang akan mengerjakan produk seperti mesin bubut, drill, bor, dan lain-lain.

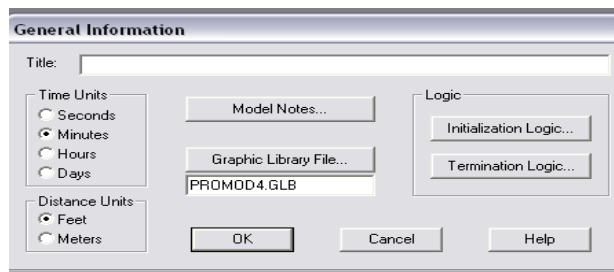
Berikut adalah bagian-bagian pembuatan model :

1. Membuka Model



Gambar 4.2 Tampilan Promodel

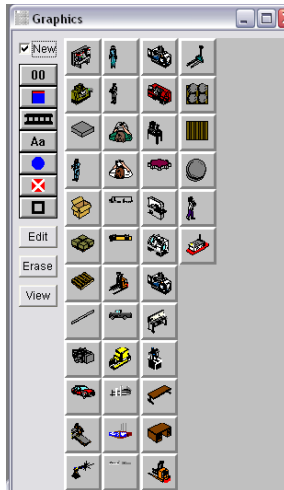
2. Judul Model



Gambar 4.3 Judul Model

3. Location

Locations							
Icon	Name	Cap.	Units	Dts...	Stars...	Rules...	Notes...



Gambar 4.4 Location

4. Entitas

Icon	Name	Speed (fps)	Stats...	Notes...



Gambar 4.5 Entitas

5. Path

Graphic	Name	Type	T/S	Paths...	Interfaces...	Mapping...	Nodes
	Net1	Passing	Speed & Distance	0	0	0	0

From	To	BI	Distance

Gambar 4.6 Path

6. Resource

Icon	Name	Units	Dts...	Stats...	Specs...	Search...	Logic...	Pcs...	Notes...

Resource Graphics

New

Edit Erase

Layout Position:

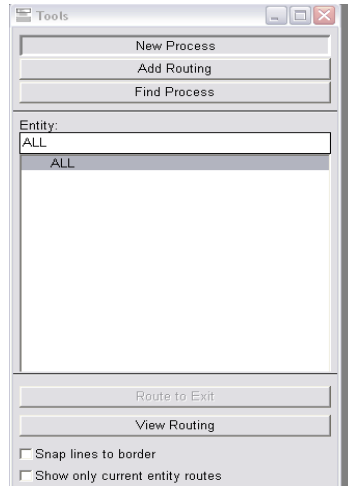
Add Delete

Gambar 4.7 Resource

7. Routing & Process

Entity...	Location...	Operation...

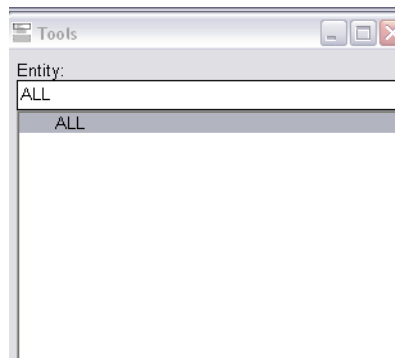
Blk	Output...	Destination...	Rule...	Move Logic...



Gambar 4.8 Routing dan Process

8. Arrivals

Entity...	Location...	Qty each...	First Time	Occurrences	Frequency	Logic	Disable



Gambar 4.9 Arrivals

4.5 Study Kasus Software Promodel

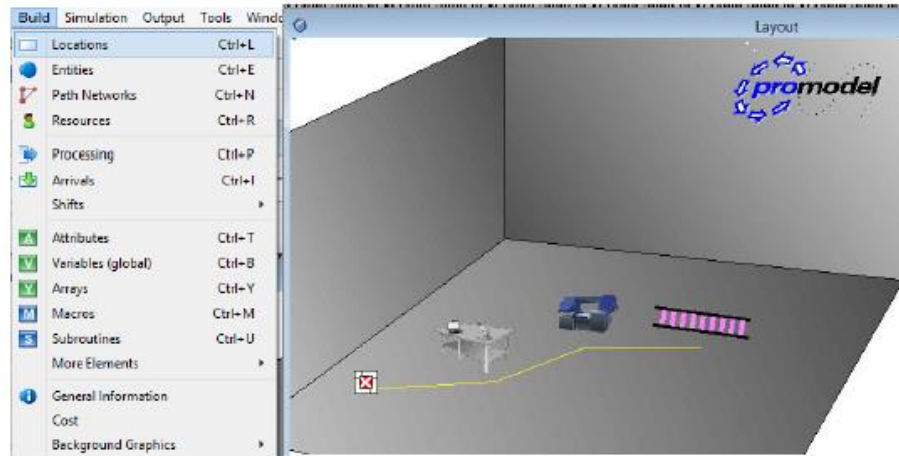
Studi kasus GROUP/UNGROUP

Perusahaan LID sedang melakukan proses pengisian botol kecap. botol akan datang setiap 1 menit sebanyak 20 botol dengan kedatangan 25 kali, botol kecap tersebut akan masuk pada mesin pengisian sebanyak 40 botol yang sebelumnya dikumpulkan terlebih dahulu di pengumpulan kecap dan akan di isi secara bersama-sama. Waktu pengumpulan botol berdistribusi normal (5,1) menit. Setelah melalui proses pengisian selama 15 menit, botol kecap akan satu persatu

dimasukan ke dalam kardus. Buatlah simulasi selama 8 jam menggunakan *software* ProModel.

Berikut adalah langkah-langkah pengerjaan pembuatan model :

1. Buka *software* ProModel 2014.
2. Klik *build location* dan buat lokasi-lokasi seperti gambar berikut



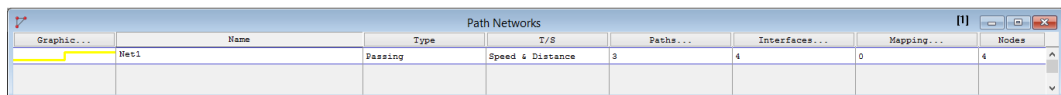
Gambar 4.10 *Build Location*

Berikut ini merupakan tabel *location*

<i>Name</i>	<i>Capacity</i>	<i>Unit</i>
Kedatangan botol kecap	INFINITE	1
Pengumpulan kecap	50	1
Pack kecap	40	1
Keluar	INFINITE	1

Gambar 4.11 *Location*

3. Buat *path network*, dengan cara klik *build -> path network*
4. Buat 1 *path* yaitu Net1



Gambar 4.12 *Path*

NET 1 : tarik garis dari kedatangan botol kecap ke pengumpulan kecap (dengan cara klik kiri dari kedatangan botol kecap, tarik ke pengumpulan kecap kemudian klik kanan untuk berhenti) lalu tarik dari pengumpulan kecap ke *pack* kecap dan klik kanan, kemudian tarik dari *pack* kecap ke luar. Untuk mendefinisikan *interface* , klik kolom *interface* , kemudian tarik garis dari setiap titik ke lokasi yang dituju, nanti akan muncul garis putus-putus.

5. *Build entities* dengan klik *build -> entities*

Icon	Name	Speed (fsm)	Stats	Notes...
	botol_kecap	150	Time Series	
	batch_botol_kecap	150	Time Series	

Name	Speed	Stats
Botol kecap	150	Time series
Kecap isi	150	Time series
Batch botol kecap	150	Time series

Gambar 4.13 Build Entities

6. Build resources dengan klik build -> resources

Icon	Name	Units	Specs	Search...	Logic...	Pos...	Notes...
	operator_pengumpulan	1	None	By Unit, Time Ser; Net1, N3, Rtn Home	None	0	1
	mesin_pengisian	1	None	By Unit, Time Ser; Net1, N2	None	0	1

Name	Units	Specs
Operator pengumpulan	1	By Units, N3, Rtn Home
Mesin pengisian	1	By Units, N2, Rtn Home

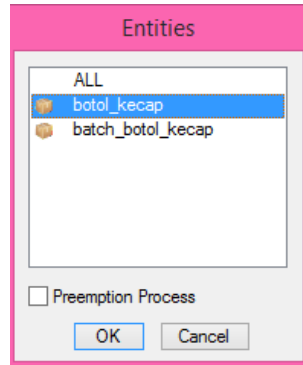
Gambar 4.14 Build Resources

Untuk kolom *specs* operator pengumpulan, pilih *path network* Net1 dan *home* N2, dan centang *return home if idle*. Untuk kolom *specs* mesin pengisian, pilih *path network* Net1 dan *home* N3, dan centang *return home if idle*.

7. Build process dengan isi tabel proses dan routing sebagai berikut

Entites	Process		Routing			
	Location	Operation	Output	Destination	Rule	Move logic
Botol_kecap	Kedatangan_botol_kecap		Botol_kecap	Pengumpulan_kecap	FIRST 1	
Botol_kecap	Pengumpulan_botol_kecap	Group 40 as batch_botol_kecap				
Batch_botol_kecap	Pengumpulan_botol_kecap	Use operator_pengumpulan for N(5,1) min	Batch_botol_kecap	Pack_kecap	FIRST 1	Move with operator_pengumpulan then free
Batch_botol_kecap	Pack_kecap	Use mesin_pengisian for 15 min				
Batch_botol_kecap	Pack_kecap	Ungroup				
Botol_kecap	Pack_kecap		kecap_isi	keluar		Move with operator_pengumpulan then free
kecap_isi	keluar		kecap_isi	EXIT	FIRST 1	

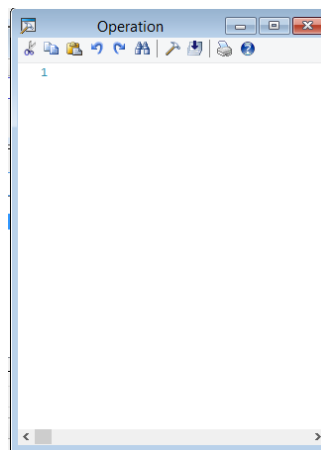
Untuk mengisi tabel *entities*, klik tulisan *entities* di kolom *entities*, kemudian pilih entitas yang telah didefinisikan di *build entities* sebelumnya




Gambar 4.15 *Entities*

Untuk tabel *locations*, caranya sama seperti pada pengisian tabel *entities*. Untuk *operation*, dapat dilakukan dengan langkah berikut ini.

- a. Klik tulisan *operation* pada kolom *operation*, kemudian akan tampil gambar berikut



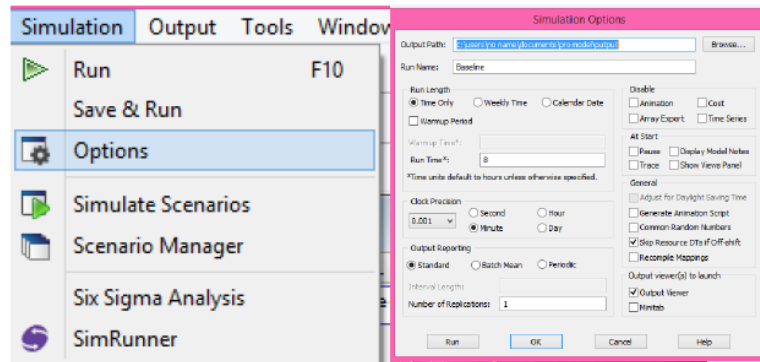
Gambar 4.16 *Operation*

- b. Klik tombol palu  untuk memilih perintah yang akan digunakan
8. *Build arrival* dengan cara klik *arrival* pada *menu bar build*

Entity...	Location...	Qty Each...	First Time...	Occurrences	Frequency	Logic...	Disable
botol_kecap	kedatangan_botol_kecap	20		25	1 min		No

Gambar 4.17 *Arrival*

9. *Setting* simulasi dengan cara sebagai berikut



Gambar 4.18 Setting

10. Jalankan simulasi dengan klik *Run*

MODUL V

EXTEND

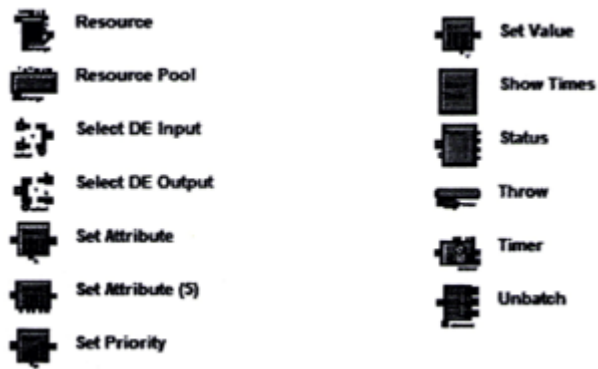
5.1 *Extend*

Extend (ImagineThat,Inc.) digunakan untuk memodelkan, menganalisis dan mengoptimalkan proses. Memiliki sejumlah fitur seperti kumpulan komponen, hirarki model, *link* dengan MSOffice dan memodelkan sistem *kontinyu*, *diskrit* dan *hybrid*. *Extend* mempunyai bahasa pemodelan sendiri (ModL) yang mirip dengan C, dan mampu memanggil kode dari bahasa lainnya. Mempunyai paket khusus untuk sistem industri, riset operasional dan simulasi proses *kontinyu*.

5.2 Bagian – Bagian *Extend*

5.2.1 DE.LIX





Gambar 4.1 *De.Lix*

5.2.2 *GENERIC.LIX*



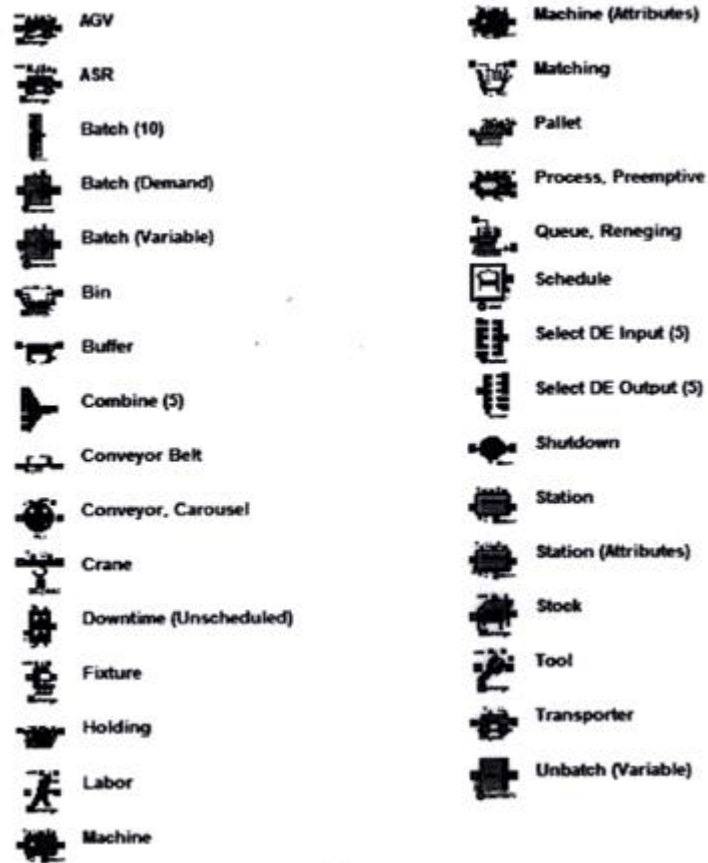
Gambar 4.2 *Generic.Lix*

5.2.3 *STATS.LIX*



Gambar 4.3 *Stats.Lix*

5.2.4 *MFG.LIX*



Gambar 4.4 *MFG.Lix*

MODUL VI

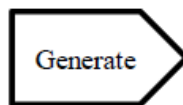
SOFTWARE ARENA

6.1 *Activity Cycle Diagram*

Activity cycle diagram (ACD) adalah bahasa grafik/gambar yang memodelkan sistem dengan menunjukkan hubungan interaksi antar elemen dengan perubahan secara diskrit terhadap waktu. Entitas di ACD ada dua, yaitu permanen dan sementara. Sedangkan aktivitas pada ACD ada dua, yaitu pasif dan aktif. Simbol-simbol yang dipergunakan pada ACD adalah :

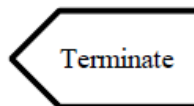
1. Segilima ke kanan

Merepresentasikan menciptakan (*create*) atau membangkitkan (*generate*) entitas.



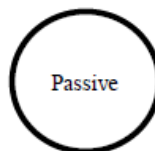
2. Segilima ke kiri

Merepresentasikan membuang (*dispose*) atau memberhentikan (*terminate*) entitas.



3. Lingkaran (*passive state*)

Merepresentasikan aktivitas pasif.



4. Segi empat (*active state*)

Merepresentasikan aktivitas aktif



5. Panah (*connect*)

Merepresentasikan relasi urutan antar node yang menunjukkan bahwa status/aktivitas pendahulu berubah/berlanjut menjadi status/aktivitas berikutnya



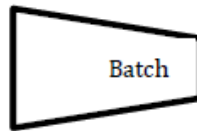
6. Belah ketupat (*alternate*)

Merepresentasikan kondisi (*condition*) pilihan dua alternatif kemungkinan yang perlu diputuskan (*decide*)



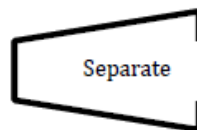
7. Trapesium kanan (*assembly/batch*)

Merepresentasikan aktivitas aktif yang melibatkan dua entitas (atau lebih) dan bertransformasi menjadi satu entitas (lain)

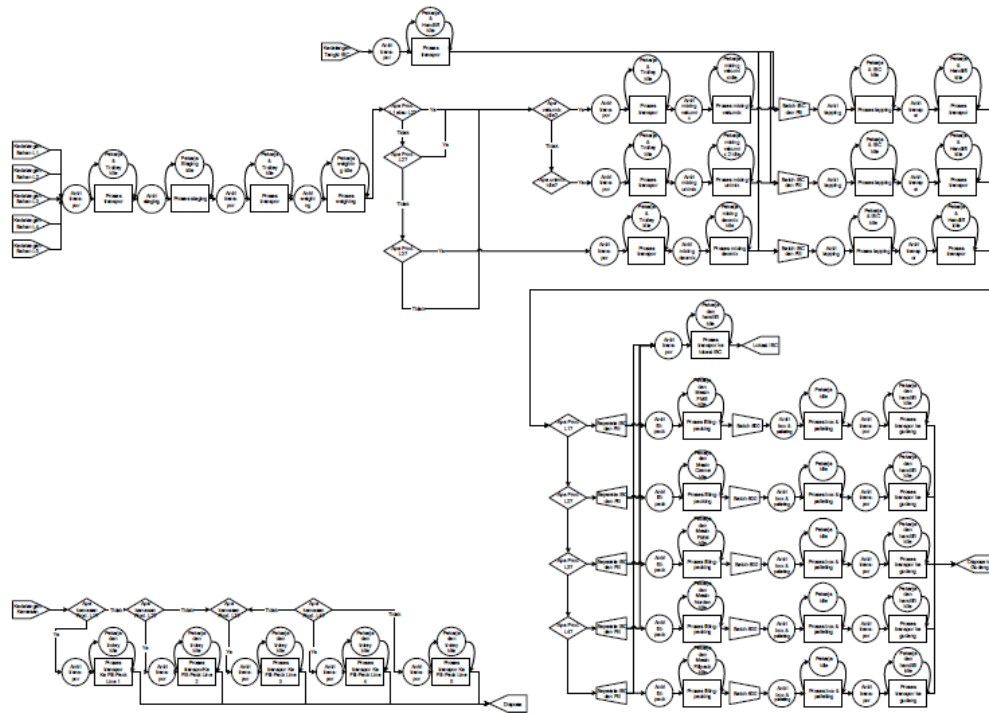


8. Trapesium kiri (*disperse/separate*)

Merepresentasikan aktivitas aktif yang mentransformasikan satu entitas menjadi dua entitas (atau lebih)



Berikut merupakan contoh ACD:



Gambar 5.1 Contoh ACD

Sumber: Putu Ambarisha K, 2015

6.2 Definisi ARENA

Arena adalah sebuah program penyusun model dan juga merupakan simulator. Arena merupakan kombinasi antara kemudahan pemakaian yang dimiliki *high level program* dan fleksibilitas/kelenturan yang menjadi ciri *general purpose simulation language (GPSL)*.

Arena masuk dalam kategori *high level program* karena ia bersifat sangat interaktif. Pengguna dapat membangun sebuah model hampir sama mudahnya dengan membuat poster dengan menggunakan Corel Draw atau membangun *flowchart* dengan Visio. Hal yang membedakan hanyalah, dalam Arena dibutuhkan pengetahuan mengenai sistem yang akan diamati sebelum memodelkannya.

Arena juga termasuk *software* simulasi yang memiliki ciri *general purpose simulation language*, dimana pengguna dapat membangun model, *templates*, bahkan membuat sendiri modul jika diperlukan dengan menggunakan bantuan program seperti Visual Basic, FORTRAN, dan C/C++. Dalam profesional *edition*,

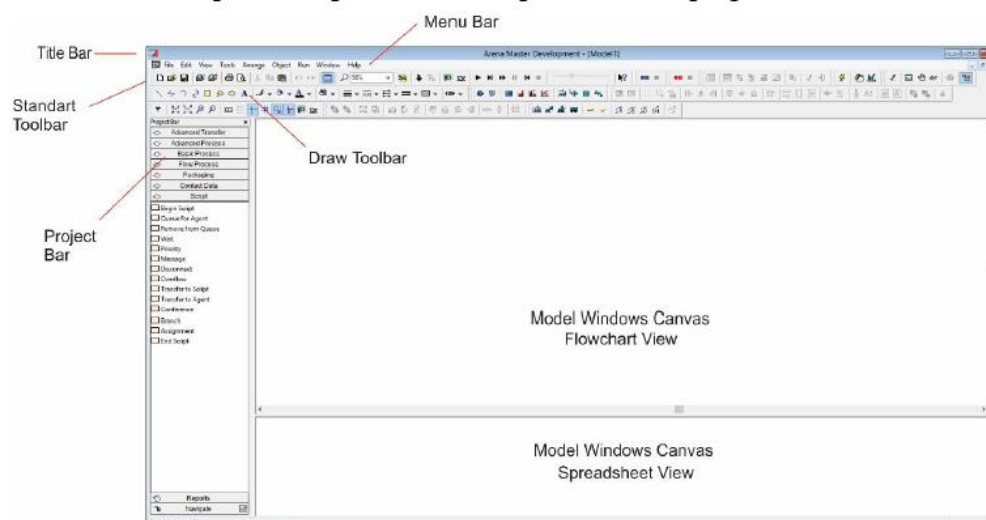
Arena memfasilitasi pengguna yang ingin membangun sendiri modul dan *templatnya*.

Orientasi dari Arena adalah memodelkan sistem dengan *process orientation* dan memberikan informasi mengenai kejadian dalam sistem secara *event orientation*. Dengan menggunakan Arena, pengguna mendapatkan :

1. Kemudahan dalam penggunaan terutama dalam pemodelan sistem dan analisa hasil simulasi dengan *interface*-nya yang sangat interaktif
2. Fleksibilitas yang sangat besar dalam membangun model yang sesuai dengan sistem sesungguhnya dengan menggunakan modul dan blok yang beragam
3. Kemudahan dalam memodelkan dan mensimulasikan sistem manufaktur seperti *material handling, inventory, quality control, dan bottleneck analysis*, maupun industri jasa seperti perbankan, rumah sakit, dan *order fulfillment*.

6.3 ARENA Home Screen

Berikut merupakan tampilan dari Arena pada saat awal program Arena dibuka.



Gambar 5.2 Arena Home Screen

Tampilan diatas adalah tampilan standar dari *home screen* Arena. Berikut adalah *toolbars* yang ada di Arena:

5. *Title Bar* menunjukkan nama dari model yang dibuat.
6. *Menu Bar* yang terdiri dari menu yang umum dan menu spesifik dari Arena.
7. *Project Bar* digunakan untuk memilih modul yang digunakan untuk membangun program menggunakan.
8. *Flowchart View* untuk membangun model di area.
9. *Spreadsheet view* untuk mengedit data dari modul-modul yang digunakan bisa diedit menggunakan.

6.4 Pengenalan Software ARENA

STUDY CASE 1

Suatu panggilan telepon datang mengikuti distribusi exponential setiap 20 menit. Lalu akan dilakukan proses sortir panggilan oleh seorang *customer service* 1 selama 1 menit. Probabilitas panggilan dilanjutkan kepada *customer service* 2 adalah 80%. Jika panggilan telepon tidak dilanjutkan, maka panggilan akan diputus. Jika disambungkan maka panggilan akan disambungkan kepada *customer service*. Sambungan panggilan tersebut dilakukan secara bergantian oleh 2 orang operator panggilan dengan lama proses berdistribusi normal yang berata-rata 1 menit dan *standard deviasi* sebesar 0,2. Buatlah sistem simulasi tersebut!

Langkah-langkah pembuatan:

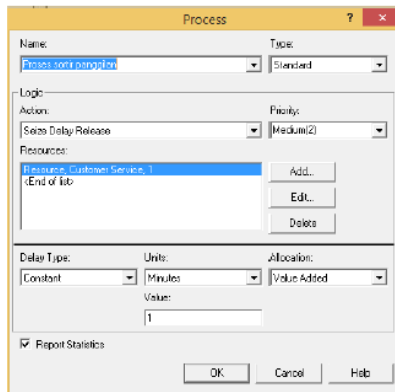
1. Untuk memulai nya, pertama buka program Arena 14.
2. *Drag and Drop Modul Create* di Model Window Canvas.
3. *Double click* pada modul *create* yang sudah dibuat, lalu pengisiannya :

a. Name	Panggilan datang
b. Entity Type	call
c. Type	Random (Expo)
d. Value	5
e. Units	Minutes
f. Entities per Arrival	1
g. Max Arrivals	Infinite
h. First Creation	0.0

Gambar 5.3 Create Module

4. *Drag and Drop* modul Proses di Model Window Canvas

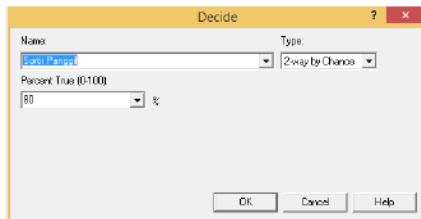
5. *Double click* pada modul proses yang sudah dibuat, lalu pengisiannya.



a. Name	Proses sortir panggilan
b. Type	Standard
c. Action	Seiza Delay Release
d. Priority	Medium
e. Resources	Klik Add, kemudian: Type: Resources Resource Name: Customer Service Quantity: 1
f. Delay Type	Constant
g. Units	Minutes
h. Allocation	Value Added
i. Value	1

Gambar 5.4 Process Module

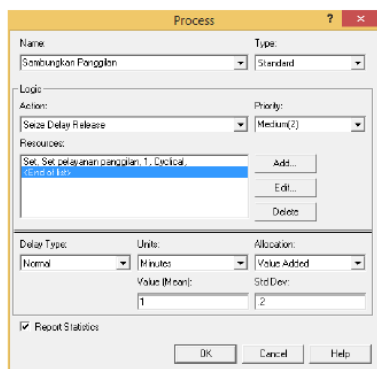
6. Drag and Drop modul Decide di Modul Window Canvas
7. *Double click* pada modul *Decide* yang sudah dibuat, lalu pengisiannya:



a. Name	Sortir panggil
b. Type	2-way by Chance
c. Percent True	80%

Gambar 5.5 Decide Module

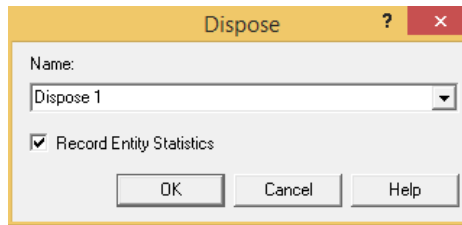
8. Drag and Drop modul Proses di Model Window Canvas
9. *Double click* pada modul proses yang sudah dibuat, lalu pengisiannya:



a. Name	Sambungkan panggilan
b. Type	Standard
c. Action	Seiza Delay Release
d. Priority	Medium
e. Resources	Klik Add, kemudian: Type: Set Set Name: Set pelayanan panggilan Quantity: 1 Selection Rule: Cyclical Save Attribute:
f. Delay Type	Normal
g. Units	Minutes
h. Allocation	Value Added
i. Value	1
j. Std Dev	0.2

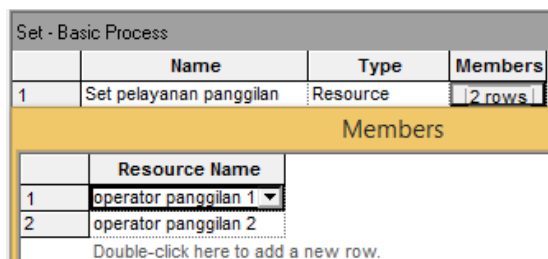
Gambar 5.6 Process Module

10. Drag and Drop modul *Dispose* di Model Window Canvas Name: *Dispose 1*



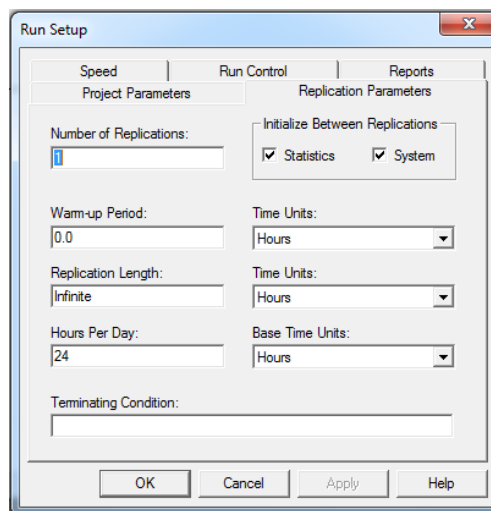
Gambar 5.7 *Dispose Module*

11. *Drag and Drop* modul *Dispose* di Model Window Canvas Name: *Dispose 2*
12. Definisikan *set resource* diatas. Klik pada modul *resource* di *basic process*. Klik 0 rows pada *members* di *set* pelayanan panggilan. Isi *resource name* dengan nama operator panggilan 1 dan operator panggilan 2.



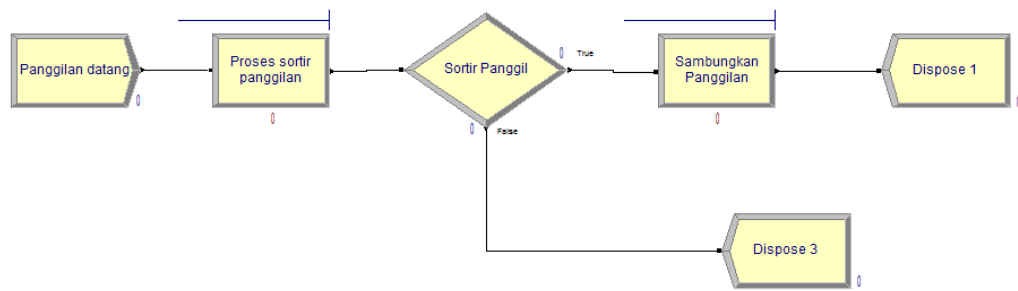
Gambar 5.8 *Set Basic Process*

13. Klik *Run – Run Setup*:
 - a. Klik *Replication Parameters*
 - b. *Number replications*: 5
 - c. *Replication Length*: 8
 - d. Klik OK



Gambar 5.9 *Run Setup*

14. Klik *Icon Play* pada *toolbars*, maka simulasi akan berjalan



Gambar 5.10 Hasil Simulasi

MODUL VII

REPLIKASI, VERIFIKASI, DAN VALIDASI

7.1 Perhitungan Replikasi

Jumlah replikasi yang diperlukan dalam simulasi biasanya bergantung pada *half-width* yang diharapkan pada interval kepercayaan tertentu. *Half-width* menunjukkan ketidakpastian dari hasil replikasi (Stokes, 2004).

$$n' = \left[\frac{(z_{\alpha/2})s}{e} \right]^2$$

Di mana:

n' = jumlah replikasi

s = standar deviasi dari sampel yang diambil

e = *absolute error*

$$hw = \frac{(t_{n-1, \alpha/2})s}{\sqrt{n}}$$

Di mana:

hw = *half-width*

n = jumlah replikasi

s = standar deviasi dari sampel yang diambil

7.2 Verifikasi

Verifikasi dan validasi merupakan tahapan untuk menguji kredibilitas atau kesesuaian *system* nyata dengan model simulasi. Verifikasi adalah proses untuk menentukan apakah model telah beroperasi sesuai yang diinginkan oleh *programmer*. Verifikasi berkaitan dengan melakukan perbandingan antara model konseptual dengan model simulasi (Banks, Carson, dan Nelson, 1995). Verifikasi adalah proses pemeriksaan logika operasional model (program komputer) sesuai dengan logika diagram alur (Hoover dan Perry, 1989).

7.2.1 Teknik Verifikasi

Menurut Harrel (2004:178), terdapat beberapa teknik dalam melakukan verifikasi, diantaranya adalah sebagai berikut :

1. Melakukan pemeriksaan ulang terhadap model, dapat dilakukan secara *bottom-up* yaitu melakukan pemeriksaan satuan dan logika proses yang digunakan dalam model.
2. Melakukan pengecekan terhadap *output* yang dihasilkan pada masing-masing proses pada model dengan menggunakan *trace*. *Trace* adalah daftar kejadian yang akan terjadi sampai simulasi selesai. Daftar *trace* dapat dilihat dalam berbagai cara, yaitu:
 - a. *Off*: digunakan untuk menghentikan *trace*.
 - b. *Step*: digunakan untuk membuat *list trace* dengan hanya satu kejadian dalam 1 kali *trace*.
 - c. *Continuous*: digunakan untuk membuat *list trace* terus menerus.
3. Mengamati animasi dari model yang dijalankan, apakah tingkah laku dari sistem telah sesuai dengan model yang diinginkan.
4. Melakukan *compile error* atau *debugging* pada model simulasi.

7.3 Validasi

Verifikasi dan validasi merupakan tahapan untuk menguji kredibilitas atau kesesuaian *system* nyata dengan model simulasi. Validasi adalah proses penentuan apakah model merupakan representasi yang akurat dan sesuai dengan sistem nyata (Hoover dan Perry, 1989).

7.3.1 Validasi

Menurut Harrel (2004:183), teknik validasi yang dapat digunakan adalah sebagai berikut:

1. Mengamati animasi pada model yang dijalankan, membandingkan tingkah laku pada model dengan tingkah laku pada sistem nyata menurut pengetahuan orang lain mengenai sistem tersebut.
2. Membandingkan model dengan sistem nyata dengan cara menjalankan model dan sistem nyata dalam kondisi yang sama.

3. Melakukan perbandingan antara *output* model dengan *output* pada sistem nyata.
4. Melakukan analisis sensitivitas, yakni dengan cara melakukan perubahan terhadap nilai *input* untuk mengetahui akibat pada perilaku yang terjadi pada sistem atau pada *output* sistem.

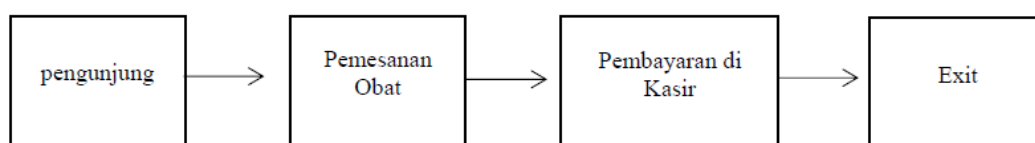
MODUL VIII
EVALUASI PERFORMANSI DAN RANCANGAN MODEL ALTERNATIF
(PROMODEL)

a. Pendahuluan

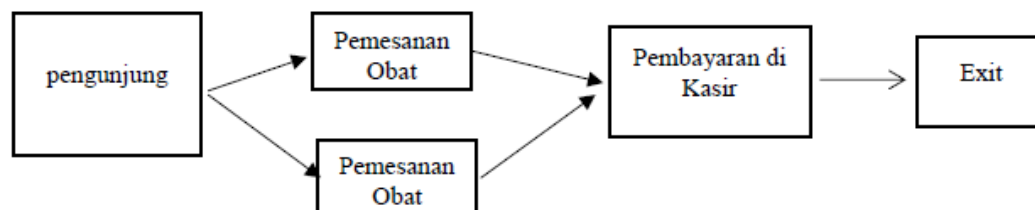
Pada modul ini akan dilakukan evaluasi performansi sistem antrian dengan menggunakan *software* promodel. Evaluasi performansi sistem antrian dilakukan pada server yang memiliki tingkat utilitas (tingkat kesibukan) tertinggi, ataupun server yang memiliki tingkat utilitas (tingkat kesibukan) terendah. Apabila sebuah server memiliki tingkat utilitas (tingkat kesibukan) tertinggi maka akan dilakukan evaluasi performansi dengan cara melakukan penambahan jumlah server pada server dengan utilitas tertinggi. Apabila sebuah server memiliki tingkat utilitas (tingkat kesibukan) rendah maka akan dilakukan evaluasi performansi dengan cara melakukan pengabungan jumlah server pada server dengan tingkat utilitas rendah.

b. Gambaran Evaluasi Sistem *Manufacture* atau Jasa

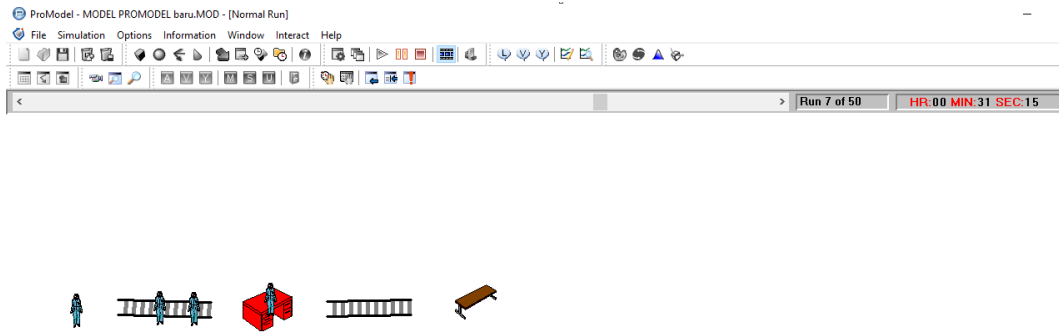
Berikut ini merupakan gambaran evaluasi *layout* dari sistem manufaktur atau jasa pada model *sistem* antrian :



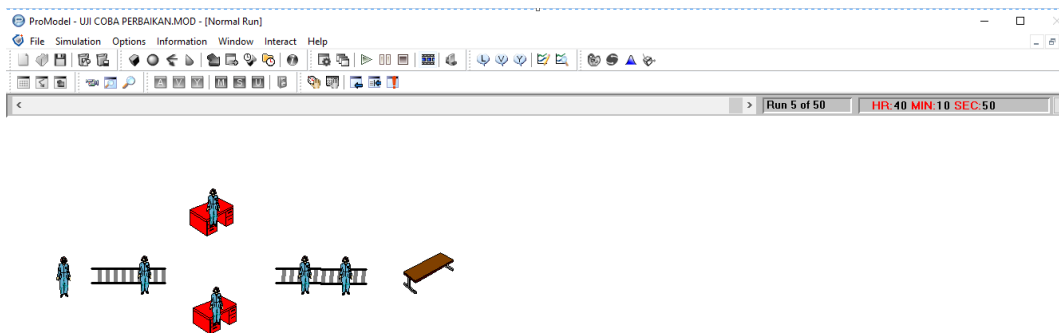
Gambar 8.1 Model *Layout* Sebelum Evaluasi Perbaikan



Gambar 8.2 Model *Layout* Sesudah Evaluasi Perbaikan



Gambar 8.3 Model *Layout* Sebelum Evaluasi Perbaikan dengan Menggunakan *Software* Promodel



Gambar 8.4 Model *Layout* Sesudah Evaluasi Perbaikan dengan Menggunakan *Software* Promodel

c. Performansi Sistem Antrian Sebelum Perbaikan

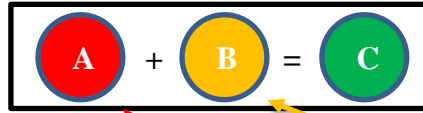
Performansi sistem antrian merupakan perhitungan total biaya yang dikeluarkan dalam sistem antrian. Berikut ini merupakan parameter dalam menentukan performansi sistem antrian :

1. Total Pelanggan yang Masuk dalam Sistem Antrian

Total pelanggan yang masuk dalam system antrian adalah jumlah responden yang diamati selama penelitian yaitu 50 orang.

2. Total Waktu Pelayanan

Total waktu pelayanan merupakan jumlah waktu pelayanan dari semua pelayanan yang terdapat pada data pengamatan. Berikut ini merupakan tabel total waktu pelayanan adalah :



Tabel 8.1 Total Waktu Pelayanan

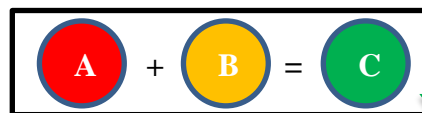
No	Waktu Antar Kedatangan	Waktu Menunggu 1	Waktu Proses 1	Waktu Menunggu 2	Waktu Proses 2	Total Waktu Pelayanan
1	0	1859	52	10	36	88
2	58	1863	67	8	42	109
3	75	1863	53	7	36	89
4	59	1864	57	9	38	95
5	64	1866	52	10	36	88
6	58	1870	62	11	40	102
7	70	1873	55	12	37	92
8	61	1879	52	15	36	88
9	58	1888	69	17	43	112
10	78	1896	52	14	36	88
11	58	1904	50	14	35	85
12	56	1912	56	10	37	93
13	62	1916	56	8	38	94
14	62	1918	53	7	36	89
15	59	1919	58	9	38	96
16	64	1922	53	10	36	89
17	59	1926	65	11	41	106
18	73	1929	52	12	36	88
19	58	1935	51	15	35	86
20	56	1945	50	17	35	85
21	55	1957	51	14	35	86
22	56	1966	52	14	36	88
23	58	1974	53	10	36	89
24	58	1979	56	8	38	94
25	63	1980	52	7	36	88
26	58	1981	51	9	35	86
27	56	1985	51	10	35	86
28	56	1990	51	11	36	87
29	57	1995	52	12	36	88
30	57	2002	64	15	40	104
31	72	2009	55	17	37	92
32	62	2019	50	14	35	85
33	55	2028	50	14	35	85
34	59	2033	52	10	36	88

Tabel 8.1 Lanjutan Total Waktu Pelayanan

No	Waktu Antar Kedatangan	Waktu Menunggu 1	Waktu Proses 1	Waktu Menunggu 2	Waktu Proses 2	Total Waktu Pelayanan
35	57	2038	52	8	36	88
36	57	2041	53	7	36	89
37	59	2042	60	9	39	99
38	66	2045	51	10	35	86
39	56	2050	57	11	38	95
40	64	2054	75	12	45	120
41	85	2056	56	15	37	93
42	62	2065	60	17	39	99
43	67	2075	55	14	37	92
44	61	2083	54	14	37	91
45	60	2091	57	10	38	95
46	63	2095	71	8	44	115
47	80	2094	55	7	37	92
48	62	2094	54	9	37	91
49	60	2097	52	10	36	88
50	57	2102	54	11	37	91
TOTAL WAKTU PELAYANAN						4632

3. Total Pelanggan Menunggu Antrian

Total pelanggan menunggu antrian merupakan jumlah waktu tunggu yang terdapat pada data pengamatan. Berikut ini merupakan data pengamatan total pelanggan menunggu antrian adalah :



Tabel 8.2 Total Pelanggan Menunggu Antrian

No	Waktu Antar Kedatangan	Waktu Menunggu 1	Waktu Proses 1	Waktu Menunggu 2	Waktu Proses 2	Total Pelanggan Menunggu Antrian
1	0	1859	52	10	36	1869
2	58	1863	67	8	42	1871
3	75	1863	53	7	36	1870
4	59	1864	57	9	38	1873
5	64	1866	52	10	36	1876
6	58	1870	62	11	40	1881

Tabel 8.2 Lanjutan Total Pelanggan Menunggu Antrian

No	Waktu Antar Kedatangan	Waktu Menunggu 1	Waktu Proses 1	Waktu Menunggu 2	Waktu Proses 2	Total Pelanggan Menunggu Antrian
7	70	1873	55	12	37	1885
8	61	1879	52	15	36	1894
9	58	1888	69	17	43	1905
10	78	1896	52	14	36	1910
11	58	1904	50	14	35	1918
12	56	1912	56	10	37	1922
13	62	1916	56	8	38	1924
14	62	1918	53	7	36	1925
15	59	1919	58	9	38	1928
16	64	1922	53	10	36	1932
17	59	1926	65	11	41	1937
18	73	1929	52	12	36	1941
19	58	1935	51	15	35	1950
20	56	1945	50	17	35	1962
21	55	1957	51	14	35	1971
22	56	1966	52	14	36	1980
23	58	1974	53	10	36	1984
24	58	1979	56	8	38	1987
25	63	1980	52	7	36	1987
26	58	1981	51	9	35	1990
27	56	1985	51	10	35	1995
28	56	1990	51	11	36	2001
29	57	1995	52	12	36	2007
30	57	2002	64	15	40	2017
31	72	2009	55	17	37	2026
32	62	2019	50	14	35	2033
33	55	2028	50	14	35	2042
34	59	2033	52	10	36	2043
35	57	2038	52	8	36	2046
36	57	2041	53	7	36	2048
37	59	2042	60	9	39	2051
38	66	2045	51	10	35	2055
39	56	2050	57	11	38	2061
40	64	2054	75	12	45	2066

Tabel 8.2 Lanjutan Total Pelanggan Menunggu Antrian

No	Waktu Antar Kedatangan	Waktu Menunggu 1	Waktu Proses 1	Waktu Menunggu 2	Waktu Proses 2	Total Pelanggan Menunggu Antrian
41	85	2056	56	15	37	2071
42	62	2065	60	17	39	2082
43	67	2075	55	14	37	2089
44	61	2083	54	14	37	2097
45	60	2091	57	10	38	2101
46	63	2095	71	8	44	2103
47	80	2094	55	7	37	2101
48	62	2094	54	9	37	2103
49	60	2097	52	10	36	2107
50	57	2102	54	11	37	2113
TOTAL PELANGGAN MENUNGGU ANTRIAN						99530
RATA-RATA PELANGGAN MENUNGGU ANTRIAN						1990,6

4. Rata – Rata Waktu Pelayanan ($1/\mu$) :

Rata-rata waktu pelayanan yang diberikan adalah :

= Total waktu pelayanan / banyak pelanggan yang dilayani

= 4632 / 50

= 92,64 detik per orang.

5. Perhitungan Biaya Pembangunan Model

Rumus Biaya Total Per Jam

$$TC = SC + WC$$

Keterangan :

TC = Total Biaya Per Jam

SC = Biaya Pelayanan Per Jam

WC = Biaya Menunggu Per Jam

λ = Jumlah Pelanggan yang Masuk Sistem

a. Menghitung Biaya Pelayanan

- Total waktu pelayanan = 92,64 detik/orang = 0,0257 jam / orang
- Biaya *Teller* Pemesanan Obat = Rp. 5.400.000 / bulan
 = Rp. 5.400.00 : 720 jam dalam 1 bulan
 = Rp. 7.500 / jam

- Biaya *Teller* Pembayaran Kasir = Rp. 3.700.000 / bulan
 = Rp.3.700.000 : 720 jam dalam 1 bulan
 = Rp. 5.000 / jam
- Biaya Fasilitas Pemesanan Obat = Rp. 1.000.000 / bulan
 = Rp. 1.000.000 : 720 jam dalam 1 bulan
 = Rp. 1.388 / jam
- Biaya Fasilitas Kasir = Rp. 1.500.000 / bulan
 = Rp. 1.500.000 : 720 jam dalam 1 bulan
 = Rp. 2.083 / jam
- Biaya Pelayanan 1 (Pesan Obat) = (Waktu Pelayanan x Biaya *Teller*) +
 (Waktu Pelayanan x Biaya Fasilitas)
 = (0,0257 x 7500) + (0,0257 x 1388)
 = (192,75) + (35,67)
 = Rp. 228,42 / jam
- Biaya Pelayanan 2 (Kasir) = (Waktu Pelayanan x Biaya *Teller*) +
 (Waktu Pelayanan x Biaya Fasilitas)
 = (0,0257 x 5000) + (0,0257 x 2083)
 = (128,5) + (53,53)
 = Rp. 182,03 / jam
- Total Biaya Pelayanan (SC) = Biaya Pelayanan 1 + Biaya Pelayanan 2
 = Rp. 228,42 / jam + Rp. 182,03 / jam
 = Rp. 410,45 / jam

b. Menghitung Biaya Menunggu

- Rata-Rata Waktu Pelanggan Menunggu = 1990,6 detik / orang
 = 1990,6 : 3600
 = 0,553 jam / orang
- Jumlah Pelanggan yang Masuk Sistem = 50 orang (sesuai jumlah data)
- Rata-Rata Biaya Menunggu 1 = Rp. 5.400.000 / bulan
 = Rp. 5.400.00 : 720 jam / bulan
 = Rp. 7.500 / jam
- Rata-Rata Biaya Menunggu 2 = Rp. 3.700.000 / bulan
 = Rp.3.700.000 : 720 jam / bulan

- = Rp. 5.000 / jam
 - Biaya Menunggu 1
 - = (Rata-Rata Waktu Pelanggan
Menunggu X Jumlah Pelanggan
yang Masuk Sistem X Rata-Rata
Biaya Menunggu 1)
 - = (0,553 x 50 x 7500)
 - = Rp. 207.375 / jam
 - Biaya Menunggu 2
 - = (Rata-Rata Waktu Pelanggan
Menunggu X Jumlah Pelanggan
yang Masuk Sistem X Rata-Rata
Biaya Menunggu 2)
 - = (0,553 x 50 x 5000)
 - = Rp. 138.250 / jam
 - Total Biaya Menunggu (WC)
 - = Biaya Menunggu 1 + Biaya
Menunggu 2
 - = Rp. 207.375 + Rp. 138.250
 - = Rp. 345.625 / jam

c. Total Biaya Per jam

- $TC = SC + WC$
 - = Total Biaya Pelayanan + Total Biaya Menunggu
 - = Rp. 410,45 / jam + Rp. 345.625 / jam
 - = Rp. 346.035,45 / jam.

d. Performansi Sistem Antrian Sesudah Perbaikan

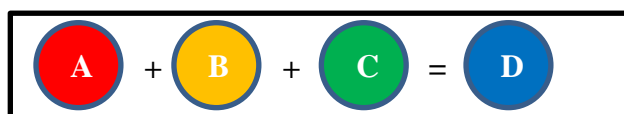
Performansi sistem antrian merupakan perhitungan total biaya yang dikeluarkan dalam sistem antrian. Berikut ini merupakan parameter dalam menentukan performansi sistem antrian :

1. Total Pelanggan yang Masuk dalam Sistem Antrian

Total pelanggan yang masuk dalam system antrian adalah jumlah responden yang diamati selama penelitian yaitu 50 orang.

2. Total Waktu Pelayanan

Total waktu pelayanan merupakan jumlah waktu pelayanan dari semua pelayanan yang terdapat pada data pengamatan. Berikut ini merupakan tabel total waktu pelayanan adalah :



Tabel 8.1 Total Waktu Pelayanan

No	Waktu Antar Kedatangan	Waktu Menunggu 1	Waktu Proses 1 Server 1	Waktu Proses 1 Server 2	Waktu Menunggu 2	Waktu Proses 2	Total Waktu Pelayanan
1	0	1859	52	52	10	36	140
2	58	1863	67	67	8	42	176
3	75	1863	53	53	7	36	142
4	59	1864	57	57	9	38	152
5	64	1866	52	52	10	36	140
6	58	1870	62	62	11	40	164
7	70	1873	55	55	12	37	147
8	61	1879	52	52	15	36	140
9	58	1888	69	69	17	43	181
10	78	1896	52	52	14	36	140
11	58	1904	50	50	14	35	135
12	56	1912	56	56	10	37	149
13	62	1916	56	56	8	38	150
14	62	1918	53	53	7	36	142
15	59	1919	58	58	9	38	154
16	64	1922	53	53	10	36	142
17	59	1926	65	65	11	41	171

Tabel 8.1 Lanjutan Total Waktu Pelayanan

No	Waktu Antar Kedatangan	Waktu Menunggu 1	Waktu Proses 1 Server 1	Waktu Proses 1 Server 2	Waktu Menunggu 2	Waktu Proses 2	Total Waktu Pelayanan
18	73	1929	52	52	12	36	140
19	58	1935	51	51	15	35	137
20	56	1945	50	50	17	35	135
21	55	1957	51	51	14	35	137
22	56	1966	52	52	14	36	140
23	58	1974	53	53	10	36	142
24	58	1979	56	56	8	38	150
25	63	1980	52	52	7	36	140
26	58	1981	51	51	9	35	137
27	56	1985	51	51	10	35	137
28	56	1990	51	51	11	36	138
29	57	1995	52	52	12	36	140
30	57	2002	64	64	15	40	168
31	72	2009	55	55	17	37	147
32	62	2019	50	50	14	35	135
33	55	2028	50	50	14	35	135
34	59	2033	52	52	10	36	140
35	57	2038	52	52	8	36	140
36	57	2041	53	53	7	36	142
37	59	2042	60	60	9	39	159
38	66	2045	51	51	10	35	137
39	56	2050	57	57	11	38	152
40	64	2054	75	75	12	45	195
41	85	2056	56	56	15	37	149
42	62	2065	60	60	17	39	159
43	67	2075	55	55	14	37	147
44	61	2083	54	54	14	37	145
45	60	2091	57	57	10	38	152
46	63	2095	71	71	8	44	186
47	80	2094	55	55	7	37	147
48	62	2094	54	54	9	37	145
49	60	2097	52	52	10	36	140
50	57	2102	54	54	11	37	145
TOTAL WAKTU PELAYANAN							7403

3. Total Pelanggan Menunggu Antrian

Total pelanggan menunggu antrian merupakan jumlah waktu tunggu yang terdapat pada data pengamatan. Berikut ini merupakan data pengamatan total pelanggan menunggu antrian adalah :



Tabel 8.2 Total Pelanggan Menunggu Antrian

No	Waktu Antar Kedatangan	Waktu Menunggu 1	Waktu Proses 1 Server 1	Waktu Proses 1 Server 2	Waktu Menunggu 2	Waktu Proses 2	Total Pelanggan Menunggu Antrian
1	0	1859	52	52	10	36	1869
2	58	1863	67	67	8	42	1871
3	75	1863	53	53	7	36	1870
4	59	1864	57	57	9	38	1873
5	64	1866	52	52	10	36	1876
6	58	1870	62	62	11	40	1881
7	70	1873	55	55	12	37	1885
8	61	1879	52	52	15	36	1894
9	58	1888	69	69	17	43	1905
10	78	1896	52	52	14	36	1910
11	58	1904	50	50	14	35	1918
12	56	1912	56	56	10	37	1922
13	62	1916	56	56	8	38	1924
14	62	1918	53	53	7	36	1925
15	59	1919	58	58	9	38	1928
16	64	2091	53	53	10	36	1932
17	59	2095	65	65	11	41	1937
18	73	2094	52	52	12	36	1941
19	58	2094	51	51	15	35	1950
20	56	2097	50	50	17	35	1962
21	55	2102	51	51	14	35	1971
22	56	2091	52	52	14	36	1980
23	58	2095	53	53	10	36	1984
24	58	2094	56	56	8	38	1987
25	63	2094	52	52	7	36	1987
26	58	2097	51	51	9	35	1990

Tabel 8.2 Lanjutan Total Pelanggan Menunggu Antrian

No	Waktu Antar Kedatangan	Waktu Menunggu 1	Waktu Proses 1 Server 1	Waktu Proses 1 Server 2	Waktu Menunggu 2	Waktu Proses 2	Total Pelanggan Menunggu Antrian
27	56	2102	51	51	10	35	1995
28	56	2091	51	51	11	36	2001
29	57	2095	52	52	12	36	2007
30	57	2094	64	64	15	40	2017
31	72	2094	55	55	17	37	2026
32	62	2097	50	50	14	35	2033
33	55	2102	50	50	14	35	2042
34	59	2091	52	52	10	36	2043
35	57	2095	52	52	8	36	2046
36	57	2094	53	53	7	36	2048
37	59	2094	60	60	9	39	2051
38	66	2097	51	51	10	35	2055
39	56	2102	57	57	11	38	2061
40	64	2091	75	75	12	45	2066
41	85	2095	56	56	15	37	2071
42	62	2094	60	60	17	39	2082
43	67	2094	55	55	14	37	2089
44	61	2097	54	54	14	37	2097
45	60	2102	57	57	10	38	2101
46	63	2095	71	71	8	44	2103
47	80	2094	55	55	7	37	2101
48	62	2094	54	54	9	37	2103
49	60	2097	52	52	10	36	2107
50	57	2102	54	54	11	37	2113
TOTAL PELANGGAN MENUNGGU ANTRIAN							99530
RATA-RATA PELANGGAN MENUNGGU ANTRIAN							1990,6

4. Rata – Rata Waktu Pelayanan ($1/\mu$) :

Rata-rata waktu pelayanan yang diberikan adalah :

= Total waktu pelayanan / banyak pelanggan yang dilayani

= 7403 / 50

= 148,06 detik per orang.

5. Perhitungan Biaya Pembangunan Model

Rumus Biaya Total Per Jam

$$TC = SC + WC$$

Keterangan :

TC = Total Biaya Per Jam

SC = Biaya Pelayanan Per Jam

WC = Biaya Menunggu Per Jam

λ = Jumlah Pelanggan yang Masuk Sistem

a. Menghitung Biaya Pelayanan

- Total waktu pelayanan = 148,06 detik/orang = 0,041 jam / orang
- Biaya *Teller* Pemesanan Obat (2 org) = Rp. 5.400.000 / bulan x 2
= Rp. 10.800.000 : 720 jam / bulan
= Rp. 15.000 / jam
- Biaya *Teller* Pembayaran Kasir = Rp. 3.700.000 / bulan
= Rp. 3.700.000 : 720 jam / bulan
= Rp. 5.000 / jam
- Biaya Fasilitas Pemesanan Obat (2 org) = Rp. 1.000.000 / bulan x 2
= Rp. 2.000.000 : 720 jam / bulan
= Rp. 2.777 / jam
- Biaya Fasilitas Kasir = Rp. 1.500.000 / bulan
= Rp. 1.500.000 : 720 jam / bulan
= Rp. 2.083 / jam
- Biaya Pelayanan 1 (Pesan Obat) = (Waktu Pelayanan x Biaya Teller)
+
(Waktu Pelayanan x Biaya Fasilitas)
= (0,041 x 15.000) + (0,041 x 2.777)
= (615) + (113,86)
= Rp. 728,86 / jam
- Biaya Pelayanan 2 (Kasir) = (Waktu Pelayanan x Biaya Teller)
+

(Waktu Pelayanan x Biaya Fasilitas)

$$= (0,041 \times 5000) + (0,041 \times 2083)$$

$$= (205) + (85,40)$$

$$= \text{Rp. } 290,4 / \text{jam}$$

- Total Biaya Pelayanan (SC)

$$= \text{Biaya Pelayanan 1} + \text{Biaya Pelayanan 2}$$

$$= \text{Rp. } 728,86 / \text{jam} + \text{Rp. } 290,4 / \text{jam}$$

$$= \text{Rp. } 1.019,26 / \text{jam}$$

b. Menghitung Biaya Menunggu

- Rata-Rata Waktu Pelanggan Menunggu = 1990,6 detik / orang

$$= 1990,6 : 3600$$

$$= 0,553 \text{ jam} / \text{orang}$$

- Jumlah Pelanggan yang Masuk Sistem = 50 orang (sesuai jumlah data)

- Rata-Rata Biaya Menunggu 1

$$= \text{Rp. } 10.800.000 / \text{bulan}$$

$$= \text{Rp. } 10.800.000 : 720 \text{ jam} / \text{bulan}$$

$$= \text{Rp. } 15.000 / \text{jam}$$

- Rata-Rata Biaya Menunggu 2

$$= \text{Rp. } 3.700.000 / \text{bulan}$$

$$= \text{Rp. } 3.700.000 : 720 \text{ jam} / \text{bulan}$$

$$= \text{Rp. } 5.000 / \text{jam}$$

- Biaya Menunggu 1

$$= (\text{Rata-Rata Waktu Pelanggan}$$

Menunggu X Jumlah Pelanggan yang Masuk Sistem X Rata-Rata Biaya Menunggu 1)

$$= (0,553 \times 50 \times 15.000)$$

$$= \text{Rp. } 414.750 / \text{jam}$$

- Biaya Menunggu 2

$$= (\text{Rata-Rata Waktu Pelanggan}$$

Menunggu X Jumlah Pelanggan yang Masuk Sistem X Rata-Rata Biaya Menunggu 2)

$$= (0,553 \times 50 \times 5000)$$

$$= \text{Rp. } 138.250 / \text{jam}$$

- Total Biaya Menunggu (WC) = Biaya Menunggu 1 + Biaya Menunggu 2
= Rp. 414.750 + Rp. 138.250
= Rp. 553.000 / jam

c. Total Biaya Per jam

- $TC = SC + WC$
= Total Biaya Pelayanan + Total Biaya Menunggu
= Rp. 1019,26 / jam + Rp. 553.000 / jam
= Rp. 554.019,26 / jam.

DAFTAR PUSTAKA

- Banks, J. (1998). *Hand Book of Simulation : Aplication, Methodology, Advances, Applications and Practices*,. John and Willey Sons.
- Law. A. M, , K. (2000). *Simulation Modeling and Analysis 3rd Editions*. Mc. Graw Hill,.

LAMPIRAN

LAPORAN PRAKTIKUM SIMULASI KOMPUTER



Disusun Oleh :

KELOMPOK X

- | | |
|---------|-------------|
| 1. | 16.0000.001 |
| 2. | 16.0000.002 |
| 3. | 16.0000.003 |
| 4. | 16.0000.004 |
| 5. | 16.0000.005 |

**LABORATORIUM DECISION SUPPORT SYSTEM
PROGRAM STUDI TEKNIK INDUSTRI
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG
2022-2023**

Lampiran 2 : Contoh Format Halaman Pengesahan Laporan

LEMBAR PENGESAHAN LAPORAN

Laporan ini disusun untuk memenuhi salah satu tugas Praktikum Simulasi Komputer pada Fakultas Teknologi Industri Program Studi Teknik Industri Semester Gasal Tahun Ajaran 2022-2023, Telah disetujui dan disahkan pada :

Hari :

Tanggal :

Kelompok :

Oleh;

Asisten Praktikum Simulasi Komputer

1. **Ali Wedo Sarjono, S.T, M.Kom**
2. **Muchamad Maknun, S.T**
3. **Dio Rama Harpriyanto**
4. **Hasna Fauziah**
5. **Nisa Atun Nikmah**
6. **Reza Rizki Saputro**
7. **Taufiq Ulin Nuha**

Mengetahui,

Dosen Pengampu Praktikum Simulasi Komputer

Brav Deva Bernadhi., ST, MT

Lampiran 2 : Contoh Format Laporan

HALAMAN LAPORAN

LEMBAR PENGESAHAN

DAFTAR ISI

BAB I TEKNIK PENGAMBILAN DATA SIMULASI KOMPUTER

1.1 Pendahuluan

1.2 Hasil Rekapitulasi Pengambilan Data

1.3 Analisa Uji Statistik dengan Menggunakan Software SPSS

**BAB II PEMBANGKITAN SEJARAH SISTEM DENGAN
MENGUNAKAN SOFTWARE PROMODEL**

2.1 Pendahuluan

2.2 Pengolahan Data Manual

2.2.1 Gambar Layout Sisem Manufactur atau Jasa

2.2.2 Entities

2.2.3 Resources

2.2.4 Aktivitas dan Event

2.3 Pengolahan Data Simulasi Komputer

2.3.1 Uji Distribusi Manual

2.3.2 Uji Distribusi Dengan Menggunakan Software Promodel

2.3.3 Pembuatan Model Dengan Menggunakan Software Promodel

**BAB III PEMBANGKITAN SEJARAH SISTEM DENGAN
MENGUNAKAN SOFTWARE EXTEND**

3.1 Pendahuluan

3.2 Pengolahan Data Manual

3.2.1 Gambar Layout Sisem Manufactur atau Jasa

3.2.2 Tata Urutan Proses

3.2.3 Asumsi-asumsi

3.2.4 Entitas

3.2.5 Aktivitas dan Event

3.3 Pengolahan Data dengan Menggunakan Software Extend

3.3.1 Pembuatan Model Dengan Menggunakan Software Extend

3.3.2 Quene, FIFO Antrian

3.3.3 Activity, Delay Antrian

**BAB IV PEMBANGKITAN SEJARAH SISTEM DENGAN
MENGUNAKAN SOFTWARE ARENA 14 SIMULATION**

4.1 Pendahuluan

4.2 Pengolahan Data Manual

4.2.1 *Flowchart Sistem*

4.2.2 *Rich Picture Diagram*

4.2.3 *Activity Cycle Diagram*

4.3 Pengolahan Data Dengan Menggunakan *Software Arena 14*

4.3.1 Pembuatan Model Dengan Menggunakan *Software Arena 14*

4.3.2 Rekapitulasi Hasil *Report*

BAB V VERIFIKASI DAN VALIDASI SISTEM

5.1 Pendahuluan

5.2 Software Promodel

5.2.1 Verifikasi

5.2.2 Validasi

5.3 Software Extend

5.3.1 Verifikasi

5.3.2 Validasi

5.4 Software Arena

5.3.1 Verifikasi

5.3.2 Validasi

BAB VI PERANCANGAN ALTERNATIF DAN PEMILIHAN MODEL TERBAIK

6.1 Pengolahan Data Perbaikan dengan Menggunakan Software Promodel

6.1.1 Pembuatan Model Dengan Menggunakan Software Promodel

6.1.2 Perhitungan tingkat ekonomis sebelum dan sesudah perbaikan

BAB VII ANALISA

7.1 Analisa Software Promodel

7.1.1 Analisa Distribusi

7.1.2 Analisa Pengolahan Data Manual

7.1.3 Analisa Output Running

7.2 Analisa Software Extend

7.2.1 Analisa Antrian Queue Saat Pengamatan

7.2.2 Analisa Antrian Activity Saat Pengamatan

7.3 Analisa Software Arena

7.2.1 Analisa Entity

7.2.2 Analisa Queue

7.2.3 Analisa Resource

7.3 Analisa Perbaikan Software Promodel

7.3.1 Analisa Distribusi

7.3.2 Analisa Pengolahan Data Manual

7.3.3 Analisa Output Running

7.3.4 Analisa Perbandingan Tingkat Ekonomis

BAB VIII PENUTUP

8.1 Kesimpulan

8.2 Saran