

**SIMULASI SISTEM PERSEDIAAN SPARE PART DENGAN PENDEKATAN
COMPOUND POISSON PROCESS
(Studi Kasus di PT. PETROKIMIA GRESIK PABRIK I)**

**SKRIPSI
KONSENTRASI SISTEM MANUFAKTUR DAN JASA**

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



Disusun Oleh :

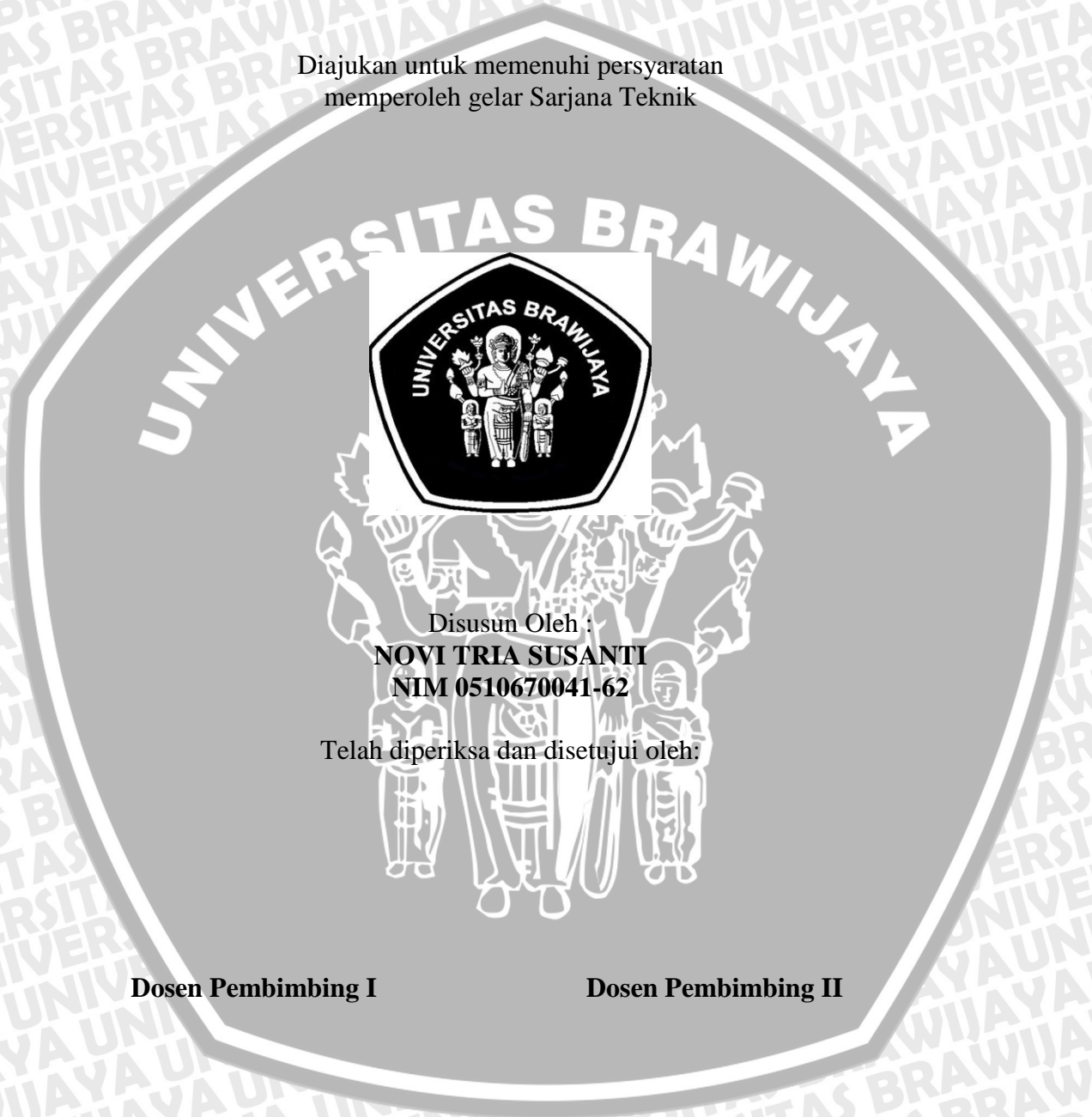
**NOVI TRIA SUSANTI
NIM 0510670041-62**

**DEPARTEMEN PENDIDIKAN NASIONAL
UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
JURUSAN TEKNIK MESIN
PROGRAM STUDI TEKNIK INDUSTRI
MALANG
2009**

**SIMULASI SISTEM PERSEDIAAN SPARE PART DENGAN PENDEKATAN
COMPOUND POISSON PROCESS
(Studi Kasus di PT. PETROKIMIA GRESIK PABRIK I)**

**SKRIPSI
KONSENTRASI SISTEM MANUFAKTUR DAN JASA**

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



Disusun Oleh :
NOVI TRIA SUSANTI
NIM 0510670041-62

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Ir. Bambang Indrayadi, MT.
NIP. 19600905 198701 1 001

Arif Rahman ST., MT.
NIP. 19740528 200801 1 010

LEMBAR PENGESAHAN**SIMULASI SISTEM PERSEDIAAN SPARE PART DENGAN PENDEKATAN
COMPOUND POISSON PROCESS
(Studi Kasus di PT. PETROKIMIA GRESIK PABRIK I)****SKRIPSI
KONSENTRASI SISTEM MANUFAKTUR DAN JASA**

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik

Disusun Oleh :

**NOVI TRIA SUSANTI
NIM 0510670041-62**

Skripsi ini telah diuji dan dinyatakan lulus pada
Tanggal 13 Agustus 2009

Penguji Skripsi I

Penguji Skripsi II

Dr. Ir. Pratikto, MMT.
NIP. 19461110 198103 1 001

Dra. Murti Astuti, MSIE.
NIP. 19610620 198603 2 001

Penguji Komprehensif

Ir. Masduki, MM
NIP. 19450816 197009 1 001

**Mengetahui,
Plh. Ketua Program Studi Teknik Industri**

Nasir Widha Setyanto ST., MT.
NIP. 19700914 200501 1 001

KATA PENGANTAR

Alhamdulillah, dengan mengucapkan puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya sehingga penyusunan skripsi ini dapat terselesaikan dengan baik. Shalawat serta salam tak lupa kita berikan pada sosok yang agung Rasulullah SAW, keluarga, sahabat, serta umatnya. Semoga kita mendapat syafa'at beliau di akhir zaman. Dalam penulisan skripsi ini penulis banyak mendapatkan bantuan dari berbagai pihak, oleh sebab itu penulis mengucapkan terima kasih yang tulus kepada :

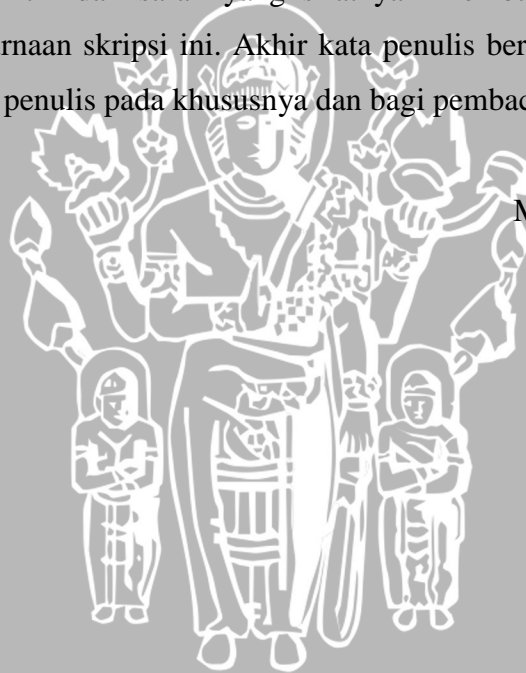
1. Papa dan mama tercinta. mbak dan mas yang telah memberikan dukungan moril dan semangat yang sangat berarti bagi penulis.
2. Bapak Nasir W. Setyanto, ST., MT. selaku Plh. Ketua Program Studi Teknik Industri Fakultas Teknik Universitas Brawijaya.
3. Bapak Ir. Bambang Indrayadi, MT. selaku Ketua Kelompok Konsentrasi Sistem Manufaktur dan Jasa dan Dosen Pembimbing I yang telah memberikan kritik dan saran yang membangun selama masa studi dan dalam penyusunan skripsi ini.
4. Bapak Arif Rahman, ST., MT. selaku Dosen Wali dan Dosen Pembimbing II yang telah memberikan kritik dan saran yang membangun selama masa studi dan dalam penyusunan skripsi ini.
5. Bapak Ishardita Pambudi Tama, ST., MT. selaku Mantan Kepala Laboratorium Perancangan Kerja dan Ergonomi.
6. Bapak Ir. Arif Teguh Kabag Candal Dephar 1 PT. Petrokimia Gresik selaku pembimbing skripsi di perusahaan, terima kasih atas bantuan dalam pengambilan data.
7. Bapak Parmono dan Mas Reza selaku pegawai administrasi PS. Teknik Industri, terima kasih atas bantuan dan kesabarannya dalam pengurusan administrasi.
8. Sahabat-sahabat tersayang Cecylia, Indri, Ika, Eli, Dek Ulid, Dek Berlian, Dek Tia, Erita, Coi, Vino, Feri, Dian, Yuyun, Qarina, Mbak Vina terima kasih atas semangat dan perhatian yang telah diberikan.
9. Teman-teman industri 2005 dan Mesin 2005 khususnya untuk teman-teman yang sangat berkesan Angga, Junta, Bungkring, Gery, Bayu, Amanda, Anggun, Destri, Riza, Anes, terima kasih atas ilmu dan cerita-cerita yang tak terlupakan.

10. Teman-teman Industri 2006, 2007, 2008 atas bantuan yang diberikan.
11. Teman-teman Ikatan Alumni Beswan Djarum DSO Malang.
12. Anisa, Miranda, Mbak Diah, Norma dan Keluarga Besar Yayasan Cendekia Kota Malang yang telah memberikan semangat dan selalu mengingatkan untuk tetap istiqomah.
13. *My Motivator* terima kasih untuk semangat dan do'a yang tak henti-hentinya sehingga skripsi ini dapat terselesaikan.
14. Semua pihak yang secara langsung atau tidak langsung yang telah membantu dalam kelancaran terselesaikannya skripsi ini dengan baik.

Penulis menyadari sepenuhnya bahwa skripsi ini masih terdapat banyak kekurangan karena itu kritik dan saran yang sifatnya membangun sangat penulis harapkan dalam kesempurnaan skripsi ini. Akhir kata penulis berharap semoga skripsi ini dapat bermanfaat bagi penulis pada khususnya dan bagi pembaca pada umumnya.

Malang, Agustus 2009

Penulis



DAFTAR ISI

Halaman

HALAMAN JUDUL	
LEMBAR PERSETUJUAN	
PERNYATAAN ORISINALITAS	
PENGANTAR	i
DAFTAR ISI	iii
DAFTAR TABEL	vii
DAFTAR GAMBAR	viii
DAFTAR LAMPIRAN	ix
RINGKASAN	x
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	4
1.3 Rumusan Masalah	4
1.4 Batasan Masalah	5
1.5 Tujuan Penelitian	5
1.6 Manfaat Penelitian	5
1.7 Manfaat Penelitian	6
BAB II TINJAUAN PUSTAKA	7
2.1 Manajemen <i>Spare Part</i>	7
2.1.1 Definisi Komponen	7
2.2 Manajemen Persediaan	8
2.2.1 Fungsi-fungsi Persediaan	9
2.3 Metode Pengendalian Persediaan	10
2.3.1 Model Service Level	11
2.3.2 Reorder Point	11



2.4	Proses Stokastik	12
2.5	Jenis-jenis Distribusi	12
2.5.1	Jenis-jenis Distribusi Diskrit	12
2.5.2	Jenis-jenis Distribusi Kontinyu	13
2.5.3	Estimasi Parameter Distribusi dengan <i>Maximum Likelihood</i>	14
2.6	<i>Compound Poisson Process</i>	14
2.7	Bilangan Acak	15
2.7.1	Pembangkitan Bilangan Acak	15
2.8	Sistem	16
2.8.1	Karakteristik Sistem	17
2.8.2	Klasifikasi Sistem	18
2.9	Model	19
2.9.1	Latar Belakang Model	19
2.9.2	Batasan dan Asumsi Model	20
2.9.3	Prinsip Pemodelan	20
2.10	Simulasi	21
2.10.1	Keuntungan Simulasi	21
2.11	Pengujian Statistik	22
2.11.1	Pengujian <i>Independent T-Test</i>	22
2.11.2	Pengujian ANOVA	23
2.12	Metode “ <i>what if analysis</i> ”	24
2.13	<i>Visual Basic 6.0</i>	25
2.13.1	<i>Integrated Development Environment (IDE)</i>	26
2.13.2	Menjalankan IDE	27
2.13.3	Jendela IDE	28
2.13.4	<i>ToolBox</i>	30

BAB III METODOLOGI PENELITIAN 33

3.1	Metode Penelitian	33
3.2	Obyek Penelitian	33
3.3	Jenis dan Sumber Data	33
3.4	Metode Alur Penelitian	34



3.4.1	<i>Survey Lapangan</i>	34
3.4.2	<i>Studi Literatur</i>	34
3.4.3	Identifikasi Masalah.....	34
3.4.4	Penetapan Tujuan Penelitian.....	34
3.4.5	Pengumpulan Data	34
3.4.6	Uji Distribusi.....	35
3.4.7	Penentuan Parameter Distribusi.....	35
3.4.8	Pembuatan Model Matematis	35
3.4.9	Perancangan Program Simulasi	35
3.4.10	Verifikasi Model	35
3.4.11	Keluaran Simulasi	35
3.4.12	Validasi Model	36
3.4.13	Analisa Hasil Simulasi.....	36
3.4.14	Rancangan Percobaan	36
3.4.15	Analisa Hasil Eksperimen	36
3.4.16	Kesimpulan dan Saran	36
3.5	Diagram Alir Metodologi Penelitian	37

BAB IV PENGUMPULAN dan PENGOLAHAN DATA 39

4.1	<i>Studi Lapangan</i>	39
4.1.1	<i>Consumption Rate (CR)</i>	42
4.1.2	<i>Lead Time (LT)</i>	42
4.1.3	<i>Safety Factor (SF)</i>	43
4.2	<i>Pengumpulan data</i>	43
4.2.1	Kebijakan Sistem Persediaan PT. Petrokimia Gresik	44
4.2.2	Jenis-jenis <i>Spare Part</i>	44
4.2.3	Nilai <i>Inventory Level (IL)</i> , nilai ROP, nilai Q dan <i>service level, lead time</i>	45
4.2.4	Data Pengambilan <i>Spare part</i>	45
4.2.5	Data <i>Inventory Value Spare part</i>	45
4.3	<i>Uji Pendugaan Parameter</i>	45
4.3.1	Pendugaan Parameter untuk waktu antar	



pengambilan <i>Spare part</i>	46
4.3.2 Pendugaan parameter untuk jumlah part yang diambil	46
4.4 Pembuatan Model Matematis	47
4.5 Model Simulasi	49
4.5.1 Pendekatan Program Simulasi	49
4.5.2 <i>Flowchart</i>	50
4.5.2.1 <i>Flowchart</i> Anatomi Program Simulasi	50
4.5.2.2 <i>Flowchart</i> Permintaan <i>Maintenance</i> dan Barang Datang	53
4.6 Perancangan dan Verifikasi Program Simulasi	57
4.6.1 Perancangan Desain Antar Muka	57
4.6.2 Kode Pemograman	59
4.6.2.1 Kode <i>Library Routine</i>	59
4.6.2.2 Kode <i>Main Program</i>	64
4.6.2.3 Kode Inisialisasi	65
4.6.2.4 <i>Timing</i>	66
4.6.2.5 <i>Update</i> Statistik	66
4.6.2.6 <i>Event</i> Permintaan <i>Maintenance</i>	67
4.6.2.7 <i>Event</i> Barang Datang	67
4.6.3 Verifikasi Model	68
4.7 Keluaran Program Simulasi dalam <i>Pilot Run</i>	68
4.8 Validasi Model	69
4.9 Analisa Hasil Simulasi	72
4.9.1 Hasil eksperimen dan Pengujian Hasil Simulasi	73
4.9.1.1 Hasil eksperimen	73
4.9.1.2 Uji ANOVA Hasil Eksperimen	73
4.10. Analisa Hasil Eksperimen	77
BAB V PENUTUP	80
5.1 Kesimpulan	80
5.2 Saran	81
DAFTAR PUSTAKA	82
LAMPIRAN	83

DAFTAR TABEL

No.	Judul	Halaman
Tabel 1.1.	Target Produksi pada Pabrik 1	1
Tabel 1.2.	Tabel Pencapaian produksi, Waktu <i>Stream Day</i> , dan Waktu <i>Down Time</i>	2
Tabel 1.3.	Tabel Data Pemakaian <i>Spare Part</i>	3
Tabel 2.1.	Tabel ANOVA	24
Tabel 4.1.	Jenis-Jenis <i>Spare Part</i>	44
Tabel 4.2.	Nilai Q, ROP, IL, Nilai <i>Service Level</i> dan Nilai <i>Lead Time</i>	45
Tabel 4.3.	<i>Inventory Value Spare Part</i>	45
Tabel 4.4.	Hasil Pendugaaan Parameter Untuk Waktu antar Pengambilan <i>Spare Part</i>	46
Tabel 4.5.	Tabel Nilai Lamda Untuk Distribusi Poisson	46
Tabel 4.6.	Keluaran Simulasi <i>Pilot Run</i>	69
Tabel 4.7.	Hasil Perbandingan Sistem Nyata dengan Hasil Simulasi	70
Tabel 4.8.	Perancangan Eksperimen Simulasi Sistem Persediaan	72
Tabel 4.9.	Tabel ANOVA	73
Tabel 4.10.	Pengujian ANOVA Nilai IL rata-rata <i>Spare Part</i>	74
Tabel 4.11.	Uji BNT Untuk <i>Spare Part</i>	76
Tabel 4.12.	Sistem Persediaan Usulan dengan Menggunkan " <i>what if analysis</i> "	77
Tabel 4.13	Perbandingan <i>Inventory Value</i> Antara Sistem Nyata dengan Hasil Simulasi	79

DAFTAR GAMBAR

No.	Judul	Halaman
Gambar 1.1.	Grafik Prersentase Faktor-faktor yang mempengaruhi Realisasi Produksi	2
Gambar 2.1.	Kotak Dialog <i>New Project</i>	27
Gambar 2.2.	Jendela IDE	29
Gambar 2.3.	<i>Tool Box</i> Pada VB 6.0.	31
Gambar 3.1	Diagram alir penelitian	36
Gambar 4.1	Logo PT. Petrokimia Gresik	39
Gambar 4.2	Proses Pengadaan Barang di PT. Petrokimia Gresik	40
Gambar 4.3	Proses Pengadaan Barang ROL	41
Gambar 4.4	<i>Influence Diagram</i>	47
Gambar 4.5	<i>Event Graph</i> Sistem	49
Gambar 4.6	<i>Main Program</i> Program Simulasi	51
Gambar 4.7	<i>Flowchart</i> Permintaan <i>Maintenance</i>	55
Gambar 4.8	<i>Flowchart</i> Barang Datang	56
Gambar 4.9	Desain Antar Muka Program	58
Gambar 4.10	Gambar Eksekusi Simulasi	58



DAFTAR LAMPIRAN

No.	Judul	Halaman
Lampiran 1.	Data Waktu Pengambilan <i>Spare Part</i>	83
Lampiran 2.	Nilai IL Hasil Simulasi	89



RINGKASAN

Novi Tria Susanti, Program Studi Teknik Industri Jurusan Teknik Mesin, Fakultas Teknik, Universitas Brawijaya, Agustus 2009, *Simulasi Sistem Persediaan Spare Part dengan Pendekatan Compound Poisson Process di PT.Petrokimia Gresik Pabrik1*, Dosen Pembimbing : Ir. Bambang Indrayadi, MT., dan Arif Rahman, ST., MT.

Sistem Persediaan *spare part* merupakan hal yang sangat penting dalam *maintenance organization*. Dengan sistem persediaan yang tepat akan meningkatkan produktifitas dan mampu mengurangi *down time* mesin. Sistem persediaan *spare part* dipengaruhi oleh banyak faktor salah satunya adalah tingkat pemakaian dari *spare part* tersebut. Pemakaian *spare part* di PT.Petrokimia Gresik mengalami fluktuasi hal ini menyebabkan tingkat persediaan yang terlalu tinggi.

Untuk menentukan sistem persediaan yang tepat, banyak cara yang digunakan salah satunya adalah dengan menggunakan simulasi. Dimana simulasi merupakan metode untuk membuat tiruan dari sistem nyata ke dalam sistem buatan, yang biasanya dibuat dalam komputer melalui program komputer. Dalam pembuatan simulasi bahasa pemrograman yang digunakan adalah Visual Basic 6.0. Data yang dibutuhkan dalam pembuatan simulasi adalah waktu permintaan *spare part* dan jumlah pengambilan *spare part*. Untuk menggambarkan sistem yang terjadi dilakukan pendekatan *Compound Poisson Process*, hal ini dikarenakan terdapat dua distribusi yang terjadi dalam 1 sistem. Sistem persediaan rancangan percobaan adalah sistem Q, dengan menggunakan "what if analysis".

Penelitian yang dilakukan pada 10 jenis *spare part* yang tergolong RO atau *spare part* yang harus selalu terdapat di gudang. Dengan menjalankan simulasi selama 1460 hari dan replikasi sebanyak 10 kali, didapatkan hasil untuk *spare part* Kode A nilai Q 20 unit, ROP 12 unit dan IL rata-rata 25 unit. *Spare part* kode B nilai Q 50 unit, nilai ROP 30 unit, nilai IL rata-rata 72 unit. *Spare part* kode C nilai Q 52 unit, nilai ROP 25 unit, nilai IL rata-rata 66 unit. *Spare part* kode D nilai Q 37 unit, nilai ROP 19 unit, nilai IL rata-rata 42 unit. *Spare part* kode E nilai Q 58 unit, nilai ROP 38 unit, nilai IL rata-rata 75 unit. *Spare part* kode F nilai Q 20 unit, nilai ROP 10 unit, nilai IL rata-rata 20 unit. *Spare part* kode G nilai Q 23 unit, nilai ROP 14 unit, nilai IL rata-rata 31 unit. *Spare part* kode H nilai Q 26 unit, nilai ROP 13 unit, nilai IL rata-rata 31 unit. *Spare part* kode I nilai Q 60 unit, nilai ROP 29 unit, nilai IL rata-rata 76 unit. *Spare part* kode J nilai Q 34 unit, nilai ROP 19 unit, nilai IL rata-rata 36 unit. Hasil percobaan yang dilakukan pada 10 jenis *spare part* menunjukkan *service level* dari sistem mencapai 100% hal ini berarti tidak terjadi *shortage* sehingga sistem Q yang digunakan mampu mengurangi waktu *down time*.

Kata Kunci : Simulasi, *Compound Poisson Process*, Sistem Q, *whai if analysis*

BAB I PENDAHULUAN

I. 1 Latar Belakang

Saat ini pertumbuhan dunia industri sangatlah pesat. Persaingan antar bisnis juga semakin ketat, hal ini menyebabkan industri-industri moderen yang ada saat ini harus melakukan berbagai usaha yang dapat meningkatkan produktivitas bisnis mereka. Ditambah lagi dengan adanya krisis global yang sedang melanda perekonomian dunia saat ini, memberikan pengaruh yang sangat besar kepada para pelaku industri dan juga para pelaku bisnis di dunia. Dengan adanya persaingan yang sangat ketat menuntut para pelaku industri untuk melakukan perbaikan dan peningkatan dalam hal ketersediaan fasilitas produksi. Seperti peningkatan keandalan mesin, sistem persediaan *spare part* yang tepat dan efisien.

PT. Petrokimia Gresik adalah salah satu perusahaan BUMN yang memproduksi pupuk. PT. Petrokimia Gresik memiliki 3 pabrik yaitu unit 1, unit II dan unit III. Pada pabrik 1 yang dihasilkan adalah pupuk, urea, ZA I dan ZA II, Amonia, CO₂ dan lain-lain. Sistem produksi yang digunakan di pabrik 1 adalah *flow shop* dengan menggunakan mesin-mesin seperti *reactor*, *stripper carb*, *condenser*, *scrubber*, *hp decomposer*, *lp decomposer*, *hot water pump*, *steam condensate pump*, *sealing water pump*, dan sebagainya. Pabrik 1 terdiri dari 5 unit yaitu unit urea, ZA 1, ZA 11 Amoniak dan unit utilitas. Pabrik 1 memiliki target produksi yang berbeda setiap tahunnya, terlihat pada tabel 1.1.

Tabel 1.1
Target Produksi Pada Pabrik 1

Nama Produk	Tahun 2008 (dlm ton)	Tahun 2007 (dlm ton)	Tahun 2006 (dlm ton)	Tahun 2005 (dlm ton)
Urea	196.000	200.000	195.000	190.000
ZA I	208.000	205.000	200.000	200.000
ZA II	420.000	390.000	440.000	385.000
Amoniak	365.000	370.000	410.000	355.000

Sumber : PT. Petrokimia Gresik

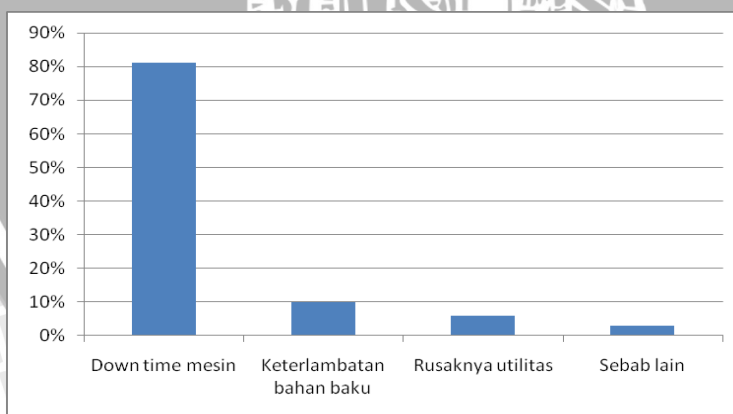
Dari target yang telah ditentukan, pabrik 1 hanya dapat memenuhi target produksi sebesar 95% seperti terlihat pada tabel 1.2 berikut ini.

Tabel 1.2 Pencapaian Produksi, waktu *Stream Day* dan waktu *Down Time*

Nama Produk	Tahun 2008(dlm ton)		<i>Stream Day</i>	<i>Down Time</i>
	Target	Capaian		
Urea	196000	192000	320	46
ZA I	208000	202000	340	25
ZA II	420000	399000	341	25
Amoniak	365000	358000	332	34

Sumber : PT. Petrokimia Gresik

Pada tabel 1.2 terlihat dari target yang telah ditentukan, PT. Petrokimia Gresik dapat merealisasikan target rata-rata sebesar 95 %. Misalnya pada tahun 2008 pada unit ZA 1 target produksi adalah sebesar 208.000 ton pertahun hanya dapat direalisasikan sebesar 202.000 ton pertahun dengan waktu *stream day* selama 340 hari dan waktu *down time* selama 25 hari, untuk produk Urea dengan waktu *stream day* selama 320 hari dan waktu *down time* mesin selama 46 hari realisasi produksi sebesar 192000 dari target produksi sebanyak 196000 ton per tahun, hal ini juga terjadi pada ZA II dan Amoniak. Banyak faktor yang mempengaruhi jumlah produksi selama 1 tahun sehingga tidak sesuai dengan target yang telah ditentukan. Faktor-faktor yang mempengaruhi jumlah produksi diantaranya adalah keterlambatan bahan baku, waktu *down time* mesin, dan rusaknya utilitas. Dari beberapa faktor yang mempengaruhi produksi pupuk di setiap tahunnya *down time* mesin adalah salah satu faktor utama yang mempengaruhi realisasi produksi setiap tahunnya. Seperti terlihat pada gambar 1.1.



Gambar 1.1 Grafik Persentase Faktor-faktor yang mempengaruhi realisasi produksi

Sistem perawatan yang digunakan akan berpengaruh terhadap keandalan mesin, sistem perawatan yang tepat dan efisien akan menambah *life time* dari mesin tersebut yang artinya akan meningkatkan produktifitas dan keandalan mesin. Sistem perawatan yang bagus didukung juga dengan adanya sistem persediaan *spare part* yang tepat dan

efisien. Dengan adanya manajemen persediaan *spare part* yang tepat akan berpengaruh terhadap lamanya waktu *down time* dan dapat meningkatkan produktifitas (B.S Dhillon, 2002:101)

PT. Petrokimia Gresik dalam manajemen *spare part* membagi *spare part* tersebut dalam 4 katagori yaitu : *asset class H*, *asset class I*, *asset class RO* (berdasarkan re-order Point), *asset class Z* (berdasarkan Persediaan Barang). Pada tipe *asset class RO*, *spare part* yang tergolong didalamnya harus selalu ada di gudang. Pada tipe RO permintaan berdasarkan pada *consumption rate* atau tingkat penggunaan barang. Bagian PGM atau Perencanaan Gudang Material akan melakukan pengadaan barang apabila *spare part* yang diminta telah mencapai *reorder point*, dan jumlah pengadaan *spare part* yang di pesan akan disesuaikan dengan rata-rata atau laju pemakaian *spare part* yang bersangkutan.

Dalam hal ini *consumption rate* atau tingkat pemakaian *spare part* mengalami fluktuasi. Pada pemakaian *spare part* misalnya Belt V dari tahun ketahun mengalami perbedaan tingkat pemakaian, seperti terlihat dalam tabel 1.3 berikut :

Tabel 1.3
Data Pemakaian *Spare part* Tipe
BELT,V -- V-BELT-SPB2000LW-RED-SEAL -- BANDO

Tahun	Waktu Pengambilan	Jumlah Pengambilan	Stok On Hand
2005	19/01/2005	10	38
	24/02/2005		
2006	30/05/2006	26	38
	11/07/2006		
	23/08/2006		
	13/10/2006		
2007	08/08/2007	10	38
2008	01/01/2008	42	38
	28/02/2008		
	03/10/2008		
	10/11/2008		

Sumber : Data PGM PT. Petrokimia Gresik

Dari tabel 1.3 diatas dapat dilihat bahwa waktu penggunaan dan jumlah pemakaian *spare part* mengalami fluktuasi. Hal ini menyebabkan PT. Petrokimia Gresik mengalami kesulitan untuk menentukan stok *spare part* yang ada di gudang. Dan juga PT. Petrokimia Gresik mengalami kesulitan untuk menentukan waktu pesan barang yang tepat. Menyebabkan nilai *stok on hand* atau *inventory level* yang tinggi.

Banyak perhitungan matematis yang dapat membantu menemukan formula yang tepat untuk menentukan jumlah dan waktu pesan yang tepat. Salah satu cara yang dapat digunakan adalah dengan menggunakan simulasi. Hal ini dikarenakan jumlah pengambilan *spare part* dan waktu pengambilan *spare part* berdistribusi sehingga dapat disimulasikan. Selain itu juga simulasi merupakan eksperimen semu terhadap suatu sistem. Dengan pendekatan *Compound Poisson Process* dimana terdapat dua distribusi dalam suatu sistem, maka pada penelitian ini akan disimulasikan Sistem Persediaan *Spare part* dengan Menggunakan Program Komputer. Dari penelitian ini akan dilakukan eksperimen untuk mengetahui nilai ROP dan Q yang tepat sehingga akan memenuhi *service level* yang diinginkan dengan nilai *inventory level* rata-rata yang rendah sehingga, dapat menurunkan *Inventory Value*.

1.2 Identifikasi Masalah

Berdasarkan latar belakang tersebut, didapatkan identifikasi masalah sebagai berikut :

1. Realisasi target baru tercapai sekitar $\pm 95\%$ yang disebabkan oleh beberapa faktor dan faktor yang paling utama adalah waktu *down time* mesin. Salah satu faktor yang mempengaruhi waktu *down time* mesin adalah sistem manajemen *spare part*.
2. *Consumption rate* yang berfluktuasi dalam setiap tahun.
3. Waktu antar pengambilan atau pemakaian *spare part* mengalami proses stokastik.
4. Kuantitas setiap pemakaian berfluktuasi atau merupakan variabel acak.

1.3 Rumusan Masalah

Berdasarkan latar belakang tersebut, maka dapat dirumuskan permasalahan sebagai berikut :

“Melakukan Perencanaan Pengendalian Sistem Persediaan *Spare Part* dengan Menggunakan Simulasi dengan Pendekatan *Compound Poisson Process*?”

1.4 Tujuan Penelitian

Berdasarkan permasalahan di atas, maka tujuan yang ingin dicapai dari penelitian ini adalah :

1. Memodelkan Sistem Persediaan *Spare part* dalam simulasi berbasis program komputer dengan pendekatan *Compound Poisson Process*.
2. Menggunakan "what if analysis" untuk menentukan ROP dan nilai Q.
3. Untuk menekan *down time* dengan pengadaan persediaan model Q.

1.5 Batasan Masalah

Mengingat luasnya ruang lingkup permasalahan, maka diperlukan adanya batasan-batasan agar pembahasan yang dilakukan lebih fokus dan tidak terlalu melebar.

Adapun batasan masalah yang dikaji adalah :

1. Pembahasan dilakukan pada *spare part* yang tergolong RO .
2. Penelitian dilakukan di Pabrik 1 PT. Petrokimia Gresik.
3. Tidak membahas mengenai jenis mesin, fungsi *spare part* dan sistem perawatan yang digunakan.
4. Pengolahan data dan penyelesaiannya menggunakan pemrograman komputer Visual Basic 6.0

1.6 Asumsi Penelitian

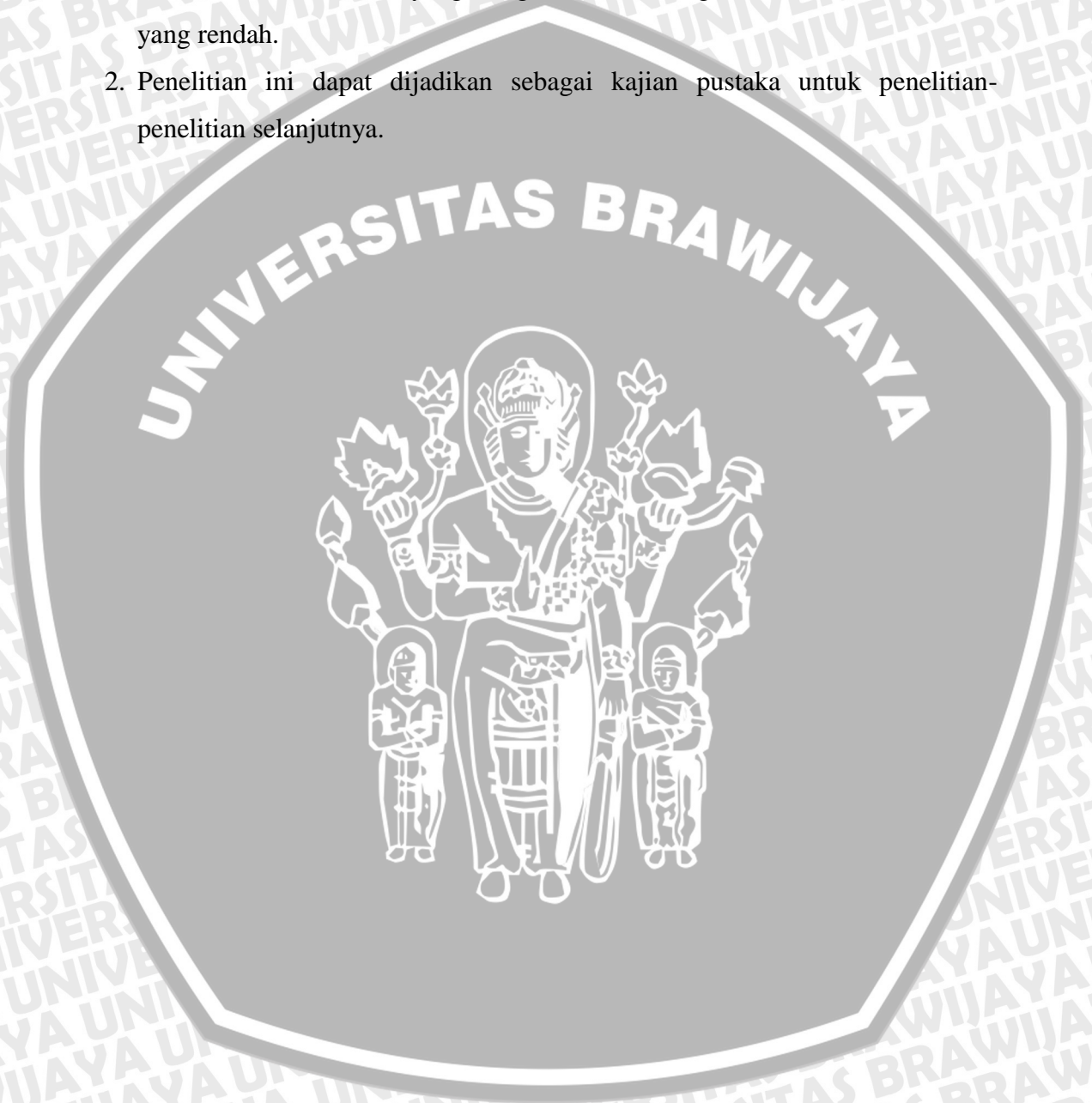
Dalam penelitian ini digunakan asumsi-asumsi sebagai berikut :

1. Waktu *lead time* untuk semua barang tetap atau sama untuk setiap barang yaitu 3 bulan.
2. Tidak memperhitungkan komponen-komponen biaya persediaan.
3. Distribusi untuk waktu antar pemakaian atau pengambilan *spare part* yang ada digudang mengikuti distribusi eksponensial.

1.7 Manfaat Penelitian

Adapun manfaat penelitian ini adalah sebagai berikut :

1. Memberikan masukan pada perusahaan nilai Q dan ROP yang tepat sehingga memenuhi *service level* yang diinginkan dan menghasilkan nilai IL rata-rata yang rendah.
2. Penelitian ini dapat dijadikan sebagai kajian pustaka untuk penelitian-penelitian selanjutnya.



BAB II TINJAUAN PUSTAKA

2.1 Manajemen *Spare Part*

Manajemen *spare part* adalah suatu kegiatan manajemen persediaan atau *inventory spare part* dimana memiliki tujuan untuk memenuhi *service level* yang diinginkan sehingga tidak terjadinya kekurangan persediaan *spare part* sehingga, dapat meningkatkan produktifitas sistem pemeliharaan mesin dan dapat mengurangi waktu *down time* mesin. Manajemen *spare part* merupakan manajemen sistem persediaan yang kompleks, hal ini disebabkan oleh beberapa faktor yaitu, permintaan akan *spare part* yang tidak dapat diprediksi, adanya *spare part* pengganti yang menyesuaikan dengan *spare part* yang asli, manajemen *inventory spare part* yang sesuai dengan *service level* yang diinginkan sulit dilakukan, dan yang terpenting adalah adanya keputusan yang harus diambil apakah akan mengadakan suatu pengadaan *spare part* dengan nilai *inventory* yang optimal dan peramalan permintaan akan *spare part* yang dapat diprediksi. Apabila mesin atau komponennya tidak diganti akan menyebabkan :

- Produksi mengalami penurunan
- Kualitas produk menurun
- Biaya tenaga kerja meningkat
- Meningkatkan biaya *maintenance*

2.1.1 Definisi Komponen

Komponen atau *part* merupakan bagian dari sistem yang mempunyai fungsi dan karakteristik tertentu yang mempengaruhi performansi suatu sistem atau peralatan yang ditempati. Perawatan untuk komponen pada dasarnya dapat dibedakan menurut sifat dari komponen tersebut yang dibagi menjadi (O'connor;2002:9):

- *Non-repairable item*

Keandalanya adalah probabilitas ketahanan selama masa hidup komponen yang diharapkan, atau periode selama hidupnya, ketika hanya satu kali kerusakan terjadi. Jika dalam sistem terdapat beberapa *part* serupa maka

kendalan sistem adalah fungsi waktu kerusakan *part* yang pertama kali terjadi.

- *Partially repairable item*

Sistem tersebut terdiri dari *part* yang bisa diperbaiki dan *part* yang tidak bisa diperbaiki. Sehingga dalam sistem keandalan dan perawatannya harus mempertimbangkan bagian-bagian yang bisa diperbaiki dan yang harus diganti.

- *Fully repairable item*

Jika terjadi kerusakan *part* bisa diperbaiki sampai beberapa kali sehingga membentuk suatu fungsi tertentu dimana *part* tidak dapat diperbaiki lagi.

Spare part atau komponen berbeda dengan *production part* karena *spare part* memiliki karakteristik sebagai berikut :

1. Permintaan akan *spare part* sebagian besar tergantung dari sistem perawatan secara *preventive maintenance* atau *predictive maintenance*, yang dapat dilihat dari nilai MTTF. Namun apabila terjadi breakdown mesin atau kerusakan mesin secara mendadak dan tidak terprediksi maka akan menyebabkan adanya inventory penyangga *spare part*.
2. Permintaan pergantian *spare part* tergantung dari *lifecycle* peralatan atau mesin dan juga mengikuti grafik kurva *bathtub*.
3. *Spare part* yang jarang digunakan akan menyebabkan sulit men-generate secara statistik untuk melakukan peramalan permintaan akan *spare part* tersebut.
4. Ketidakberadaan *spare part* akan menyebabkan terganggunya produksi dan akan berakibat langsung terhadap biaya *downtime* mesin.

2.2 Manajemen Persediaan

Setiap perusahaan, apakah perusahaan itu jasa ataupun perusahaan manufaktur, selalu memerlukan persediaan. Tanpa adanya persediaan, para pengusaha akan dihadapkan resiko bahwa perusahaanya pada suatu waktu tidak dapat memenuhi keinginan para pelangganya. Persediaan ini diadakan apabila keuntungan yang diharapkan dari persediaan tersebut terjamin kelancarannya. Persediaan adalah semua

hal-hal atau barang-barang yang sifatnya berwujud, seperti bensin, oli, serta bahan-bahan lain yang sejenis, merupakan persediaan bagi perusahaan. Persediaan adalah sesuatu yang sangat penting karena akan berpengaruh terhadap sistem produksi yang berlangsung.

Persediaan adalah sumber daya menganggur (*idle resource*) yang menunggu proses lebih lanjut tersebut adalah berupa kegiatan produksi pada sistem manufaktur, kegiatan pemasaran pada sistem distribusi ataupun kegiatan konsumsi pangan pada sistem rumah tangga. (Arman hakim, 1999:95). Selain itu persediaan merupakan salah satu unsur aktif dalam operasi perusahaan yang secara kontinu diperoleh, diubah kemudian dijual kembali.

Menurut Freddy Rangkut setiap jenis persediaan memiliki karakteristik tersendiri dan cara pengelolaan yang berbeda. Persediaan dapat dibedakan :

1. Persediaan bahan mentah (*raw material*) yaitu persediaan barang-barang berwujud, seperti besi, kayu, serta komponen-komponen lain yang digunakan dalam proses produksi.
2. Persediaan komponen – komponen rakitan (*purchased parts components*), yaitu persediaan barang-barang yang terdiri dari dari komponen-komponen yang berasal dari perusahaan lain dimana secara langsung dapat dirakit menjadi suatu produk
3. Persediaan bahan pembantu atau penolong (*supplies*), yaitu persediaan barang-barang yang diperlukan dalam proses produksi, tetapi tidak merupakan bagian atau komponen barang jadi.
4. Persediaan barang dalam proses (*work in proses*), yaitu persediaan barang-barang yang merupakan keluaran dari tiap-tiap bagian dalam proses produksi atau yang telah diolah menjadi suatu bentuk, tetapi masih perlu diproses lebih lanjut menjadi barang jadi.
5. Persediaan barang jadi (*finished goods*), yaitu persediaan barang-barang yang telah selesai diproses atau diolah dalam pabrik dan siap untuk dipakai atau dijual kepada pelanggan atau konsumen.

2.2.1 Fungsi-Fungsi Persediaan

1. Fungsi *Decoupling*

Adalah persediaan yang memungkinkan perusahaan untuk dapat memenuhi permintaan pelanggan tanpa bergantung dari supplier. Persediaan bahan mentah diadakan agar perusahaan tidak sepenuhnya tergantung pada pengadaanya dalam hal kuantitas dan waktu pengiriman. Persediaan dalam proses diadakan agar departemen-departemen dan proses-proses individual perusahaan terjaga kebebasannya. Persediaan barang jadi diperlukan untuk memenuhi permintaan konsumen yang tidak pasti. Persediaan yang diadakan untuk menghadapi fluktuasi permintaan konsumen yang tidak dapat diperkirakan atau diramalkan disebut *fluctuation stock*.

2. Fungsi *Economic Lot Sizing*

Persediaan tipe ini perlu memperhatikan penghematan-penghematan yang juga memperhatikan pemotongan dalam pembelian, biaya pengangkutan per unit jadi lebih murah dan sebagainya. Hal ini disebabkan karena perusahaan melakukan pembelian dalam kuantitas yang lebih besar, dibandingkan dengan biaya – biaya yang timbul karena besarnya persediaan.

3. Fungsi Antisipasi.

Apabila perusahaan menghadapi fluktuasi permintaan yang dapat diperkirakan dan diramalkan berdasar pengalaman atau data-data masa lalu yaitu permintaan musiman. Dalam hal ini perusahaan dapat mengadakan persediaan musiman (*seasonal inventory*). Disamping itu, perusahaan juga sering menghadapi ketidakpastian jangka waktu pengiriman dan permintaan akan barang-barang selama periode tertentu. Dalam hal ini perusahaan memerlukan persediaan ekstra yang disebut persediaan pengaman (*Safety stock inventories*).

2.3 Metode Pengendalian Persediaan

Model pengendalian persediaan dapat didefinisikan dalam 3 metode yaitu metode tradisional, metode perencanaan kebutuhan material (MRP) dan metode kanban. Dalam metode pengendalian secara tradisional menggunakan matematika dan statistika sebagai alat utama dalam memecahkan masalah. Dalam metode pengendalian tradisional berusaha mencari jawaban optimal dalam menentukan :

- jumlah pemesanan ekonomis

- jumlah pesanan kembali (*Reorder Point*)
- jumlah cadangan pengamanan (*safety stock*) yang diperlukan

Metode ini sering disebut juga persediaan secara statistik biasanya digunakan untuk mengendalikan barang yang permintaannya bersifat bebas (*dependent*) dan dikelola saling tidak bergantung. Yang dimaksud permintaan bebas adalah permintaan yang hanya dipengaruhi mekanisme pasar sehingga bebas dari fungsi operasi produksi. Sebagai contoh untuk permintaan *spare part*.

Ditinjau dari sejarah perkembangannya metode ini secara formal diperkenalkan oleh Wilson pada tahun 1929 dengan mencoba mencari jawaban 2 pertanyaan dasar, yaitu :

- Berapa jumlah barang yang harus dipesan untuk setiap kali pemesanan?
- Kapan saat pesan yang harus dilakukan?

Pengembangan formula Wilson kemudian dikembangkan pada keadaan yang terlebih realistik, terutama untuk fenomena yang bersifat probabilistik. Hal ini kemudian memunculkan dua metode yaitu :

1. **Metode P**, yang menganut aturan bahwa saat pemesanan bersifat regular mengikuti suatu periode yang tetap, sedangkan kuantitas pemesanan akan berulang-ulang
2. **Metode Q**, yang menganut aturan bahwa jumlah ukuran pemesanan (kuantitas pemesanan) selalu tetap untuk setiap kali pesan, sehingga saat pemesanan dilakukan akan bervariasi.
3. Diantara kedua metode P dan Q terdapat metode gabungan P dan Q (Arman Hakim, 1999:100)

2.3.1 Model *Service Level*

Kalau pada model di atas telah dibahas, bahwa apabila kehilangan biaya secara logika bisa dihitung secara alternative juga dapat dihitung berdasarkan perbedaan permintaan secara spesifik (misalnya 95% dengan minimum biaya). Di sini keterbatasan dari estimasi kehilangan biaya akibat kekurangan stok dapat diubah dengan menghitung tingkat pelayanan (*service level*). *Service level* merupakan persentase dari permintaan tanpa pemesanan kembali (*filled off – the – shelf*).

2.3.2 Reorder Point

Reorder Point (ROP) digunakan untuk menentukan kapan mengadakan pemesanan. ROP model terjadi apabila jumlah persediaan yang terdapat di dalam stok berkurang terus sehingga kita harus menentukan berapa banyak batas minimal tingkat persediaan yang harus dipertimbangkan sehingga tidak terjadi kekurangan persediaan. Jumlah yang diharapkan tersebut dihitung selama masa tenggang, mungkin juga dapat ditambahkan dengan safety stock yang biasanya mengacu pada probabilitas atau kemungkinan terjadinya kekurangan stok selama masa tenggang. ROP atau biasa disebut dengan batas/titik jumlah pemesanan kembali termasuk permintaan yang diinginkan atau dibutuhkan selama masa tenggang, misalnya suatu tambahan/ekstra stok.

2.4 Proses Stokastik

Stokastik merupakan pengaruh aktivitas berubah secara acak terhadap banyak keluaran yang mungkin terjadi. Keacakan aktivitas stokastik seolah-olah menyatakan bahwa aktivitas merupakan bagian dari lingkungan sistem karena keluaran yang pasti pada suatu saat tidaklah diketahui.

2.5 Jenis-jenis Distribusi

Ada banyak pola-pola distribusi baku yang sudah ada baik pola distribusi kontinyu maupun pola distribusi diskrit. (Asmungi, 2004:165) Beberapa diantaranya yang berhubungan dengan kajian skripsi ini adalah:

2.5.1 Jenis – jenis Distribusi Diskrit

1. Distribusi Bernoulli

Suatu eksperimen yang setiap trialnya hanya memberikan dua macam hasil yang mungkin. Biasanya menyatakan suatu kejadian gagal atau sukses.

2. Distribusi Binomial (n, p)

Banyak digunakan untuk menyatakan banyaknya produk cacat dalam batch, banyak item yang dibutuhkan dari gudang.

3. Distribusi Poisson

Distribusi Poisson adalah distribusi yang digunakan untuk menghitung probabilitas terjadinya kejadian menurut satuan waktu ataupun ruang. Rumus

dari dsitribusi Poisson adalah :

$$f(x) = \begin{cases} \frac{e^{-\lambda} \lambda^x}{x!}, & \text{untuk } x \in (0,1,2,\dots) \\ 0, & \text{untuk } x \text{ yang lain} \end{cases} \quad (2-1)$$

$$F(x) = \begin{cases} e^{-\lambda} \sum_{i=0}^{\lfloor x \rfloor} \lambda^i / i! \end{cases} \quad (2-2)$$

Distribusi Poisson sering dijumpai dalam permasalahan-permasalahan mengenai teori antrian atau juga distribusi Poisson erat kaitanya dengan pengelolaan, seperti kebutuhan perawatan pelayanan suatu barang pada suatu periode. Atau untuk menghitung distribusi datang armada truk ke suatu gudang setiap selang sepuluh menit, dan sebagainya. Distribusi ini disebut Poisson karena pertama kali ditemukan dan dikembangkan oleh seorang Perancis, Simoon Denis Poisson (1781 – 1840). Distribusi Poisson erat kaitanya dengn variabel acak (random variabel).

4. Distribusi Unifofrm Diskrit (i,j)

5. Distribusi Geometrik (p)

Banyak digunakan dalam pengendalian kualitas, yaitu untuk menyatakan jumlah item yang diinspeksi sebelum item cacat yang pertama ditemukan. Jumlah item dalam batch dengan ukuranya yang acak.

2.5.2 Jenis-jenis Distribusi Kontinyu

1. Distribusi Uniform (Min,Max)

Distribusi ini penting dalam upaya pembangkitan nilai bilangan acak untuk semua distribusi variabel acak dan dapat digunakan untuk waktu antar kedatangan

2. Distribusi Eksponensial (β)

Distribusi eksponensial (β) seringkali dikaitkan dengan waktu antar kedatangan pelanggan ke suatu sistem layanan yang terjadi pada laju kedatangan yang konstan. Bentuk umum fungsi kepadatan dari fungsi akumulatifnya adalah :

$$f(x) = \begin{cases} \frac{1}{\beta} e^{-x/\beta}, & \text{untuk } 0 \leq x \\ 0, & \text{untuk } x \text{ yang lain} \end{cases} \quad (2-3)$$

$$F(x) = \begin{cases} 1 - e^{-x/\beta}, & \text{untuk } 0 \leq x \\ 0, & \text{untuk } x \text{ yang lain} \end{cases} \quad (2-4)$$

3. Distribusi Gamma (α, β)

Distribusi ini acap kali cocok pada persoalan lama waktu layanan dan kedatangan, misal di loket layanan pelanggan, layanan perbaikan mesin dan lain sebagainya.

4. Distribusi Weibull (α, β)

Distribusi ini sering terkait dengan waktu layanan dan antar kedatangan seperti pada distribusi Gamma (α, β). Disamping itu juga sering terkait dengan waktu kerusakan sebuah komponen dalam masalah perawatan.

5. Distribusi Lognormal (Logmean, LogSTD)

Distribusi ini banyak digunakan pada penggambaran waktu layanan dan antar kedatangan seperti halnya pada distribusi Gamma (σ, β) dan distribusi Weibull (σ, β).

6. Distribusi Beta (Alpha1, Alpha2)

Distribusi ini memiliki kemampuan untuk menggambarkan berbagai macam bentuk data serta seringkali digunakan untuk menaksir suatu bentuk distribusi yang belum umum digunakan.

2.5.3 Estimasi Parameter Distribusi dengan *Maximum Likelihood*

Setelah hasil perkiraan fungsi distribusi probabilitasnya diperoleh, selanjutnya estimasi akan parameter fungsi tersebut. Ada beberapa estimator yang yang dapat digunakan untuk mengestimasi suatu parameter salah satunya adalah *Maximum Likelihood* (MLE). Konsep dasar dari MLE adalah mencari nilai parameter yang dapat memberikan peluang terbesar. Misalnya terdapat dapat $x_1, x_2, x_3, \dots, x_n$ yang mempunyai parameter θ . Anggap $f_{\theta}(x)$ menyatakan fungsi kepadatan yang diperkirakan, maka fungsi likelihoodnya dituliskan sebagai berikut :

$$\begin{aligned} L(\theta) &= \prod_{i=1}^n f_{\theta}(X_i) \\ &= f_{\theta}(X_1), f_{\theta}(X_2), f_{\theta}(X_3), \dots, f_{\theta}(X_n) \end{aligned} \quad (2-5)$$

Guna mendapatkan nilai θ yang mampu memaksimalkan fungsi *likelihood*, maka konsep dasar kalkulus diterapkan. Pada awalnya dicari terlebih dahulu logaritma natural dari fungsi *likelihood*, selanjutnya dicari turunan pertama fungsi *likelihood* terhadap

parameter θ . Nilai ekstrim *Likelihood* dapat diperoleh jika turunan pertamanya sama dengan nol. Adapun nilai *maksimum likelihood* didapat manakala turunan keduanya < 0 .

2.6 Compound Poisson Process

Compound Poisson Process adalah suatu model stokastik dimana $\{X(t), t \geq 0\}$. Dapat juga *compound Poisson Process* diwakili dengan, dari $t \geq 0$, dengan (Ross.Sheldon, 1996:87) :

$$X(t) = \sum_{i=1}^{N(t)} X_i \quad (2-6)$$

dimana $\{N(t), t \geq 0\}$ adalah Poisson proses, dan $\{X_i, I = 1, 2, \dots\}$ adalah variabel yang bebas atau *independent* dan pengidentifikasi distribusi variabel yang random independent atau bebas terhadap fungsi $\{N(t), t \geq 0\}$.

Sebagai contoh proses *Compound Poisson Process* adalah kejadian datangnya pembeli pada suatu toko yang mengikuti distribusi Poisson dengan rata-rata $\lambda \geq 0$. Jumlah pembeli yang datang berbeda-beda waktu kedatangannya dan juga jumlah uang yang dihabiskan oleh pembeli tersebut berbeda-beda setiap orangnya selain itu jumlah uang yang dihabiskan oleh setiap pembeli bebas atau tidak sama dengan distribusi kedatangan orang.

2.7 Bilangan Acak

Dalam suatu kejadian yang bersifat probabilistik, akan terdapat ketidakpastian. Semakin tinggi tingkat probabilistik suatu sistem maka akan semakin tinggi pula ketidakpastiannya. Untuk menggambarkan fenomena probabilistik adalah dengan melakukan pembangkitan bilangan acak (*random variabel*). Bilangan acak yang telah dibangkitkan dapat membangkitkan variabel acak yang dapat menggambarkan jenis distribusi yang terjadi, apakah mengikuti distribusi Normal, Eksponensial, Binomial, dan lain-lain. Jadi, bilangan acak menurut Asmungi (2006:105) adalah suatu bilangan sembarang yang tidak sembarangan. Bilangan acak dikatakan tidak sembarangan, karena bilangan acak harus memenuhi kriteria sebagai berikut :

1. Bilangan acak harus mempunyai distribusi serba sama (*uniform*)
Semua bilangan acak yang ada memiliki kemungkinan yang sama untuk dipilih.
Dari bilangan 0.0 sampai 1.0 memiliki peluang yang sama besar untuk dipilih

sebagai bilangan acak. Untuk bilangan acak yang serba sama dari 0.0 sampai 1.0 biasa dituliskan dengan $U(0,1)$.

2. Masing-masing bilangan acak tidak saling bergantung atau *independent*.

Bilangan acak dikatakan sebagai bilangan yang tidak saling bergantung karena terambilnya bilangan acak tidak dipengaruhi bilangan acak sebelumnya dan tidak mempengaruhi bilangan acak sesudahnya.

Bilangan acak yang telah memenuhi kedua kriteria diatas maka, bilangan acak tersebut sudah *Independent Identically Distributed (IID)*.

2.7.1 Pembangkitan Bilangan Acak

Untuk mendapatkan bilangan acak terdapat banyak cara yang dapat dilakukan salah satu yang dapat dilakukan untuk mendapatkan bilangan acak adalah dengan melemparkan dadu lalu mencatat angka yang keluar. Angka yang keluar tersebut dicatat sebagai bilangan acak. Dalam perkembangannya pembangkitan bilangan acak dapat dilakukan dengan banyak cara seperti menggunakan mesin yang bernama ERNI (*Electronic Random Number Indicator electronic*). Selain itu terdapat metode-metode pembangkitan bilangan acak yaitu :

1. Metode Midsquare

Metode ini adalah metode pembangkitan bilangan acak dengan cara mengambil nilai tengah dari hasil kuadrat suatu bilangan, selanjutnya bilangan acak berikutnya diperoleh dari hasil kuadrat bilangan tengah sebelumnya. Begitu seterusnya hingga didapatkan bilangan acak yang diinginkan.

2. Metode LCG

Metode LCG atau *Linier Congruential Generators*. Metode ini diperoleh dengan menggunakan rumus :

$$Z_i = (aZ + c)(\text{mod } m) \quad (2-7)$$

dimana : a = konstan pengali

b = konstan pergeseran

m = konstan modulus

Z_0 = bilangan awal (seed number)

3. Metode Pembangkit Bauran (Mixed LCG)

Metode ini merupakan kelanjutan dari metode LCG yaitu mencoba memperpanjang periode siklus pembangkitan bilangan acak disamping meningkatkan bilangan acak dalam periode tersebut. Metode ini adalah dengan memperpanjang periode dari persamaan sehingga didapatkan m yang semakin besar. Oleh sebab itu, harga m hendaknya sebesar 2^b dimana b adalah jumlah bit dalam satu kata (*word*) pada computer yang digunakan.

4. Pembangkit Multiplikatif (*Multiplicative Generators*)

Pembangkitan dengan metode ini adalah jika constanta c pada persamaan dipilih dengan harga nol, dan masih dimungkinkan mendapatkan periode $m-1$ jika harga m dan a diambil dengan hati – hati.

2.8 Sistem

Sistem adalah suatu kesatuan yang terdiri dari sekumpulan elemen-elemen (entitas atau subsistem yang saling berinteraksi, selaras atau bertolak belakang, bergabung, atau terurai) secara simultan dan terpadu untuk mewujudkan tujuan tertentu dengan mendapatkan masukan tertentu dengan menghasilkan keluaran pada lingkungan di luar batasan sistem. (Asmungi, 2006:19)

2.8.1 Karakteristik Sistem

Sistem memiliki beberapa karakteristik antara lain :

1. Kejadian (*event*)

Perubahan kondisi yang dialami elemen sistem

2. Aktivitas (*activity*)

Operasi yang dilakukan elemen sistem

3. Hubungan (*relationship*)

Interaksi yang menghubungkan antar elemen sistem. Hubungan ini bisa sebab akibat, penghasil-pengguna, selaras, bertolak belakang, bergabung-terurai. Dan lain-lain.

4. Antarmuka penghubung (*interface*)

Adalah media penghubung antara satu elemen dengan elemen lain, yang memungkinkan elemen-elemen sistem tersebut berinteraksi membentuk kesatuan.

5. Elemen-Elemen

Komponen bagian dari sistem yang berupa entitas atau subsistem :

- Entitas : objek yang menjadi bagian dari sistem berwujud benda atau dibendakan.
- Sub sistem : sistem yang lebih kecil sebagai bagian dari sistem yang menjadi mata sistem atau super sistem-nya.

6. Atribut

Sebutan, sifat atau karakteristik yang memiliki elemen sistem.

7. Parameter : atribut entitas yang tetap atau konstanta

Variable : atribut entitas atau interaksi antar entitas yang dapat berubah.

8. Batas sistem (*boundary*)

Setiap sistem memiliki batasan yang membatasinya dengan sistem lain atau dengan lingkungan luarnya.

9. Lingkungan luar (*enviromtment*)

Segala sesuatu di luar sistem yang dapat mempengaruhi sistem, baik menguntungkan ataupun merugikan sistem tersebut.

10. Masukan sistem (*input*)

Masukan sistem untuk diproses menghasilkan keluaran atau agar sistem dapat beroperasi.

11. Pengganggu (*disturbance/noise*)

Masukan sistem yang tidak dapat dikendalikan, namun dapat mempengaruhi sistem.

12. Keluaran sistem (*output*)

Hasil yang telah diolah dalam sistem yang berikutnya kembali menjadi masukan sistem.

13. Umpan Balik (*feddback*)

Hasil yang telah diolah dalam sistem menjadi sesuatu yang bernilai, serta sisa.

14. Ukuran performansi sistem

Merupakan kumpulan indikator yang menggambarkan keadaan sistem pada waktu tertentu

- *Transient state* : kondisi sistem yang dipengaruhi oleh perubahan waktu.
- *Steady state* : kondisi sistem yang telah tidak dipengaruhi oleh perubahan waktu (stabil).

15. Proses pengolahan (*transformation process*)

Merupakan bagian yang mengolah atau merubah masukan sistem menjadi keluaran sesuai tujuan sistem.

16. Perilaku sistem (*behaviour*)

Perubahan pada sistem karena aktivitas elemen-elemen sistem dalam proses pengolahan.

2.8.2 Klasifikasi Sistem

Klasifikasi dari sistem antara lain :

1. Sistem terbuka versus tertutup :

1. Sistem terbuka : sistem yang berinteraksi dengan lingkungannya
2. Sistem tertutup : sistem yang tidak berinteraksi dengan lingkungannya.

2. Sistem deterministik versus sistem probabilistik :

1. Sistem deterministik : sistem dengan kondisi (*state*) yang dapat diperkirakan nilainya, karena semua nilai variabel penyusunnya tidak ada yang bersifat probabilistik atau keseluruhan variable bersifat deterministik (tetap atau membentuk fungsi tertentu).
2. Sistem probabilistik : sistem dengan kondisi (*state*) yang nilainya tidak dapat diperkirakan nilainya secara pasti dan memungkinkan terjadinya penyimpangan, karena nilai variabel penyusunnya terfluktuasi tanpa mengikuti fungsi tertentu namun masih membentuk distribusi probabilitas tertentu.

3. Sistem Diskrit versus sistem kontinu:

1. Perubahan kontinu dalam waktu kontinu
2. Perubahan kontinu dalam waktu diskrit
3. Perubahan diskrit dalam waktu kontinu
4. Perubahan diskrit dalam waktu diskrit

4. Sistem statis versus sistem dinamis :

- Sistem statis : sistem yang memiliki kondisi (*state*) tetapi dan tidak terpengaruh oleh perubahan waktu. Misalnya sistem-sistem yang dapat diselesaikan dengan pemodelan matematis sederhana (bukan integral maupun diferensial terhadap waktu).
- Sistem dinamis : sistem yang memiliki kondisi (*state*) yang berubah terpengaruh oleh perubahan waktu. Misalnya sistem-sistem yang dimodelkan secara matematis dan diselesaikan oleh simulasi.

5. Sistem *stationery* versus sistem *nonstationery* :

- Sistem *stationery* : sistem yang mengalami perubahan konstan.
- Sistem *nonstationery* : sistem yang mengalami perubahan tidak tetap.

2.9 Model

Model adalah suatu representasi atau formalisasi suatu sistem nyata dalam bahasa, simbol atau bentuk tertentu yang mewakili karakteristik dan oleh perilaku sistem sesuai sudut pandang dan dasar pandangan dan dasar pengetahuan pemodel dalam tujuan mempelajari sistem.

2.9.1 Latar Belakang Model

Latar Belakang perlunya sebuah model dibuat untuk mempresentasikan sistem nyata adalah :

1. Sistem nyata terlalu kompleks
2. Sistem nyata masih dalam bentuk hipotesis
3. Tidak mungkin untuk melakukan eksperimen secara langsung
4. Model merupakan representasi ideal dari suatu sistem untuk menjelaskan perilaku sistem karena mampu menunjukkan elemen-elemen terpenting dari sistem.
5. Modifikasi dan verifikasi terhadap model lebih mudah dan murah dibandingkan bila diterapkan pada sistem nyata.

2.9.2 Batasan dan Asumsi Model

1. Batasan : memberikan rambu-rambu lingkup sistem yang dimodelkan dan cara pandang pemodel pada sistem.
2. Asumsi : menunjukkan elemen-elemen sistem yang diabaikan pemodel karena dianggap tidak krusial dengan kepentingan dalam mempelajari sistem.

2.9.3 Prinsip Pemodelan

1. Elaborasi : model dikembangkan mulai dari yang sederhana kemudian secara bertahap dielaborasi hingga diperoleh model yang lebih representative. Penggunaan asumsi dalam model harus memenuhi persyaratan konsisten, independensi, ekuivalensi dan relevansi.
2. Iterative : mengembangkan model tidak dapat dilakukan secara mekanistik dan linear. Melainkan dilakukan pengembangan melalui pengulangan dan peninjauan ulang secara iterative. Proses iterative diawali dari model awal berdasarkan dugaan atau hipotesis, kemudian dikembangkan secara bertahap, hingga dicapai tingkat kompleksitas representative yang diharapkan.
3. Sinetif : mengembangkan model dapat dilakukan secara sinetik, yaitu metode pendekatan sistem dengan menganalogkan permasalahan yang ada. Dengan begitu model lebih mudah dikenal dan dipahami, karena mengadopsi dari prinsip-prinsip, hukum, teori, aksioma, dan dalil yang sudah ada dan memiliki kemiripan identik.

2.10 Simulasi

Simulasi dapat diartikan sebagai suatu sistem yang digunakan untuk memecahkan atau menguraikan persoalan-persoalan dalam kehidupan nyata yang penuh dengan ketidakpastian dengan tidak atau menggunakan model atau metode dan lebih ditekankan pada pemakaian untuk mendapatkan solusinya.

Pada pendekatan simulasi, untuk menyelesaikan berbagai persoalan yang rumit akan lebih mudah dilakukan bila dimulai dengan membangun model percobaan dari

suatu sistem. Untuk melakukannya kita perlu memperhatikan tiga unsur penting dalam pemodelan simulasi, yaitu sistem, entities dan attributes.

2.10.1 Keuntungan Simulasi :

1. *Compress Time* (menghemat waktu)

Kemampuan di dalam menghemat waktu ini dapat dilihat dari pekerjaan yang bila dikerjakan akan memakan waktu ini dapat dilihat dari pekerjaan yang bila dikerjakan akan memakan waktu tahunan tetapi kemudian dapat disimulasikan hanya dalam beberapa menit atau detik saja.

2. *Expand Time* (Dapat Melebar-luaskan waktu)

Simulasi dapat digunakan untuk menunjukkan perubahan struktur dari suatu sistem nyata (*real sistem*) yang sebenarnya tidak dapat diteliti pada waktu yang seharusnya (*real-time*). Dengan demikian simulasi dapat mengubah *real sistem* hanya dengan memasukkan sedikit data.

3. *Control Source of Variation* (dapat mengawasi sumber-sumber yang bervariasi)

Kemampuan pengawasan dalam simulasi ini tampak terutama apabila analisis statistic digunakan untuk meninjau hubungan antara variabel bebas dengan variable terkait yang merupakan factor-faktor yang akan dibentuk dalam percobaan.

4. *Error in Measurement Correction* (mengoreksi kesalahan-kesalahan perhitungan)

Dalam simulasi komputer jarang ditemukan kesalahan perhitungan terutama bila angka-angka diambil dari komputer secara teratur dan bebas.

5. *Stop Simulation and Restart* (dapat dihentikan dan Dijalankan Kembali)

Dengan simulasi komputer, percobaan dapat dilakukan setiap saat dan dapat diulang-ulang. Pengulangan dilakukan terutama untuk mengubah berbagai komponen dan variabelnya, seperti pada perubahan

parameter, perubahan kondisi operasinya, ataupun dengan memperbanyak output.

6. *Easy to Replicate* (mudah diperbanyak)

Simulasi komputer dapat dihentikan untuk kepentingan peninjauan ataupun pencatatan semua keadaan yang relevan tanpa berakibat fatal terhadap program simulasi tersebut.

2.11 Pengujian Statistik

2.11.1. Pengujian *Independent Sample t test*

Pengujian ini dilakukan untuk membandingkan apakah suatu rata-rata kelompok X memiliki nilai yang sama dengan rata-rata kelompok Y (Hasan,2005:153)

Langkah pengujian pada dasarnya sama dengan uji beda satu sampel, kecuali pada kriteria pengujian dan perhitungan nilai *t* hitung.

1. Menyusun H_0 dan H_1 (ada 3 macam hipotesis)

$$\begin{aligned} H_0 : \mu_1 &= \mu_2 \\ H_1 : \mu_1 &\neq \mu_2 \end{aligned} \tag{2-8}$$

2. Menentukan *level of significance* (α)
3. Menentukan peraturan-peraturan pengujiannya / kriterianya

$$\begin{aligned} H_0 \text{ diterima apabila : } & -t_{\alpha/2;n-1} \leq t \leq t_{\alpha/2;n-1} \\ H_0 \text{ ditolak apabila : } & t > t_{\alpha/2;n-1} \quad \text{atau} \quad t < -t_{\alpha/2;n-1} \end{aligned} \tag{2-9}$$

4. Dari sampel yang diambil, dihitung nilai *t* dengan rumus :

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\left\{ \frac{n-1 S_1^2 + n-1 S_2^2}{n_1 + n_2 - 2} \right\} \left\{ \frac{1}{n_1} + \frac{1}{n_2} \right\}}} \tag{2-10}$$

dengan : \bar{X}_1 adalah rata-rata sampel satu

\bar{X}_2 adalah rata-rata sampel dua

S_1 dan S_2 adalah standar deviasi sampel

n_1 dan n_2 banyaknya titik sampel

5. Bandingkan langkah ke 3 dan ke 4 untuk diambil kesimpulan

2.11.2 Pengujian ANOVA



Analisis Varians atau ANOVA adalah analisa yang dilakukan untuk mengetahui perbedaan antara nilai tengah perlakuan dalam suatu percobaan. Misalkan ada k buah perlakuan dimana terdapat n_i unit eksperimen untuk perlakuan ke i ($i = 1,2,\dots,k$). Jika data pengamatan dinyatakan dengan Y_{ij} ($i = 1,2,\dots,k$ dan $j = 1,2,\dots,n_i$), Y_{ij} berarti nilai pengamatan dari unsur eksperimen ke j krena perlakuan ke i. (Sudjana, 1997:16). Dalam analisis varians atau ANOVA, perhitungan yang pertama dilakukan adalah menghitung:

1. Jumlah pengamatan tiap perlakuan

$$J_i = \sum_{j=1}^{n_i} Y_{ij} \quad (2-11)$$

2. Jumlah seluruh nilai pengamatan

$$J = \sum_{i=1}^k J_i \quad (2-12)$$

3. Rata-rata pengamatan untuk tiap perlakuan

$$\bar{Y}_i = J_i / n_i \quad (2-13)$$

4. Rata-rata nilai seluruh pengamatan

$$\bar{Y} = J / \sum_{i=1}^k n_i \quad (2-14)$$

5. Nilai JK (Jumlah Kuadrat) semua pengamatan (

$$= \sum_{i=1}^k \sum_{j=1}^{n_i} Y_{ij}^2 \quad (2-15)$$

6. Jumlah kuadrat-kuadrat (JKK) untuk rata-rata (R_y)

$$= J^2 / \sum_{i=1}^k n_i \quad (2-16)$$

7. Jumlah kuadrat (JK) antar perlakuan (P_y)

$$= \sum_{i=1}^k n_i (\bar{Y}_i - \bar{Y})^2 \quad (2-6)$$

$$= \sum_{i=1}^k (J_i^2 / n_i) - R_y \quad (2-17)$$

8. Jumlah kuadrat-kuadrat (JK) kekeliruan eksperimen (E_y)

$$= \sum_{i=1}^k \sum_{j=1}^{n_i} (\bar{Y}_{ij} - \bar{Y}_i)^2 \quad (2-6)$$

$$= \sum Y^2 - R_y - P_y \quad (2-18)$$

Adapun langkah-langkah Pengujian AOVA adalah sebagai berikut :

Langkah- langkah pengujian ANOVA

1. Menentukan Formulasi Hipotesis

$$H_0 : \mu_1 = \mu_2 = \dots = \mu_k$$

$$H_1 : \text{Ada } \mu_i \text{ yang tidak sama} \quad (2-19)$$

2. Menentukan *significance level* beserta *F* tabel
3. Menentukan kriteria pengujian
 H_0 diterima apabila $F_0 \leq F_{\alpha;v1;v2}$
 H_0 ditolak apabila $F_0 > F_{\alpha;v1;v2}$
4. Membuat analisis variannya dalam bentuk tabel ANOVA

Tabel 2.1 Tabel ANOVA

Sumber Variasi	Derajat Kebebasan (dk)	Jumlah Kuadrat-kuadrat (JK)	Kuadrat Tunggal
Rata-rata	1	R_y	$R = R_y$
Antar Perlakuan	$k-1$	P_y	$P = P_y / (k-1)$
Kekeliruan Ekperimen(Dalam Perlakuan)	$\sum_{i=1}^k (n_i - 1)$	E_y	$E = E_y / \sum (n_i - 1)$
Jumlah Total	$\sum_{i=1}^k n_i$	$\sum Y^2$	-

Sumber : Sudjana, 1989:18

5. Kesimpulan

Membandingkan Langkah ke-3 dan ke-4

2.14 Metode “*what if analysis*”

Analisa “*what if*” banyak digunakan pada studi ekonomis yang merupakan tindak lanjut dari pada evaluasi ekonomis, untuk menguji sensitivitas parameter suatu perencanaan terhadap keadaan yang akan datang, dimana dengan adanya perubahan parameter akan mempengaruhi hasil proposal yang telah direncanakan. Hasil analisa dari pengujian parameter disajikan dalam bentuk grafik sensitivitas yang menunjukkan pengaruh dari pada perubahan parameter (biasanya dalam prosentasi) terhadap hasil akhir dari pada proposal studi ekonomis. Penampilan grafik merupakan hasil konsolidasi data analisa yang mudah digunakan dan dimengerti.

Metode “*what if analysis*” banyak digunakan untuk mengambil keputusan dengan memasukan berbagai kemungkinan ketidakpastian dalam pengambilan keputusan. Metode ini merupakan metode sensitivitas. Dalam simulasi metode *what if analysis* dapat digunakan untuk melakukan percobaan atau eksperimen semu. Dengan memasukan nilai-nilai parameter yang ada dalam sistem nantinya akan dilihat *output* yang dihasilkan. Dengan menggunakan *what if analysis* akan dapat dipilih sistem yang memberikan nilai optimal.

2.13 Visual Basic 6.0

Visual basic 6.0 merupakan salah satu bahasa pemrograman yang bekerja dalam lingkup Microsoft Windows. Bahasa pemrograman adalah perintah-perintah yang dimengerti komputer untuk melakukan tugas-tugas tertentu. Bahasa pemrograman *visual basic* dikembangkan oleh Microsoft sejak tahun 1991, yang merupakan pengembangan dari bahasa pemrograman versi pendahulunya yaitu bahasa pemrograman BASIC (*Beginner's All-purpose Symbolic instruction Code*) yang dikembangkan pada era 1950-an. Pemrograman Visual Basic menggunakan bahasa yang sederhana dan mudah dalam pembuatan *script* (simple scripting language) untuk *graphic user interfacenya*. Dalam *visual basic* terdapat dua jenis teknik pemrograman yaitu Pemrograman Visual dan *Object Oriented Programing* (OOP).(Halvorson, 2001:8)

Beberapa kemampuan yang dimiliki *Visual Basic* 6.0 antara lain sebagai berikut :

- a. Membuat program aplikasi komputer berbasis Wndows.
- b. Membuat *object-object* pembantu program, misalnya file Help, kontrol AciveX dan sebagainya.
- c. Kamampuan pemrogaraman internet dengan DHTML (*Dynamic HyperText Mark language*) dan beberapa penambahan fitur database dan multimedia yang semakin baik.
- d. Menguji program dan menghasilkan program akhir berekstensi EXE yang bisa dijalankan tanpa harus membuka *Visual Basic* lagi.

Dalam pengembangan aplikasi, *Visual Basic* menggunakan teknik pemrograman Visual untuk merancang *user interface* dalam bentuk form, sedangkan untuk kodingnya menggunakan bahasa Basic yang cenderung mudah dipelajari. Kata “Visual”bararti menunjukkan cara yang digunakan untuk membuat *graphical user interface* (GUI), dimana dengan cara ini ketika membuat suatu program kita tidak lagi menuliskan instruksi – instruksi pemrograman dalam kode-kode baris, tetapi secara mudah kita dapat membuat *user interface* dengan kontrol “*drawing*” seperti Textbox, Label dan Commandbutton dan lain sebagainya, ke dalam sebuah Form. Dan kita juga bisa mengatur properti untuk Form dan kontrol-kontrol yang ada didalamnya, misal memberikan *caption, color, size, font*, dan lain sebagainya. Selanjutnya menuliskan kode-kode program bahasa Basic yang diletakkan pada masing-masing *object* atau komponen yang ada pada form.

Dalam *visual basic* juga terdapat teknik pemrograman yang berbasis *object oriented programming* (OOP). Teknik pemrograman ini adalah berbasis *object*. Maksudnya adalah penggunaan sudut pandang *object* sebagai sebuah komponen dalam sebuah program yang dimodelkan dengan benda-benda dalam dunia nyata. Contoh *object* yang ada di sekitar kita adalah hewan, mobil, televisi, sepeda, dan masih banyak lainnya. *Object* yang kita lihat tentunya memiliki hal atau karakteristik yang melekat pada dirinya sendiri seperti ukuran, warna, tinggi, dan bentuk yang disebut *attribute*. Selain itu, *object* yang kita kenal juga mempunyai tingkah laku dan kemampuan untuk melakukan suatu tugas, berasal dari dalam dirinya sendiri, yang disebut *behavior* atau *method* seperti mobil berjalan ke depan, hewan tidur, televisi dinyalakan, dan lampu dimatikan. Pembuatan *object* dalam OOP ini sangat erat hubungannya dengan pembuatan *class*.

Dengan *Visual basic* 6.0 ini, kita akan lebih mudah dalam membuat suatu aplikasi komputer karena telah tersedia beberapa pilihan perintah yang dapat digunakan sesuai kebutuhan.

2.13.1 *Integrated Development Environment* (IDE)

IDE (*Integrated Development Environment*) adalah program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak atau aplikasi. Tujuan dari IDE adalah untuk menyediakan semua utilitas yang diperlukan dalam membangun sebuah aplikasi, menulis program, menjalankan program, dan menghasilkan sebuah *executable file*.

Sebuah IDE, atau secara bebas dapat diterjemahkan sebagai Lingkungan Pengembangan Terpadu, setidaknya memiliki fasilitas:

- a. Editor, yaitu fasilitas untuk menuliskan kode sumber dari perangkat lunak.
- b. *Compiler*, yaitu fasilitas untuk mengecek sintaks dari kode sumber kemudian mengubah dalam bentuk binari yang sesuai dengan bahasa mesin.
- c. *Linker*, yaitu fasilitas untuk menyatukan data binari yang beberapa kode sumber yang dihasilkan *compiler* sehingga data-data binari tersebut menjadi satu kesatuan dan menjadi suatu program komputer yang siap dieksekusi.

- d. *Debugger*, yaitu fasilitas untuk mengetes jalannya program, untuk mencari *bug*/kesalahan yang terdapat dalam program.

Sampai tahap tertentu IDE modern dapat membantu memberikan saran yang mempercepat penulisan. Pada saat penulisan kode, IDE juga dapat menunjukkan bagian-bagian yang jelas mengandung kesalahan atau keraguan.

2.13.2 Menjalankan IDE

Salah satu cara untuk mengaktifkan IDE visual Basic adalah menjalankannya dari menu *Start*, lalu arahkan mouse pada group *All Programs*, dan pilih *Microsoft Visual Basic 6.0*. Sesaat anda aktif di IDE Visual Basic 6.0, maka akan muncul pada layar kotak dialog untuk memilih jenis *project* yang akan anda buat sebagaimana yang ditunjukkan oleh Gambar 1.1, misal pilih *new project Standard EXE*. *Project* adalah sekumpulan form, modul, fungsi, data dan laporan yang digunakan dalam suatu aplikasi.



Gambar 2.1
Kotak Dialog *New Project*

Dalam *Visual Basic 6.0* tersedia 13 jenis *project* yang bisa dibuat seperti terlihat pada gambar 1.1 di atas. Ada beberapa *project* yang biasa digunakan oleh banyak pengguna *Visual Basic*, antara lain:

- a. ***Standard EXE***: *Project* standar dalam *Visual Basic* dengan komponen-komponen standar. Jenis *project* ini sangat sederhana, tetapi memiliki

keunggulan bahwa semua komponennya dapat diakui oleh semua unit komputer dan semua user meskipun bukan administrator.

- b. **ActiveX EXE**: *Project* ini adalah *project* ActiveX berisi komponen-komponen kemampuan untuk berinteraksi dengan semua aplikasi di sistem operasi windows.
- c. **ActiveX DLL** : *Project* ini menghasilkan sebuah aplikasi library yang selanjutnya dapat digunakan oleh semua aplikasi di sistem operasi windows.
- d. **ActiveX Control** : *Project* ini menghasilkan komponen-komponen baru untuk aplikasi *Visual Basic* yang lain
- e. **VB Application Wizard** : *Project* ini memandu pengguna untuk membuat aplikasi secara mudah tanpa harus pusing-pusing dengan perintah-perintah pemrograman.
- f. **Addin** : *Project* seperti Standard EXE tetapi dengan berbagai macam komponen tambahan yang memungkinkan kebebasan kreasi dari pengguna.
- g. **Data Project** : *Project* ini melengkapi komponennya dengan komponen-komponen database. Sehingga bisa dikatakan *project* ini memang disediakan untuk keperluan pembuatan aplikasi database.
- h. **DHTML Application** : *Project* ini digunakan untuk membuat aplikasi internet pada sisi client (*client side*) dengan fungsi-fungsi DHTML.
- i. **IIS Application** : *Project* ini menghasilkan aplikasi internet pada sisi server (*server side*) dengan komponen-komponen CGI (*Common Gateway Interface*).

2.13.3 Jendela IDE

IDE Visual Basic 6 menggunakan model MDI (*Multiple Document Interface*). Jendela IDE mempunyai sejumlah menu, toolbar, dan window seperti yang terlihat pada gambar 1.2 Berikut ini akan dijelaskan pada masing-masing item:



Gambar 2.2
Jendela IDE

- a. *Menu Bar*, digunakan untuk memilih tugas-tugas tertentu seperti menyimpan *project*, membuka *project*, dll.
- b. *Main Toolbar*, digunakan untuk melakukan tugas-tugas tertentu dengan cepat.
- c. *Jendela Project*, jendela ini berisi gambaran dari semua modul yang terdapat dalam aplikasi anda. Anda dapat menggunakan *icon Toggle Folders* untuk menampilkan modul-modul dalam jendela tersebut secara di group atau berurut berdasarkan nama. Anda dapat menggunakan *Ctrl+R* untuk menampilkan jendela *project*, ataupun menggunakan *icon Project Explorer*.
- d. *Jendela Form Designer*, jendela ini merupakan tempat anda untuk merancang *user interface* dari aplikasi anda. Jadi jendela ini menyerupai kanvas bagi seorang pelukis.
- e. *Jendela Toolbox*, jendela ini berisi komponen-komponen yang dapat anda gunakan untuk mengembangkan *user interface*.
- f. *Jendela Code*, merupakan tempat bagi anda untuk menulis koding. Anda dapat menampilkan jendela ini dengan menggunakan kombinasi *Shift-F7*.

- g. Jendela *Properties*, merupakan daftar properti-properti *object* yang sedang terpilih. Sebagai contohnya anda dapat mengubah warna tulisan (*foreground*) dan warna latar belakang (*background*). Anda dapat menggunakan F4 untuk menampilkan jendela properti.
- h. Jendela *Color Palette*, adalah fasilitas cepat untuk mengubah warna suatu *object*.
- i. Jendela *Form Layout*, akan menunjukkan bagaimana form bersangkutan ditampilkan ketika runtime.
- j. *Immediate window*, digunakan untuk memasukkan ekspresi untuk melihat hasilnya dengan menggunakan perintah “print” atau “?”. Jendela ini biasanya digunakan bersama *watch window* pada saat sebuah program di-debug. Short-cut untuk jendela ini adalah ctrl-G
- k. *Object Browser* digunakan untuk menyelusuri *external libraries* sehingga anda dapat mempelajari *object-object* pada *property*, kejadian (*events*), dan *method* yang dimilikinya Short-cut untuk jendela ini adalah F2
- l. *Local window* akan aktif hanya pada saat program dijalankan, berisi nilai dari sejumlah *variable* yang bersifat *local* pada sebuah prosedur atau *module*.
- m. *Watch window* digunakan untuk memonitor nilai dari suatu *variable* baik bersifat *local* maupun *global*.
- n. *Call stack window* hanya ditampilkan jika suatu program yang dijalankan dihentikan untuk sementara (*break*) dan menekan ctrl-L. Menunjukkan semua prosedur yang menunggu prosedur aktif selesai. Jendela ini penting untuk proses *debugging* untuk mengetahui jalur eksekusi program hingga sampai situasi sekarang.

Anda bisa mencoba menutup jendela-jendela tersebut dengan klik di *Close button* dimasing-masing jendela (*window*) dan dengan mudah anda bisa menampilkan lagi melalui *Menu View*.

2.13.4 Toolbox

Toolbox berisi komponen-komponen yang bisa digunakan oleh suatu *project* aktif, artinya isi komponen dalam toolbox sangat tergantung pada jenis *project* yang dibangun. Komponen standar dalam *toolbox* dapat dilihat pada gambar 1.3



Gambar 2.3
ToolBox pada VB 6.0

Berikut ini kontrol – kontrol yang ada pada toolbox :

- PictureBox* adalah kontrol yang digunakan untuk menampilkan image dengan format: BMP, DIB, dll.
- Label* adalah kontrol yang digunakan untuk menampilkan teks yang tidak dapat diperbaiki oleh pemakai.
- TextBox* adalah kontrol yang mengandung string yang dapat diperbaiki oleh pemakai, dapat berupa satu baris tunggal, atau banyak baris.
- Frame* adalah kontrol yang digunakan sebagai kontainer bagi kontrol lainnya.
- CommandButton* merupakan kontrol hampir ditemukan pada setiap form, dan digunakan untuk membangkitkan *event* proses tertentu ketika pemakai melakukan klik padanya.
- CheckBox* digunakan untuk pilihan yang isinya bernilai yes/no, true/false.
- OptionButton*, sering digunakan lebih dari satu sebagai pilihan terhadap beberapa option yang hanya dapat dipilih satu.
- ListBox*, mengandung sejumlah item, dan user dapat memilih lebih dari satu (bergantung pada *property MultiSelect*).

- i. *ComboBox*, merupakan kombinasi dari *TextBox* dan suatu *ListBox* dimana memasukkan data dapat dilakukan dengan pengetikkan maupun pemilihan.
- j. *HScrollBar* dan *VScrollBar*, digunakan untuk membentuk *scrollbar* berdiri sendiri.
- k. *Timer*, digunakan untuk proses background yang diaktifkan berdasarkan interval waktu tertentu. Merupakan kontrol non-visual.
- l. *DriveListBox*, *DirListBox*, dan *FileListBox* digunakan untuk membentuk dialog box yang berkaitan dengan file.
- m. *Shape* dan *Line*, digunakan untuk menampilkan bentuk seperti garis, persegi, bulatan, oval.
- n. *Image* berfungsi menyerupai *image box*, tetapi tidak dapat digunakan sebagai kontainer bagi kontrol lainnya. Data digunakan untuk data binding
- o. OLE dapat digunakan sebagai tempat bagi program eksternal seperti *Microsoft Excel*, *Word*, dll.



BAB III METODOLOGI PENELITIAN

3.1. Metode Penelitian

Metode penelitian yang digunakan adalah metode eksperimen semu dengan menggunakan simulasi sistem yang terjadi.

3.2. Obyek penelitian

Pada penulisan skripsi ini, tempat pengambilan data dilakukan di PT. Petrokimia Gresik Pabrik I pada bulan Februari dan Maret tahun 2009.

3.3. Jenis dan Sumber Data

Menurut Hasan (2002:38), pengumpulan data adalah pencatatan-pencatatan atau keterangan-keterangan sebagian atau keseluruhan dari elemen populasi yang akan menunjang atau mendukung penelitian. Adapun cara pengumpulan data dalam penelitian ini dilakukan dengan cara:

a. Studi Pustaka

Studi pustaka dilakukan untuk memperoleh rumusan kerangka teoritis dari masalah yang diteliti, yang bisa diperoleh dari buku-buku literatur, jurnal-jurnal ilmiah dan lain-lain.

b. Studi lapangan

Merupakan metode pengumpulan data yang dilakukan secara langsung pada obyek penelitian. Pengumpulan data tersebut bisa dilakukan dengan jalan:

- *Observasi*, merupakan cara pengumpulan data yang dilakukan dengan jalan mengamati langsung aktifitas obyek yang diteliti.
- *Interview* (wawancara), merupakan cara pengumpulan data dengan jalan mengadakan wawancara langsung dengan pihak-pihak perusahaan yang berhubungan dengan penelitian.
- *Dokumentasi*, merupakan cara pengumpulan data dengan mengambil data-data perusahaan yang berupa laporan, catatan-catatan atau arsip yang sudah ada.

3.4 Metode Alur Penelitian

3.4.1 Survey Lapangan

Dalam Tahap ini peneliti melakukan *survey* langsung dilapangan yaitu di PT. Petrokimia Gresik untuk mencari dan menentukan topik penelitian yang sesuai dengan bakat, minat dan konsentrasi yang diambil.

3.4.2 Studi Literatur

Langkah berikutnya adalah dengan melakukan studi pustaka dengan mengumpulkan dan mempelajari buku-buku, literatur, jurnal dan *website* serta sumber lain yang berhubungan dengan permasalahan yang sedang dibahas.

3.4.3 Identifikasi Masalah

Setelah peneliti mengetahui keadaan yang terjadi dilapangan selanjutnya peneliti mengidentifikasi permasalahan yang terjadi. Identifikasi masalah dapat dilakukan dengan melakukan *brainstroming* dengan pihak-pihak yang terkait dengan objek penelitian.

3.4.4. Penetapan Tujuan Penelitian

Dalam tahap ini peneliti menentukan tujuan penelitian sehingga nantinya penelitian yang dilakukan tidak keluar dari permasalahan yang dibahas. Dan juga penetapan tujuan penelitian adalah untuk menetapkan target dari penelitian yang dilakukan.

3.4.5 Pengumpulan Data

Adapun data-data yang dibutuhkan dalam penelitian ini meliputi :

a. Data Kualitatif

1. Data kebijakan persediaan perusahaan *order quantity*, *reorder point*, dan *service level*.
2. Data jenis-jenis *spare part*

b. Data Kuantitatif

1. Data jumlah permintaan suku cadang
2. Data waktu permintaan atau *interval time*

3. Data nilai *Stok On Hand* (IL), nilai Q, nilai ROP (*reorder point*), *service level* dan *inventory value*.

3.4.5.1 Uji Distribusi

Pada tahap ini dilakukan pengujian distribusi pada dua variabel yang ada yaitu :

1. Variabel Waktu antar pengambilan *spare part*
Yaitu, merupakan selang waktu yang terjadi antar pengambilan *spare part* .
Dimana distribusi yang ada diasumsikan mengikuti distribusi eksponensial.
2. Variabel Jumlah pengambilan *spare part*
Yaitu, merupakan jumlah setiap pengambilan *spare part*. Dimana uji distribusi yang dilakukan adalah uji distribusi diskrit.

3.4.5.2 Penentuan Parameter Distribusi

Setelah hasil perkiraan fungsi probabilitas diperoleh, selanjutnya estimasi akan parameter fungsi tersebut harus dilakukan adapun parameter-parameter sesuai dengan jenis distribusinya.

3.4.6 Pembuatan Model Matematis

Pada tahap ini peneliti membuat model matematis yaitu membuat model sesuai dengan sistem persediaan yang terjadi di lapangan yaitu pada PT.Petrokimia Gresik.

3.4.7 Perancangan Program Simulasi

Pada penelitian ini perancangan program simulasi menggunakan program computer yaitu Visual Basic 6.0.

3.4.8 Verifikasi Model

Verifikasi Model merupakan tahap dalam simulasi untuk memeriksa atau melacak akan kebenaran program apakah sudah bebas dari jenis kesalahan *syntax error*, *design error*, dan *run time error*.

3.4.9 Keluaran Simulasi

Setelah program simulasi dibuat dan dapat dijalankan dengan waktu simulasi yang telah ditentukan, maka program yang telah dibuat akan memberikan hasil keluaran dari simulasi yang telah dijalankan.

3.4.10 Validasi Model

Pada tahap validasi, peneliti menguji apakah hasil keluaran simulasi telah sesuai dengan keadaan nyata. Dalam penelitian ini menggunakan uji independent T-Test untuk menguji nilai service level dan inventory level rata-rata sistem nyata dengan nilai service level dan inventory level rata-rata hasil keluaran simulasi.

3.4.11 Analisa Hasil Simulasi

Pada tahap analisa hasil simulasi, dilakukan analisa terhadap hasil simulasi yang telah dijalankan. Selanjutnya terdapat dua macam aktifitas yang dilakukan setelah menganalisa hasil simulasi yaitu:

3.4.11.1 Rancangan Percobaan dengan “*what if analysis*”

Pada tahap ini peneliti melakukan rancangan percobaan berdasarkan “*what if analysis*”. Pada tahap rancangan percobaan peneliti melakukan eksperimen dengan memasukan nilai parameter yaitu ROP dan Q yang nantinya akan dilihat hasilnya terhadap *service level* dan nilai *inventory level* rata-rata.

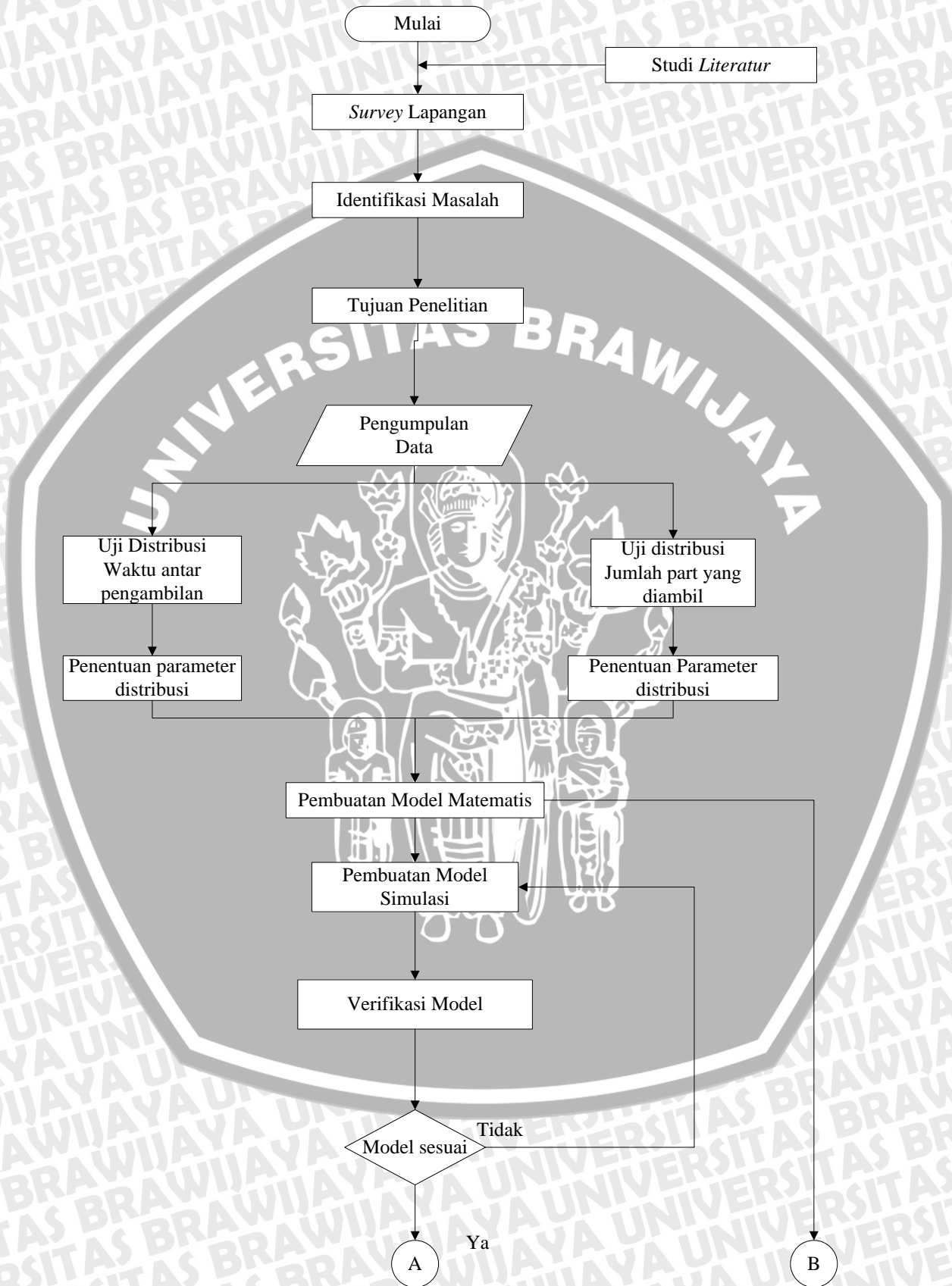
3.4.11.2 Analisa Hasil Eksperimen

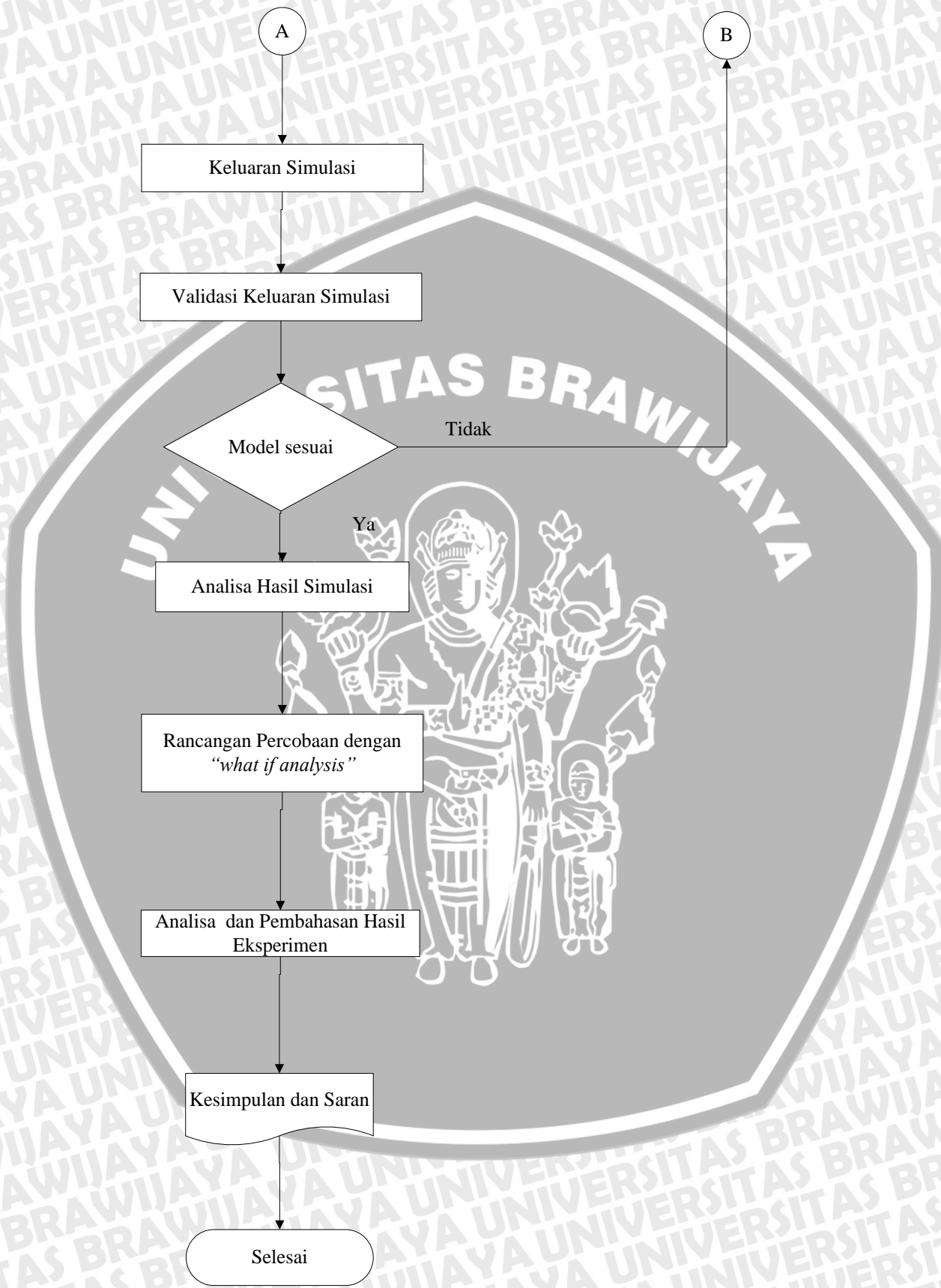
Pada tahap analisa hasil eksperimen, peneliti melakukan analisa terhadap hasil percobaan dengan melakukan uji ANOVA untuk mengetahui perbedaan diantara sistem yang ada. Setelah melakukan uji ANOVA, selanjutnya adalah melakukan uji BNT untuk mengetahui kesignifikan perbedaan rata-rata antara sistem X dan sistem Y, sistem X dan sistem X, sistem Y dan sistem Z. Setelah dilakukan uji ANOVA dan uji BNT, dari hasil perancangan eksperimen yang dilakukan akan didapatkan usulan untuk menentukan nilai ROP, Q dan IL rata-rata yang dapat mengefisiensikan *inventory value*.

3.4.12 Pembuatan Kesimpulan dan Saran

Dalam bab ini akan dibahas mengenai kesimpulan yang dapat diambil dari hasil pengolahan data. Dan saran-saran yang ada dapat digunakan untuk melakukan penelitian lebih lanjut

3.5 Diagram Alir Penelitian





Gambar 3.1 Diagram Alir Penelitian

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA

4.1 Studi Lapangan

PT. Petrokimia Gresik dibentuk dan didirikan secara resmi pada tanggal 10 Juli 1972. PT. Petrokimia Gresik memiliki logo sebagai berikut :



Gambar 4.1 Logo PT Petrokimia Gresik

Sumber : PT. Petrokimia Gresik

Arti Logo tersebut adalah :

- Gambar kerbau berwarna emas, dipilih sebagai penghormatan terhadap daerah kecamatan kebomas.
- Kerbau juga melambangkan sikap yang suka bekerja keras, loyal, dan jujur. Selain itu Kerbau adalah hewan yang dikenal luas oleh masyarakat Indonesia sebagai sahabat petani.
- Warna kuning emas pada hewan kerbau melambangkan keagungan.
- Daun hijau berujung lima melambangkan kesuburan dan kesejahteraan.
- Lima ujung daun melambangkan kelima sila dari Pancasila.
- Huruf **PG** berwarna putih singkatan dari Petrokimia Gresik.
- Warna putih pada huruf **PG** melambangkan kesucian.
- Logo mempunyai arti keseluruhan “ Dengan hati yang bersih berdasarkan kelima sila Pancasila, PT Petrokimia Gresik berusaha mencapai masyarakat yang adil dan makmur untuk menuju keagungan bangsa”

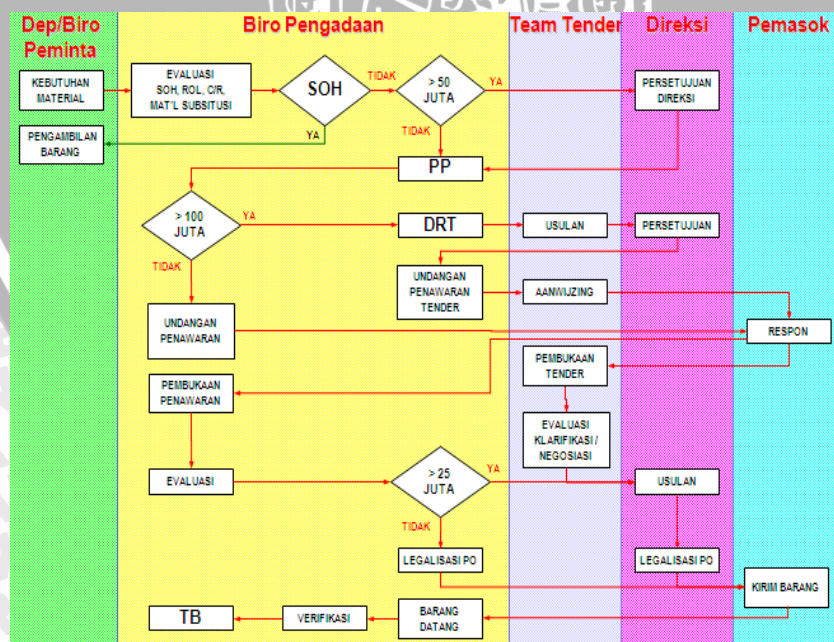
Proyek Petrokimia Surabaya, demikian nama yang digunakan untuk PETROKIMIA GRESIK. PT. Petrokimia Gresik memiliki 3 pabrik yaitu Pabrik 1, Pabrik II dan Pabrik III. Pada pabrik 1 yang dihasilkan adalah pupuk, ZA I dan ZA II, Amonia, CO₂ dan lain-lain. Sistem produksi yang digunakan di pada pabrik 1 adalah *flow process* dengan menggunakan mesin-mesin seperti *reactor, stripper, carb.*

condenser, scrubber, hp decomposer, lp decomposer, hot water pump, steam condensate pump, sealing water pump, ammonia boost up pump, hp absorbent pump, urea hydrolyzer pump, dan sebagainya. Pabrik 1 terdiri dari 5 unit yaitu unit Urea, ZA 1, ZA 11 Amoniak dan unit utilitas. Untuk menjamin kehandalan mesin-mesin yang digunakan di pabrik 1 sistem perawatan yang digunakan di PT Petrokimia menggunakan sistem *periodic maintenance*, *preventive maintenance*, dan juga *controlled maintenance*.

Pengadaan seluruh bahan baku dan *spare part* di PT Petrokimia Gresik ditangani oleh Biro Pengadaan. Biro ini mengatur tersedianya semua bahan baku dan *spare part* yang diperlukan untuk menjamin terlaksananya operasional perusahaan dengan baik. Tugas dari biro pengadaan adalah dari perencanaan dan pengendalian, pengadaan barang dan pembelian barang. Proses pengadaan barang yang terjadi mengikuti prosedur sebagai berikut :

1. Prosedur Pembuatan Permintaan Pembelian (PR-37-0001)
2. Prosedur Penerimaan, Penanganan dan Distribusi Barang/Material/Bahan (PR-37-0002)
3. Prosedur Pengadaan Barang & Jasa (PR-37-0003)
4. Prosedur Seleksi & Evaluasi Kinerja Rekanan (PR-37-0004)

Alur pengadaan barang yang terjadi ditunjukkan dalam gambar berikut ini :

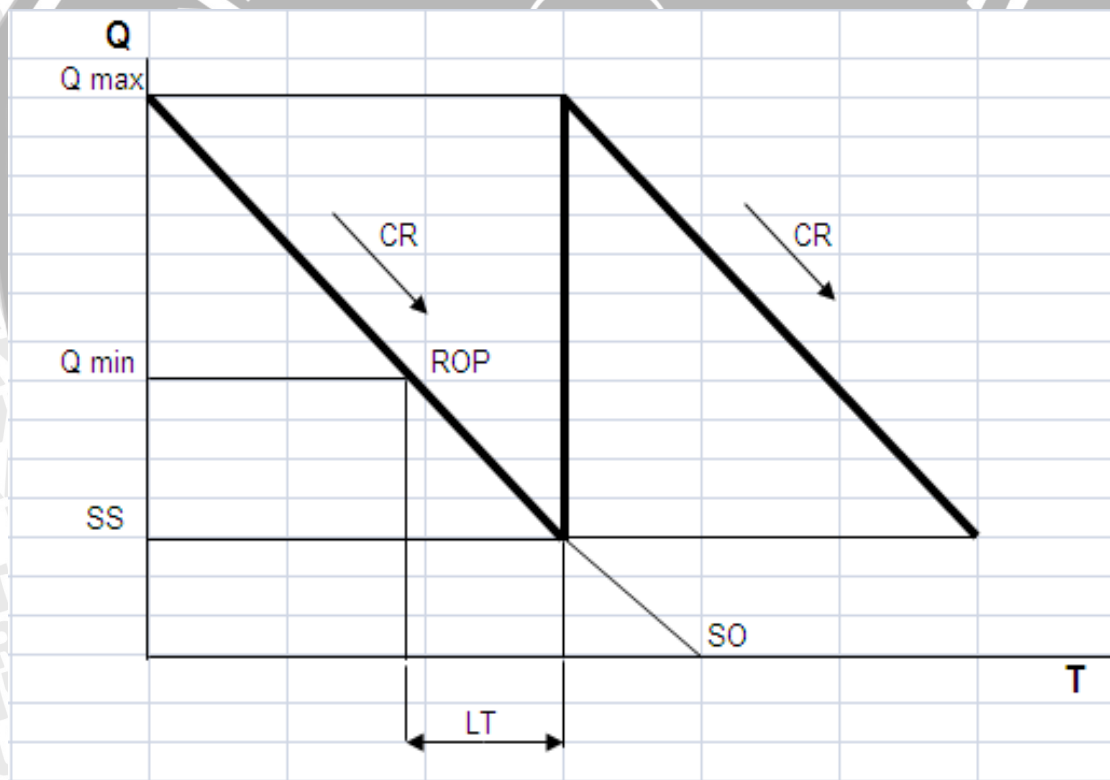


Gambar 4.2 Proses Pengadaan Barang di PT. Petrokima Gresik
Sumber : PT. Petrokimia Gresik

Biro Pengadaan menggolongkan *spare part* dalam *Asset class* yaitu pengelompokan barang yang didasarkan pada metode perencanaan dan permintaan pembelian. *Asset class* di PT. Petrokimia Gresik terdiri dari :

- *Asset class* H (berdasarkan permintaan user)
- *Asset class* I (berdasarkan permintaan user dan dititipkan di PGM)
- *Asset class* RO (berdasarkan re-order Point)
- *Asset class* Z (berdasarkan Persediaan Barang)

Dalam Sistem *Asset class* RO terdapat beberapa hal yang harus diperhatikan yaitu *consumption rate* (CR), *Lead Time* (LT) dan *safety Factor* (SF). Proses perencanaan pengadaan barang ROL harus sudah mulai diterbitkan *purchase requisition* (PR) pada suatu titik level persediaan batang yang dinamakan dengan *Re Order Point* (ROP). Adapun proses pengadaan barang ROL adalah sebagai berikut :



Gambar 4.3 Proses Pengadaan Barang ROL di PT. Petrokimia Gresik
 Sumber : PT. Petrokimia Gresik

Keterangan Gambar :

- ✓ Q = *Level stock*
- ✓ Qmax = *Level stock maksimum*
- ✓ Qmin = *Level stock minimum*

- ✓ SS = *Safety Stock*
- ✓ SO = *Stock Out*
- ✓ ROP = *ReOrder Point*
- ✓ LT = *Lead Time* (PR sampai dengan barang datang)
- ✓ T = Waktu
- ✓ CR = *Consumption Rate*
- ✓ SF = *Safety Stock*

4.1.1 Consumption Rate (CR)

Consumption Rate (CR) adalah jumlah rata-rata pengambilan barang dalam suatu satuan waktu tertentu. Satuan waktu yang dimaksud dalam hal ini bisa harian, bulanan atau tahunan. Perlu dicermati bahwa jumlah rata-rata tersebut tidak harus mutlak pengambilan dalam satu periode waktu sejumlah tersebut, namun berfluktuasi yang dipengaruhi beberapa factor. Pengambilan – pengambilan untuk pemakaian diluar pemeliharaan rutin tidak diperhitungkan dalam penetapan CR, misalnya pengambilan permintaan untuk keperluan *Turn Around (TA)* yaitu perbaikan menyeluruh dari suatu equipment dan sebagainya.

4.1.2 Lead Time (LT)

Lead Time adalah waktu yang diperlukan dalam proses pengadaan barang dimulai saat PR *Released* sampai dengan PO *closed (TB released)*. Di dalamnya terjadi beberapa tahapan proses yaitu permintaan penawaran ke rekanan, penawaran dari rekanan, evaluasi dan klarifikasi kesesuaian spesifikasi, negosiasi dan penentuan pemenang, penerbitan *purchase order (PO)*, pengiriman barang (DT), verifikasi kesesuaian barang terhadap PO dan penerimaan Barang (TB). Masing-masing proses memiliki periode waktu tersendiri total waktu yang dibutuhkan tersebut dinamakan *Lead Time*.

Secara umum (*empiris*), *Lead Time* dikelompokan menurut *source* dari barang yaitu :

1. Pengadaan barang dalam negeri adalah 3 bulan
2. Pengadaan barang luar negeri adalah 6 (enam) bulan.

Dengan catatan bahwa tidak termasuk barang yang harus difabrikasi terlebih dahulu.

4.1.3 Safety Factor (SF)

Safety Factor adalah suatu nilai yang diberikan sebagai faktor pengaman dalam proses perencanaan pengadaan suatu barang. Nilai *Safety Factor* akan menentukan berapa *stock* minimum yang harus ditetapkan. Nilai *Safety Factor* sangat bervariasi, dipengaruhi oleh beberapa hal yaitu *source* dari barang, karakteristik dari barang, aspek legal barang-barang dengan persyaratan legal yang ketat, nilai uang dari barang, dampak terhadap operasional pabrik, tingkat fluktuasi pengambilan barang, dan lain-lain.

Penentuan besaran SF merupakan agregat dari faktor-faktor tersebut, untuk memudahkan dalam formulasinya semestinya masing-masing faktor diberikan bobot tertentu. Dimana skor terakhir merupakan perkalian atau penjumlahan dari masing-masing skor.

Penetapan nilai SF dari suatu persediaan barang sangat mempengaruhi nilai persediaan gudang dari barang-barang *slow moving* karena pada dasarnya *safety factor* menggambarkan jumlah dari *dead stock* yang ada di gudang penyimpanan. Semakin tinggi *dead stock* maka semakin tinggi nilai persediaan material. Namun di sisi lain nilai SF juga menggambarkan kehandalan dari *supply* barang untuk unit pemakaian yang berarti juga kehandalan dari proses produksi yang ekuivalen dengan kehandalan produktivitas.

Untuk mengakomodasi kedua sisi tersebut maka dalam penentuan besarnya SF dari suatu persediaan harus memperhatikan dari unit persediaan tersebut. Pemilahan dapat dilakukan berdasarkan nilai satuan dari barang. Semakin rendah nilai satuan barang maka SF yang digunakan semakin tinggi begitu pula sebaliknya.

4.2 Pengumpulan data

Dalam penelitian ini pengumpulan data yang dilakukan adalah :

1. Kebijakan perusahaan tentang sistem persediaan di PT. Petrokimia Gresik.
2. Jenis-jenis *spare part* yang digunakan dalam penelitian
3. Data nilai *Inventory level spare part*, nilai *Q*, nilai *lead time* dan nilai ROP, nilai *service level*
4. Data waktu pengambilan *spare part* dan jumlah pengambilan *spare part*.
5. Nilai *Inventory Value* dari *spare part*

4.2.1 Kebijakan PT. Petrokimia

Dalam menentukan kebijakan persediaan PT. Petrokimia Gresik menggunakan rumusan sebagai berikut :

$$\begin{aligned} \text{Formulasi} \quad \text{ROP (Qmin)} &= \text{LT} \times \text{CR} + \text{Sf} (\text{LT} \times \text{CR}) \\ \text{Q max} &= 2 \times \text{Q min} \\ \text{SS} &= \text{Sf} (\text{LT} \times \text{CR}) \\ \text{Q Order} &= \text{Q max} - \text{SS} \end{aligned}$$

Dimana :

$$\begin{aligned} \text{ROP} &= \text{Re Order Point} & \text{SF} &= \text{Safety Factor} \\ \text{Q} &= \text{Jumlah barang} & \text{SS} &= \text{Safety Stock} \\ \text{LT} &= \text{Lead Time yaitu 3 bulan} \end{aligned}$$

4.2.2 Jenis-jenis Spare part

Pada penelitian ini, penelitian yang dilakukan adalah pada *spare part* yang tergolong jenis RO yaitu *spare part* yang harus selalu ada di gudang karena kebutuhannya yang rutin, dan sistem manajemen persediaan berdasarkan pada nilai ROP atau *re – order point*. Adapun jenis - jenis *spare part* yang digunakan adalah sebagai berikut :

Tabel 4.1 Jenis-jenis Spare part

No	Nama Spare part	Kode Spare part
1	ADAPTOR,SLEEVE,BEARING -- H315 – FAG	A
2	BELT,V -- V-BELT-SPB2000LW-RED-SEAL -- BANDO	B
3	BEARING -- BEARING-6305-2RS – AKG	C
4	BEARING UNIT -- BEARING-UCT211 – FYH	D
5	BEARING,BALL,ANNULAR -- BEARING-6307-2RS – AKG	E
6	BEARING,BALL,ANNULAR -- BEARING-6309 -- FAG	F
7	DEEP,GROOVE,BALL,BEARING -- BEARING-6213-2RSR – FAG	G
8	OIL SEAL -- TC-45X60X9-NBR – NOK	H
9	SHPERICAL ROLLER BEARING -- BEARING-22215EK – FAG	I
10	HOUSING,BEARING UNIT -- SNV130+DH515 -- FAG	J

Sumber : PT. Petrokimia Gresik

4.2.3 Nilai *Inventory Level* (IL), nilai ROP, nilai Q dan *service level*, *lead time*

Dari penelitian yang dilakukan di PT. Petrokimia Gresik didapatkan data nilai Q, nilai ROP, IL (*Inventory Level*) dan *Service level* sebagai berikut :

Tabel 4.2 Nilai Q, ROP,IL, Nilai *Service level*, *Lead Time*

No	Nama Spare part	IL	Nilai Q	Nilai ROP	<i>Service Level</i>	<i>Lead Time</i> (Bulan)
1	A	41	24	16	1	3
2	B	142	55	33	0,99	3
3	C	184	42	21	0,95	3
4	D	102	33	20	1	3
5	E	168	63	40	0,99	3
6	F	32	12	8	1	3
7	G	64	13	8	0,95	3
8	H	110	19	11	1	3
9	I	144	43	26	0,99	3
10	J	64	31	20	1	3

Sumber : PT. Petrokimia Gresik

4.2.4 Data Pengambilan *Spare part*

Data selanjutnya adalah data waktu pemakaian *spare part* dan jumlah pengambilan *spare part* yang dilakukan ditunjukkan pada lampiran 1.

4.2.5 Data *Inventory Value Spare part*

Pada penelitian ini didapatkan data *Inventory Value* yang merupakan data sekunder yang didapatkan dari perusahaan.

Tabel 4.3 *Inventory Value Spare part*

No	Kode <i>Spare Part</i>	<i>Inventory Value</i>
1	A	Rp 543.670
2	B	Rp 88.069
3	C	Rp 162.061
4	D	Rp 463.000
5	E	Rp 265.183
6	F	Rp 265.183
7	G	Rp 57.070
8	H	Rp 192.502
9	I	Rp 617.700
10	J	Rp 360.250

Sumber : PT. Petrokimia Gresik

4.3 Uji pendugaan Parameter

4.3.1 Pendugaan Parameter untuk waktu antar pengambilan *Spare part*

Pada pembahasan ini uji pendugaan parameter dengan menggunakan *software* Statfit menggunakan analisa *maxium likelihood*. Dengan pendekatan *Compound Poisson Proses* maka diasumsikan bahwa laju permintaan *spare part* mengikuti distribusi eksponensial. Hasil pendugaan parameter yaitu nilai beta dengan *software* Statfit ditunjukkan pada tabel 4.4.

Tabel 4.4. Hasil Pendugaan Parameter untuk waktu antar pengambilan *Spare part*

No	Kode <i>Spare part</i>	Nilai (β)
1	A	104,41
2	B	103,1
3	C	75,5
4	D	96,5
5	E	39,68
6	F	79,42
7	G	117,82
8	H	87,57
9	I	55
10	J	84,33

4.3.2 Pendugaan parameter untuk jumlah part yang diambil

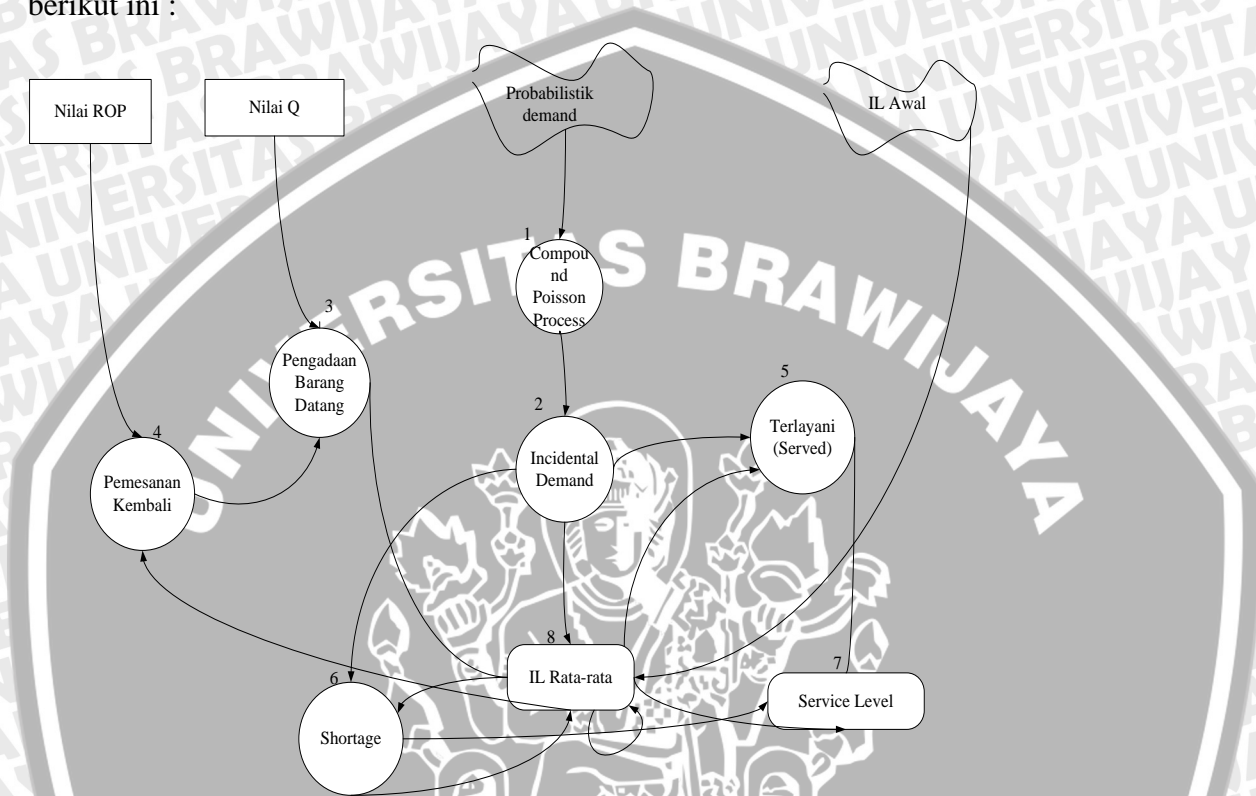
Untuk pendugaan parameter jumlah barang yang diambil, dengan bantuan *software* Statfit dengan analisa *maximum likelihood*, dengan melihat nilai *p-value*, apabila nilai *p-value* yang ada lebih besar dari 0.05% maka hipotesa tersebut diterima. Dari hasil perhitungan dengan menggunakan *software* Statfit pendugaan distribusi yang terjadi pada jumlah *spare part* yang diambil adalah distribusi Poisson dengan pendugaan nilai lamda yang ditunjukkan dalam tabel 4.5.

Tabel 4.5. Tabel Nilai lamda untuk distribusi Poisson

No	Kode <i>Spare part</i>	Nilai (λ)
1	A	2,357
2	B	8
3	C	2,346
4	D	3,857
5	E	2,068
6	F	1,2
7	G	1,5
8	H	1,8
9	I	2,611
10	J	2,8

4.4 Pembuatan Model Matematis

Dalam pembuatan model matematis terlebih dahulu membuat *Influence* diagram yaitu suatu diagram yang menunjukkan hubungan antar komponen dalam suatu sistem yang akan dimodelkan matematis. *Influence Diagram* ditunjukkan dalam gambar 4.4 berikut ini :



Gambar 4.4 Gambar *Influence Diagram*

Dari *influence diagram* diatas, model matematis Sistem *Persediaan Spare Part* dengan Pendekatan *Compound Poisson Process* terdapat beberapa komponen yaitu :

1. Input System

Dalam sistem input data terdiri dari *decision input* yaitu nilai Q dan nilai ROP, untuk input data adalah nilai IL awal dan nilai probabilistik *demand*.

2. Component System

Komponen yang terdapat didalam Sistem *Persediaan Spare part* dengan Pendekatan *Compound Poisson Process* terdiri dari 6 komponen yang ditandai dengan gambar lingkaran pada gambar 4.4. yaitu :

- Lingkaran No 1 *Compound Poisson Process*

Compound Poisson Process terdiri dari 2 yaitu fungsi β dan fungsi, yaitu sebagai berikut:

1. Fungsi β

Fungsi β adalah suatu fungsi yang dihasilkan dari fungsi distribusi eksponensial yang didapatkan dari rumus distribusi eksponensial sebagai berikut :

$$F(x) = \begin{cases} 1 - e^{-x/\beta}, & \text{untuk } 0 \leq x \\ 0, & \text{untuk } x \text{ yang lain} \end{cases}$$

mendapatkan nilai β dengan jalan invers $F(x) = F^{-1}$

$$u = F(x)$$

$$u = 1 - e^{-x/\beta}$$

$$\ln(1 - u) = -x/\beta$$

$$x = -\beta \ln(1 - u)$$

2. Fungsi λ

Fungsi λ adalah fungsi yang didapatkan dari distribusi Poisson yang didapatkan dari rumus distribusi poisson sebagai berikut :

$$f(x) = \begin{cases} \frac{e^{-\lambda} \lambda^x}{x!}, & \text{untuk } x \in (0,1,2,\dots) \\ 0, & \text{untuk } x \text{ yang lain} \end{cases}$$

Mendapatkan nilai λ dengan jalan invers $f(x)$ yaitu:

$$u = f(x) ; U = \text{Hasil Pembangkitan}$$

$$u = e^{-x} \lambda^x / x!$$

$$u \cdot e^x \cdot x! = \lambda^x$$

$$\lambda = (u \cdot e^x \cdot x!)^{1/x}$$

$$x = \ln(u \cdot e^x \cdot x!) / \lambda$$

- Lingkaran nomer 2 *Incidental Demand*
Dimana $N \approx \text{Compound Poisson Process}$ (Fungsi β dan fungsi λ)
Demand = Demand + N
- Pengadaan Barang datang
Pengadaan Barang Datang = Q + *Lead Time*
- Pemesanan Kembali
Pemesanan kembali = ROP
- Terlayani (*served*)

$$\text{Served} = \text{Served} - \text{IL}$$

- Shortage

$$\text{Shortage} = \text{Shortage} + (\text{N} - \text{IL})$$

3. Output System

Output dari sistem persediaan *Spare part* adalah

- nilai IL seperti pada nomer 7

$$\text{IL} = \text{IL} / \text{TSimulasi}$$

- nilai service level seperti pada nomer 8

$$\text{Service Level} = \text{Served} / \text{Demand}$$

4.5 Model Simulasi

4.5.1 Pendekatan Pemograman Simulasi

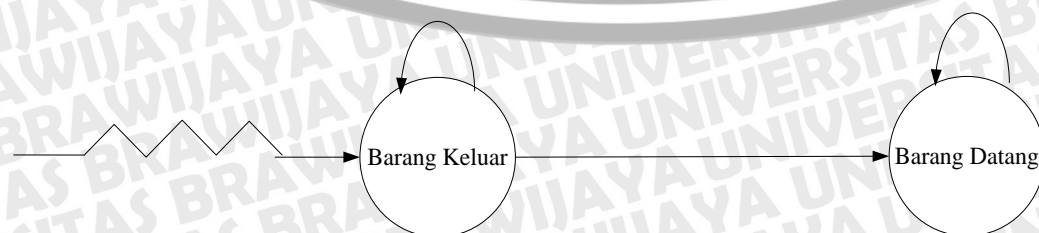
Dalam pembuatan program simulasi sistem persediaan *spare part*, peneliti menggunakan pendekatan *event (event approach)*. Pendekatan *event* dalam pembuatan program simulasi adalah upaya merancang program secara umum dengan fokus perhatian dicurahkan pada terjadinya peristiwa-peristiwa dalam sistem kajian yaitu sistem Persediaan *spare part*. Dengan begitu, rutin-rutin programnya akan berisi uraian operasi yang semuanya mengacu pada kapan dan berapa lama peristiwa itu terjadi (Asmungi, 2006;73)

Pada simulasi sistem persediaan *spare part* ini terdapat 2 *event* yang ada dalam sistem yaitu :

1. *Event Kejadian Maintenance* yang berarti barang keluar
2. *Event Barang datang*.

Dari 2 kejadian yang ada yaitu kejadian kedatangan *maintenance* dan kejadian barang datang, tergambar dalam diagram kejadian (*event graph*) yang ditunjukkan pada gambar

4.5.



Gambar 4.5 *Event Graph* Sistem

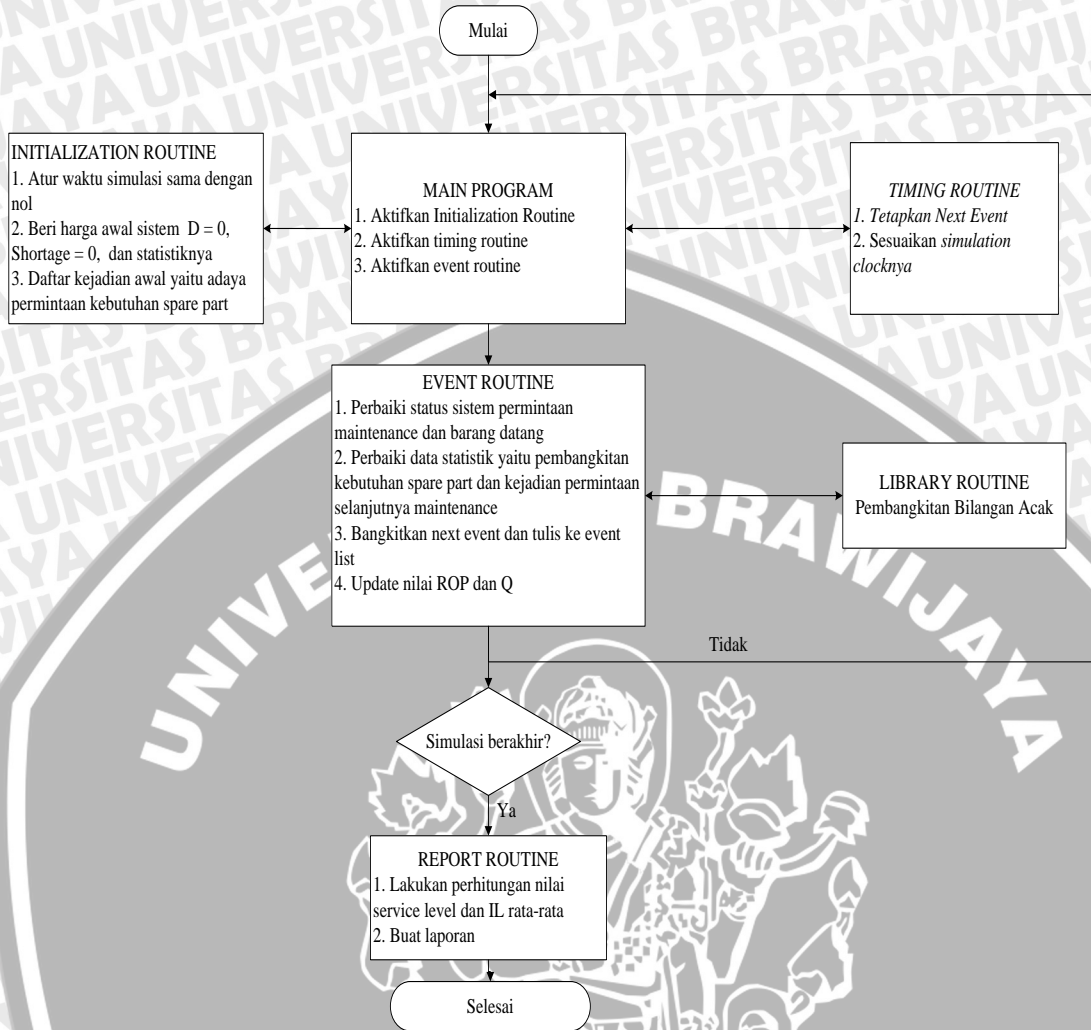
Pada simulasi sistem persediaan *spare part* terdapat dua *event* yang terjadi yaitu *event* barang keluar dan *event* barang datang. *Event* barang keluar yaitu kejadian keluarnya barang dari gudang yang disebabkan adanya permintaan *spare part* dari pihak *maintenance*. Dengan adanya barang di gudang, maka gudang dapat melayani permintaan *spare part* dari pihak *maintenance*. Apabila barang yang diminta tidak ada maka akan menunggu permintaan terlayani dan akan adanya pembangkitan permintaan barang datang.

Event kedua adalah barang datang, pada *event* ini barang yang telah diminta dalam pembangkitan akan datang untuk melayani permintaan yang ada. Apabila barang yang telah datang lebih kecil dari nilai *shortage* (kekurangan barang) maka akan ada pembangkitan permintaan barang datang dan barang yang ada akan menunggu untuk terlayani. Apabila barang yang datang telah memenuhi permintaan maka berlanjut pada *event* 1 yaitu adanya permintaan *maintenance* dan menyebabkan barang keluar.

4.5.2 Flowchart

4.5.2.1 Flow chart Anatomi Program Simulasi.

Dalam perancangan program simulasi terdapat beberapa komponen yang memiliki tugas dan wewenang yang berbeda-beda dimana masing-masing komponen tersebut memiliki keterkaitan fungsional dan juga memiliki keterkaitan logika pemrograman, disamping juga adanya keterkaitan yang disebabkan oleh tugas dan wewenang yang. Anatomi program simulasi merupakan langkah yang paling penting dalam perancangan program simulasi, *flowchart* dari anatomi program simulasi ditunjukkan pada gambar 4.6.



Gambar 4.6. Anatomi Program Simulasi

Pada gambar *flowchart* Anatomi Program Simulasi terdapat beberapa komponen yang hampir pasti dapat dijumpai pada setiap program simulasi yaitu :

1. **Komponen Inisialisasi (Initialization routine)**

Sub program ini merupakan bagian program yang pertama kali dieksekusi. Bagian ini digunakan untuk memberikan harga awal variabel statistik, daftar *event*, dan status sistem pada saat jam simulasi $t = 0$. Kejadian awal pada sistem ini adalah adanya permintaan *maintenance* dari pihak perawatan dan pada komponen ini diberikan nilai awal $ILvs = 0$, $demand = nol$, $served = 0$, $shortage = 0$, $Tsim = Hari$, $IL = IL\ awal$, dan lain-lain.

2. **Komponen pengaturan waktu (timing routine)**

Komponen ini, bertugas untuk menetapkan waktu kapan kedatangan *event* berikutnya terjadi dari *event list*, kemudian bertugas menyesuaikan jam simulasinya.

3. **Komponen operasi (*event routine*)**

Sub program yang bertugas untuk membangkitkan kejadian *event* berikutnya dan meremajakan ulang atas status sistem dan data statistik sebagai, akibat terjadinya *event* tersebut. Dalam penelitian ini *event routine* yang terjadi adalah *event* permintaan *maintenance* dan *event* barang datang. Dari *event routine* yang dijalankan akan menyebabkan nilai Q dan ROP ter-*update*. Nilai statistic yang akan ter-*update* adalah kebutuhan *spare part* dan kejadian permintaan *maintenance*.

4. **Komponen pustaka (*library routine*)**

Sub program yang berisi fungsi-fungsi dan atau prosedur-prosedur tertentu yang digunakan membantu komponen operasi. Dalam sistem ini library berisikan pembangkitan bilangan acak dan pembangkitan kontinyu 0,1 dan juga memasukan fungsi poisson dan fungsi eksponensial.

5. **Komponen pembangkit laporan (*report generator*)**

Sub program yang digunakan untuk menghasilkan laporan jika simulasi telah berakhir. Laporan berisi nilai dari *service level* dan nilai IL rata-rata.

6. **Komponen utama(*main routine*)**

Sub program ini menjadi bagian yang terpenting dari keseluruhan program, Karena bertugas sebagai koordinator yang mengkoordinir komponen-komponen:

- a. Komponen inisialisasi
- b. Komponen pengaturan waktu
- c. komponen operasi
- d. Komponen Laporan

Koordinasi antar komponen dilakukan secara terus menerus selama eksekusi simulasi berlangsung dan akan berhenti apabila jam simulasi telah menunjukkan batas waktu eksekusi yang dikehendaki. Urutan logika adalah sebagai berikut :

1. Diawali dengan main rutin mengaktifkan inisialisasi, yaitu dengan mengatur jam simulais ke nol, juga memberikan harga awal pada status sistem, data statistik, dan daftar *event*.
2. Selanjutnya pengendalian yang dilakukan oleh komponen utama dialihkan ke komponen pengaturan waktu guna menetapkan *event* tipe mana yang paling

cepat terjadi? Bila telah terjadi, maka jam simulasi digeser ke depan sejauh saat *event* terjadi.

3. Sekembali dari komponen waktu, komponen utama meneruskan pengendaliannya ke komponen operasi, yaitu untuk menyesuaikan status Sistem akibat terjadinya *event* tadi. Selanjutnya informasi tentang performansi sistem kemudian dikumpulkan, yaitu dengan cara menyesuaikan data statistik sebelumnya.
4. Menindaklanjuti dari perintah komponen utama, maka komponen operasi mengaktifkan komponen pustaka guna membangkitkan variabel-variabel acak dengan distribusi tertentu.
5. Setelah melakukan langkah ke 4, maka kembali ke langkah 2, sehingga siklus 2 - 3 - 4 - 2 akan terus menerus diulang sampai pada kondisi bahwa simulasi telah usai. Setelah simulasi selesai kemudian ditutup dengan perhitungan performansi sistem dan diikuti pembuatan laporan oleh komponen pembangkit laporan.

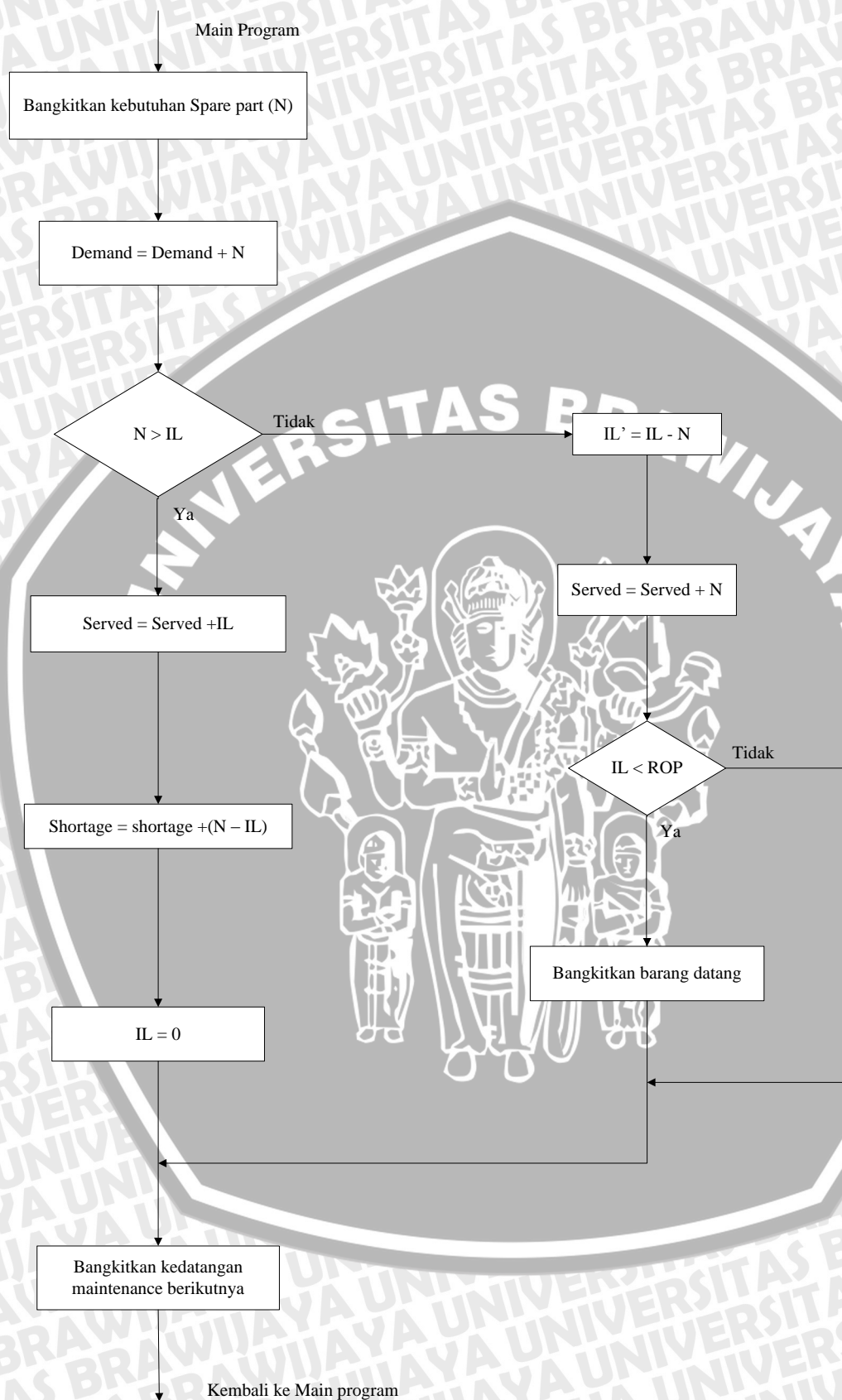
4.5.2.2 Flowchart Event Permintaan dari Maintenance dan Barang Datang

Adanya kejadian kedatangan *maintenance* pada gambar 4.7 atau kejadian perawatan menyebabkan adanya permintaan barang atau permintaan kebutuhan akan *spare part*. Maka pada gambar *flowchart* yang pertama terjadi adalah bangkitkan permintaan kebutuhan *spare part* (N). Lalu permintaan akan kebutuhan *spare part* menjadi *demand* dalam sistem, menjadi *demand* dengan menambah *demand* yang telah lalu dengan N pada $t = 0$ maka nilai $demand = N$, lalu dilihat apakah permintaan *spare part* yang dilakukan lebih besar dari ROP ataukah tidak. Apabila tidak, maka IL yang ada sekarang dikurangi dengan N maka *served* yang dilakukan adalah *served* ditambah dengan *served* yang pernah terlayani dahulunya ditambah dengan N , lalu dilihat apakah IL yang ada sekarang lebih kecil dari ROP. Apabial IL lebih kecil dari ROP maka bangkitkan kedatangan barang apabila tidak maka berlanjut ke pembangkitan *maintenance* selanjutnya, kembali pada *main program*. Apabila nilai IL lebih besar dari N maka *served* yang dapat dilakukan adalah *served* yang pernah dilakukan dalam sistem ditambah dengan IL yang ada sebagai sisa. Maka menjadi *shortage*, dimana *shortage* adalah *shortage* yang pernah terjadi dalam sistem ditambah dengan N dikurangi IL. lalu set IL sama dengan nol kembali ke *main program* bangkitkan kebutuhan *spare part*. Hal

ini nilai *served*, *demand* dan *shortage* akan ter-*update* sesuai dengan waktu simulasi yang dijalankan.

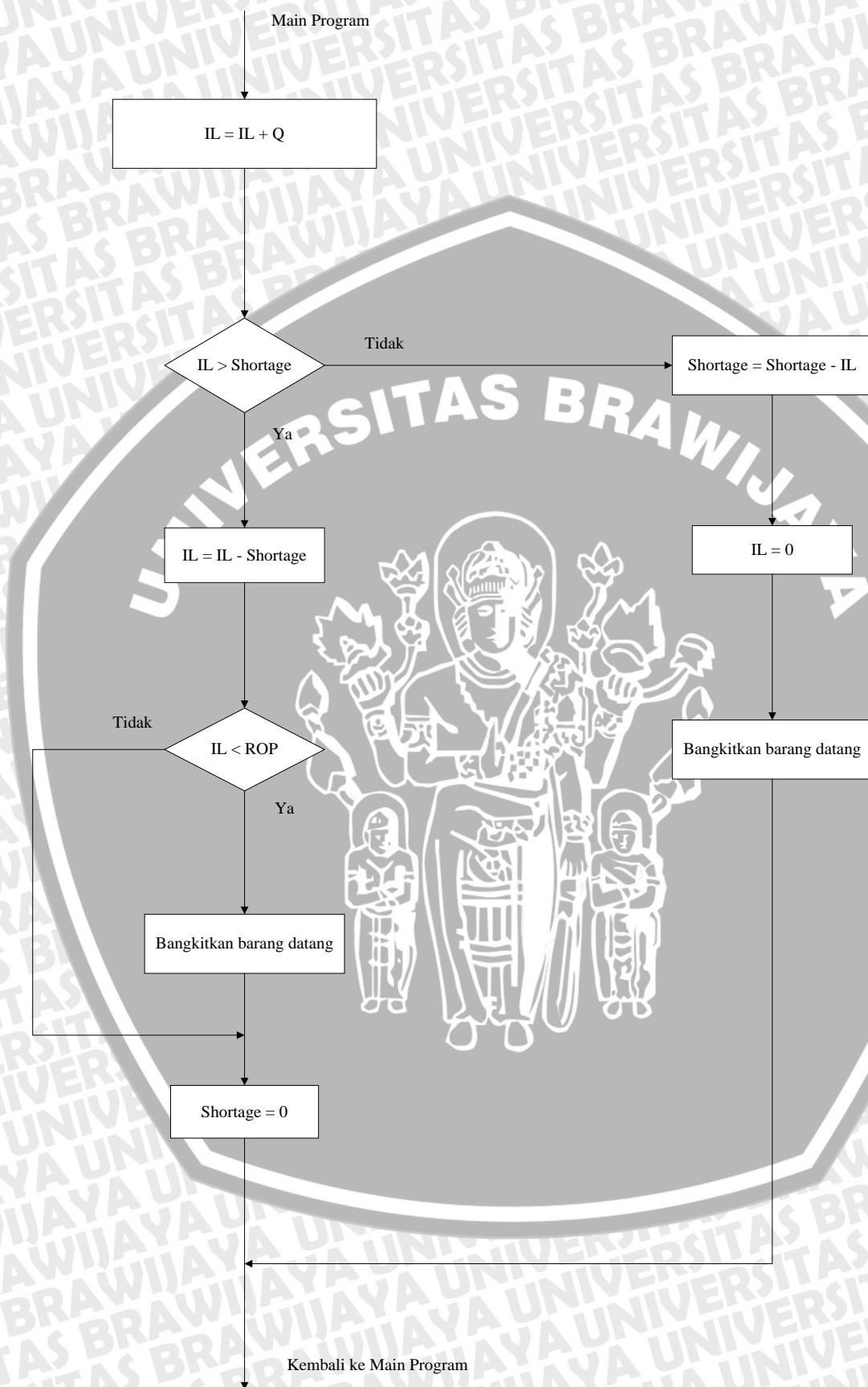
Kejadian kedua dalam sistem simulasi persediaan ini adalah kejadian barang datang yang terdapat pada gambar 4.8. Kejadian barang datang diawali dengan penambahan nilai IL yang ada sekarang ditambahkan dengan nilai Q yang ditentukan dalam sistem. Setelah penambahan nilai IL dan Q selanjutnya dilihat apakah IL yang telah ditambah nilai Q masih lebih besar dari *shortage* yang ada apabila iya maka IL yang ada dapat mengurangi nilai *shortage* maka IL dikurangi *shortage*, lalu dilihat apakah IL yang ada lebih kecil dari nilai ROP apabila tidak maka set *shortage* = 0 lalu kembali ke main program *update* nilai IL, apabila iya maka bangkitkan kedatangan barang selanjutnya. Apabila nilai IL masih lebih kecil dari *shortage* maka nilai *shortage* adalah *shortage* yang ada sekarang dikurangi nilai IL, lalu set IL = 0 hal ini disebabkan IL telah habis untuk melayani nilai *shortage* lalu bangkitkan kedatangan barang selanjutnya, kembali ke *main program*.





Gambar 4.7 Flowchart Event Routine Permintaan Maintenance





Gambar 4.8. Flowchart Event Routine Barang Datang

4.6 Perancangan dan Verifikasi Program Simulasi

4.6.1 Perancangan Desain Antar Muka

Pada penelitian ini pembuatan program simulasi komputer menggunakan *Visual Basic 6.0*. Hal ini dikarenakan dengan bahasa pemrograman yang terdapat di dalam *Visual Basic 6.0*. memudahkan peneliti untuk memecahkan permasalahan yang ada dalam sistem dan juga *Visual Basic* termasuk dalam bahasa pemrograman umum. Bahasa pemrograman umum adalah bahasa pemrograman yang dapat dipakai pada berbagai jenis persoalan (Setiawan, 167:1991)

Langkah pertama dalam pembuatan program simulasi, yaitu membuat desain antar muka. Dalam *visual basic 6.0* untuk membuat desain antar muka kita menggunakan fasilitas form yang ada dalam *Visual Basic 6.0*. Ada dua desain antar muka yang dirancang yaitu :

1. Desain anatr muka sebagai tempat untuk memasukan data (*input data*).

Pada desain antar muka terdapat 6 *text box* yang berfungsi untuk memasukan data yaitu :

1. Nilai ROP = Untuk memasukan nilai ROP yang diinginkan
2. Nilai Q = Memasukan nilai Q
3. Nilai IL Awal = Memasukan nilai IL awal
4. Nilai Lamda = Memasukan parameter distribusi Poisson
5. Nilai Beta = Memasukan parameter distribusi eksponensial
6. Hari = Untuk memasukan waktu simulasi

Selain terdapat 6 *text box* sebagai tempat untuk memasukan data, pada desian antar muka inputan data juga terdapat dua tombol *commond button* yaitu :

1. Simulasi = Perintah untuk menjalankan simulasi program komputer
2. Keluar = Perintah keluar dari antar muka program simulasi

Desain antar muka sebagai tempat untuk inputan data dapat terlihat dalam gambar 4.9 dibawah ini:

Gambar 4.9 Desain Antar muka program

2. Desain antar muka hasil eksekusi simulasi

Desain antar muka yang kedua yang dibuat adalah desain antar muka untuk menampilkan hasil dari simulasi yang telah dijalankan. Pada desain eksekusi simulasi ini akan ditampilkan nilai *service level* dari sistem dan nilai *Inventory Level* rata-rata yang dihasilkan dari menjalankan program simulasi. Pada desain antar muka terdapat tampilan grafik sebagai bentuk dari hasil simulasi yang telah dijalankan.

Adapun tampilan desain antar muka untuk eksekusi simulasi seperti berikut ini:

Gambar 4.10 Gambar Eksekusi Simulasi

4.6.2 Kode Pemograman

Setelah membuat desain antar muka pada program simulasi, selanjutnya menuliskan kode pemograman pada jendela koding yang tersedia dalam *Visual Basic* 6.0, yaitu :

4.6.2.1 Kode *Library Routine*

Pada library routine terdapat fungsi – fungsi statistik yang digunakan yaitu :

1. Bilangan Acak

Bilangan acak yang digunakan untuk pembangkitan variabel acak.

Public Sub Randdf ()

ZRNG (1) = 1973272912

ZRNG (2) = 281629970

ZRNG (3) = 20006270

ZRNG (4) = 1280689831

ZRNG (5) = 2096730329

ZRNG (6) = 1933576050

ZRNG (7) = 913566091

ZRNG (8) = 246780520

ZRNG (9) = 1363774876

ZRNG (10) = 604901985

ZRNG (11) = 1511192140

ZRNG (12) = 125981944

ZRNG (13) = 824064364

ZRNG (14) = 150493284

ZRNG (15) = 242708531

ZRNG (16) = 75253171

ZRNG (17) = 1964472944

ZRNG (18) = 120229975

ZRNG (19) = 233217322

ZRNG (20) = 1911216000

ZRNG (21) = 726370533

ZRNG (22) = 403498145

ZRNG (23) = 993232223

ZRNG (24) = 1103205531

ZRNG (25) = 762430696

ZRNG (26) = 1922803170

ZRNG (27) = 1385516923

ZRNG (28) = 76271663

ZRNG (29) = 413682397

ZRNG (30) = 726466604

ZRNG (31) = 336157058

ZRNG (32) = 1432650381

ZRNG (33) = 1120463904

ZRNG (34) = 595778810

ZRNG (35) = 877722890

ZRNG (36) = 1046574445

ZRNG (37) = 68911991

ZRNG (38) = 2088367019

ZRNG (39) = 748545416

ZRNG (40) = 622401386

ZRNG (41) = 2122378830

ZRNG (42) = 640690903

ZRNG (43) = 1774806513

ZRNG (44) = 2132545692

ZRNG (45) = 2079249579

ZRNG (46) = 78130110

ZRNG (47) = 852776735

ZRNG (48) = 1187867272

ZRNG (49) = 1351423507

ZRNG (50) = 1645973084

ZRNG (51) = 1997049139

ZRNG (52) = 922510944

ZRNG (53) = 2045512870

ZRNG (54) = 898585771

ZRNG (55) = 243649545

ZRNG (56) = 1004818771

UNIVERSITAS BRAWIJAYA



ZRNG (57) = 773686062
ZRNG (58) = 403188473
ZRNG (59) = 372279877
ZRNG (60) = 1901633463
ZRNG (61) = 498067494
ZRNG (62) = 2087759558
ZRNG (63) = 493157915
ZRNG (64) = 597104727
ZRNG (65) = 1530940798
ZRNG (66) = 1814496276
ZRNG (67) = 536444882
ZRNG (68) = 1663153658
ZRNG (69) = 855503735
ZRNG (70) = 67784357
ZRNG (71) = 1432404475
ZRNG (72) = 619691088
ZRNG (73) = 119025595
ZRNG (74) = 880802310
ZRNG (75) = 176192644
ZRNG (76) = 1116780070
ZRNG (77) = 277854671
ZRNG (78) = 1366580350
ZRNG (79) = 1142483975
ZRNG (80) = 2026948561
ZRNG (81) = 1053920743
ZRNG (82) = 786262391
ZRNG (83) = 1792203830
ZRNG (84) = 1494667770
ZRNG (85) = 1923011392
ZRNG (86) = 1433700034
ZRNG (87) = 1244184613
ZRNG (88) = 1147297105
ZRNG (89) = 539712780

UNIVERSITAS BRAWIJAYA



```
ZRNG(90) = 1545929719
ZRNG(91) = 190641742
ZRNG(92) = 1645390429
ZRNG(93) = 264907697
ZRNG(94) = 620389253
ZRNG(95) = 1502074852
ZRNG(96) = 927711160
ZRNG(97) = 364849192
ZRNG(98) = 2049576050
ZRNG(99) = 638580085
ZRNG(100) = 547070247
```

End Sub

2. Pembangkitan bilangan acak untuk mendapatkan U (0,1)

Kode pembangkitan Bilangan Acak ini ini adalah untuk mendapatkan nilai $U = (0,1)$ yang nantinya akan membantu untuk mendapatkan nilai $F(x)$ pada pola distribusi yang terjadi.

```
Public Static Function Rand_01(Stream As Integer)
As Single
Dim B2E15 As Long, B2E16 As Long, MODLUS As Long
Dim MULT1 As Long, MULT2 As Long
Dim HI15 As Long, HI31 As Long, LOW15 As Long
Dim LOWPRD As Long, OVFLOW As Long, ZI As Long

B2E15 = 32768
B2E16 = 65536
MODLUS = 2147483647
MULT1 = 24112
MULT2 = 26143
ZI = Rand_Seed(Stream)
HI15 = ZI \ B2E16
LOWPRD = (ZI - HI15 * B2E16) * MULT1
LOW15 = LOWPRD \ B2E16
```



```

HI31 = HI15 * MULT1 + LOW15
OVFLOW = HI31 \ B2E15
ZI = (((LOWPRD - LOW15 * B2E16) - MODLUS) +
(HI31 - OVFLOW * B2E15) * B2E16) + OVFLOW
If ZI < 0 Then
    ZI = ZI + MODLUS
End If
HI15 = ZI \ B2E16
LOWPRD = (ZI - HI15 * B2E16) * MULT2
LOW15 = LOWPRD \ B2E16
HI31 = HI15 * MULT2 + LOW15
OVFLOW = HI31 \ B2E15
ZI = (((LOWPRD - LOW15 * B2E16) - MODLUS) +
(HI31 - OVFLOW * B2E15) * B2E16) + OVFLOW
If ZI < 0 Then
    ZI = ZI + MODLUS
End If
Rand_Seed(Stream) = ZI

Rand_01 = (2 * (ZI \ 256) + 1) / 16777216
End Function

```

3. Pembangkitan Variabel Acak Berdistribusi Eksponensial

Kode ini dimasukan utntuk mendapatkan nilai F(x) dari nilai parameter distribusi eksponensial yang telah yang telah kita masukkan.

```

Public Static Function Expon_Rnd(Stream As
Integer, Mean_Input As Single) As Single
Dim U As Single
    U = Rand_01(Stream)
    Expon_Rnd = (-Mean_Input) * Log(U)
End Function

```

4. Pembangkitan Variabel Acak berdistribusi Poisson

Kode ini bertujuan untuk mendapatkan nilai $F(x)$ dari fungsi distribusi poisson yang telah kita masukan.

```
Public Static Function Poisson_Rnd(Stream As
Integer, Lambda_Input As Single) As Integer
'bangkitkan kebutuhan sparepart
Dim a, b, U As Single
Dim I As Integer
a = Exp(-LambdaInput)
b = 1
I = 0
Do
U = Rand_01(Stream)
I = I + 1
b = b * U
Loop Until (b < a)
Poisson_Rnd = I
End Function
```

4.6.2.2 Kode Main Program

Pembuatan kode *Main Program* bertujuan untuk menjalankan *main program* yang telah kita buat yaitu yang terdiri dari Inisialisasi, *timing*, *event routine*, *library* dan mneghasilkan *report* sebagai hasil dari simulasi yang telah dijalankan. *Main program* berfungsi sebagi alur berjalanya simulasi dengan mengaktifkan logika yang telah dibuat dalam *main program*.

```
Initialize
Do
Timing
UpdateTimeAvgStats
If NextEventType = 1 Then
```



```

    Barang
Else
    Arrive
End If
Loop Until (TimeSIM >= TimeNeed)
End Sub
Private Sub Sim_Report()
    Label7.Caption = ILvsTIME / TimeSIM
    Label10.Caption = ILMax
    ProgressBar1.Value = Int((Served / Demand) * 100)
    Label14.Caption = (Served / Demand)
End Sub

```

4.6.2.3 Kode Inisialisai (*Initialization*)

Kode Inisialisasi yaitu perintah untuk menjalankan komponen inisialisai dari program simulasi.

```

Public Sub Initialize()
    Demand = 0
    TimeSIM = 0
    TimeLastEvent = 0
    NumEvent = 2
    IL = ILMawal
    ILMMax = ILMawal
    ILvsTIME = 0
    Shortage = 0
    TimeNextEvent(1) = 1E+30
    NumEventOccur(1) = 0
    TimeNextEvent(2) = TimeSIM + Expon_Rnd(1,
                                                Beta)
    NumEventOccur(2) = 0
    Served = 0
    TimeNeed = Hari

```

End Sub

4.6.2.4 *Timing*

Timing adalah kode perintah untuk menjalankan komponen *timing*, dimana komponen *timing* untuk menjalankan logika waktu berlajanya simulasi. Kode *Timing* adalah sebagai berikut :

```
Public Sub Timing()
Dim I As Integer
Dim MinTimeNextEvent As Single
MinTimeNextEvent = 1E+29
NextEventType = 0
For I = 1 To 2
    If TimeNextEvent(I) < MinTimeNextEvent Then
        MinTimeNextEvent = TimeNextEvent(I)
        NextEventType = I
    End If
Next I
If NextEventType > 0 Then
    NumEventOccur(NextEventType)
    = NumEventOccur(NextEventType)
    + 1
End If
TimeSIM = MinTimeNextEvent
End Sub
```

4.6.2.5 *Update Statistik*

```
Public Sub UpdateTimeAvgStats()
Dim TimeSinceLastEvent As Single
TimeSinceLastEvent = TimeSIM - TimeLastEvent
TimeLastEvent = TimeSIM
ILvsTIME = ILvsTIME + (IL * TimeSinceLastEvent)
```


End Sub

4.6.2.6 Event Permintaan Maintenance

Kode *Event* Permintaan *Maintenance* berisi perintah untuk pembangkitan kejadian *maintenance*. Kode ini berisikan logika alur dari permintaan *maintenance* yang terjadi dalam sistem. Kode ini nanti akan juga terdapat dalam *main program* karena kode ini salah satu unsur dalam simulasi sistem persediaan *spare part*

```
Public Sub Arrive()
    N = Poisson_Rnd(2, Lamda)
    Demand = Demand + N
    If N > IL Then
        Served = Served + IL
        Shortage = Shortage + (N - IL)
        IL = 0
    Else
        IL = IL - N
        Served = Served + N
        If IL <= ROP Then
            TimeNextEvent(1) = TimeSIM + 90
        End If
    End If
    TimeNextEvent(2) = TimeSIM + Expon_Rnd(1, Beta)
End Sub
```

4.6.2.7 Event Barang Datang

Kode *event* Barang datang adalah kode yang berisikan perintah untuk pembangkitan barang datang. Logika yang terdapat didalamnya adalah alur yang terjadi dalam kejadian barang datang.

```
Public Sub Barang()
    IL = IL + Q
```

```

If IL > Shortage Then
  IL = IL - Shortage
  If IL <= ROP Then
    TimeNextEvent(1) = TimeSIM + 90
  Else
    TimeNextEvent(1) = 1E+30
  End If
  Shortage = 0
Else
  Shortage = Shortage - IL
  IL = 0
  TimeNextEvent(1) = TimeSIM + 90
End If
If IL > ILMax Then
  ILMax = IL
End If

End Sub

```

4.6.3 Verifikasi Program

Dalam pembuatan program simulasi selanjutnya program yang telah dibuat diuji apakah sudah terbebas dari *syntax error*, *design error* dan *run time error*. Pada Visual Basic 6.0 akan mampu melacak apabila terdapat *syntax error*, atau kesalahan yang terjadi dalam penulisan program. Pada pembuatan simulasi sistem persediaan ini, tidak terdapat kesalahan karena tidak keluarnya peringatan salah saat menjalankan program ataupun pada saat pembuatan program.

4.7 Keluaran Program Simulasi dalam *Pilot Run*

Setelah program simulasi telah selesai disusun selanjutnya adalah menjalankan simulasi dengan memasukkan nilai-nilai IL awal, nilai ROP, dan nilai Q dengan lama simulasi sebanyak 1460 hari. Hasil keluaran dari sistem simulasi persediaan *spare part* dalam tabel 4.7. sebagai berikut :

Tabel 4.6. Keluaran Simulasi *Pilot Run*

No	Kode <i>Spare part</i>	IL Awal	Nilai Q	Nilai ROP	<i>Service Level</i>	IL Rata-rata
1	A	41	24	16	1	30,3993
2	B	142	55	33	1	141,3993
3	C	184	42	21	1	169,3557
4	D	102	33	20	1	91,392
5	E	168	63	40	1	147,19
6	F	32	12	8	1	18,755
7	G	64	13	8	1	54,88
8	H	110	19	11	1	98,05
9	I	144	43	26	1	126,97
10	J	64	31	20	1	116,05

Dari hasil keluaran simulasi pada tabel di atas terlihat bahwa nilai *service level* yang ada dari sistem mencapai nilai 1, namun nilai IL rata-rata masih terlalu tinggi. Yang disebabkan oleh tingginya nilai *Stock On Hand* atau *Inventory Level* awal yang ada di gudang PT.Petrokimia Gresik.

4.8 Validasi Model

Pada pembuatan sistem persediaan ini validasi dari model yang digunakan dengan menggunakan uji *Independent T-Test* dimana membandingkan antara *service level* yang ada dalam sistem nyata dengan *service level* hasil dari simulasi, dan juga membandingkan nilai IL rata-rata yang dihasilkan dari *me-run* simulasi dengan nilai dari IL awal yang didapat dari perusahaan.

Hasil perbandingan *service level* dan IL rata-rata yang didapatkan dari keadaan nyata sistem dengan nilai yang dihasilkan dari menjalankan simulasi ditunjukkan pada tabel 4.7. berikut ini :

Tabel 4.7. Hasil Perbandingan sistem nyata dengan hasil simulasi

No	Kode Spare part	IL awal	IL Rata-rata Hasil simulasi	Service Level Keadaan nyata	Service Level Simulasi
1	A	41	30,3993	1	1
2	B	142	141,3993	0,99	1
3	C	184	169,3557	0,95	1
4	D	102	91,392	1	1
5	E	168	147,19	0,99	1
6	F	32	18,755	1	1
7	G	64	54,88	0,95	1
8	H	110	98,05	1	1
9	I	144	126,97	0,99	1
10	J	64	116,05	1	1

Untuk melakukan pengujian terhadap hasil simulasi dengan hasil sistem nyata dilakukan pengujian *Independent sample t-test* pada hasil *service level* dan nilai IL rata-rata dengan pengujian sebagai berikut :

1. Penentuan H_0 dan H_1

$H_0 : \mu_1 = \mu_2 =$ Tidak ada Perbedaan antara simulasi sistem dengan keadaan nyata

Vs

$H_1 : \mu_1 \neq \mu_2 =$ Ada perbedaan antara simulasi dengan keadaan nyata

2. $\alpha = 0,05$

$$df = 10 + 10 - 2$$

$$= 18$$

$$t_{0,05;18} = 2,101$$

3. H_0 diterima apabila : $-2,101 \leq t \leq 2,101$

H_0 ditolak apabila : $t < -2,101$ atau $t > 2,101$

4. $S_1 = 0,02$

$$S_2 = 0$$

$$n_1 = n_2 = 10$$

$$t = \frac{0,987 - 1}{\sqrt{\frac{9,0^2 + 9,0,02^2}{18} \left(\frac{1}{10} + \frac{1}{10} \right)}}$$

$$t = -2,05$$

5. Karena nilai t , yaitu $-2,05$ terletak diantara $-2,101$ dan $2,101$ maka dapat disimpulkan H_0 diterima sehingga tidak ada perbedaan antara *output* simulasi dengan *output* nyata. Oleh karena itu, model simulasi persediaan *spare part* dapat dinyatakan *valid*.

Pengujian *independent t-test* juga dilakukan untuk mengetahui perbandingan antara IL rata-rata *output* simulasi dengan nilai IL rata-rata keadaan sistem nyata.

Pengujian *Independent t-test* sebagai berikut :

1. Penentuan H_0 dan H_1

H_0 : $\mu_1 = \mu_2$ = Tidak ada Perbedaan antara simulasi sistem dengan keadaan nyata

V_s

H_1 : $\mu_1 \neq \mu_2$ = Ada perbedaan antara simulasi dengan keadaan nyata

2. $\alpha = 0,05$

$$df = 10 + 10 - 2 \\ = 18$$

$$t_{0,05;18} = 2,101$$

3. H_0 diterima apabila : $-2,101 \leq t \leq 2,101$

H_0 ditolak apabila : $t < -2,101$ atau $t > 2,101$ $S_1 = 52,75$

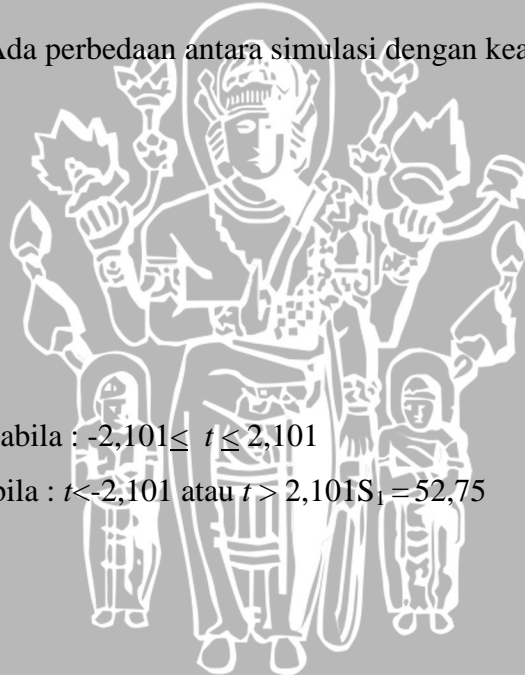
4. $S_2 = 50,80$

$$n_1 = n_2 = 10$$

$$t = \frac{112,5 - 99,44}{\sqrt{\frac{9.52,75^2 + 9.50,80^2}{18} \left(\frac{1}{10} + \frac{1}{10} \right)}}$$

$$t = 0,563$$

5. Karena nilai t , yaitu $0,563$ terletak diantara $-2,101$ dan $2,101$ maka dapat disimpulkan H_0 diterima sehingga tidak ada perbedaan antara *output* simulasi



dengan *output* nyata. Oleh karena itu, model simulasi persediaan *spare part* dapat dinyatakan *valid*.

4.9 Analisa Hasil Simulasi

Dari hasil keluaran simulasi diatas, maka simulasi sistem persediaan *spare part* yang telah dirancang memiliki validitas dengan sistem nyatanya. Hasil keluaran simulasi menyatakan bahwa nilai *service level* yang dihasilkan dari sistem menunjukkan angka 1 yang berarti nilai *service level* yang ada telah mencapai nilai 100% namun, dengan nilai IL rata-rata yang masih terlalu tinggi. Oleh sebab itu, setelah peneliti melakukan uji validitas terhadap sistem, dan menyatakan bahwa pemodelan simulasi sistem persediaan *spare part* adalah valid, maka langkah selanjutnya adalah merancang eksperimen dengan memasukan nilai-nilai parameter yaitu nilai Q dan ROP. Perancangan eksperimen yang digunakan menggunakan “*what if analysis*” suatu analisa untuk mengetahui apakah yang terjadi apabila atau jika suatu sistem diterapkan.

Dalam perancangan eksperimen sistem persediaan *spare part* perancangan eksperimen memiliki tujuan untuk mengetahui sistem mana yang menghasilkan nilai IL rata-rata yang optimal, sehingga sistem yang ada dapat dijadikan sebagai usulan dalam menentukan nilai ROP, nilai Q dan nilai IL awal pada perusahaan.

Dengan menggunakan *what if analysis*, maka dilakukan perancangan eksperimen dengan menggunakan 3 sistem yaitu sistem X, sistem Y dan sistem Z. Perancangan eksperimen yang dilakukan ditunjukkan pada tabel berikut ini :

Tabel 4.8. Perancangan Eksperimen Simulasi Sistem Persediaan

No	Kode <i>Spare part</i>	Sistem X			Sistem Y			Sistem Z		
		IL awal	Nilai ROP	Nilai Q	IL awal	Nilai ROP	Nilai Q	IL awal	Nilai ROP	Nilai Q
1	A	32	12	20	35	13	22	31	11	20
2	B	80	30	50	83	27	56	82	29	53
3	C	77	25	52	90	31	59	87	29	58
4	D	60	20	40	67	23	44	56	19	37
5	E	102	42	60	103	39	64	96	38	58
6	F	30	10	20	33	11	22	31	8	23
7	G	40	13	17	42	12	30	37	14	23
8	H	45	15	30	50	17	33	39	13	26
9	I	90	35	65	89	29	60	91	32	59
10	J	56	18	38	50	21	29	53	19	34

Dari perancangan diatas, peneliti ingin mengetahui nilai IL rata-rata dan *service level* yang dihasilkan dengan menggunakan perancangan eksperimen seperti pada tabel 4.8.

4.9.1 Hasil eksperimen dan Pengujian Hasil Simulasi

4.9.1.1 Hasil Eksperimen

Dari perancangan eksperimen yang telah direncanakan pada tabel diatas selanjutnya adalah menjalankan simulasi sesuai dengan nilai IL awal, nilai ROP dan nilai Q dalam sistem X, sistem Y, dan sistem Z. Dengan menjalankan simulasi selama 1460 hari dengan replikasi sebanyak 10 kali didapatkan hasil nilai *service level* yaitu pada seluruh *spare part* nilai *service level* yang dihasilkan 100% dan IL rata-rata yang dihasilkan terdapat pada lampiran 2. Selanjutnya dilakukan uji ANOVA dan uji BNT untuk mengetahui perbedaan antara sistem X, sistem Y dan sistem Z.

4.9.1.2 Uji ANOVA Hasil Eksperimen

Dari hasil menjalankan simulasi selama 1460 hari, dengan replikasi sebanyak 10 kali didapatkan nilai IL rata-rata dan nilai *service level*. Pada percobaan yang telah dilakukan didapatkan nilai *service level* mencapai nilai 100% pada setiap sistem. Oleh sebab itu tidak dilakukan uji ANOVA pada hasil *service level*.

Pada hasil simulasi, IL rata-rata pada setiap sistem memiliki perbedaan, oleh sebab itu dilakukan uji ANOVA untuk mengetahui perbedaan yang terjadi pada setiap sistem yang diterapkan. Adapun langkah-langkah pengujian ANOVA adalah sebagai berikut:

1. Hipotesis

$$H_0 : \mu_1 = \mu_2 = \dots = \mu_k = \dots = \mu_f = \text{Tidak Terdapat Perbedaan diantara sistem X, Y dan Z}$$

V_s

$$H_1 : \mu_1 \neq \mu_2 = \dots = \mu_k = \dots = \mu_f = \text{Terdapat Perbedaan diantara sistem X, Y dan Z}$$

2. Nilai $\alpha = 0,05$ $F_{0,05;3;26} = 2,89$

3. Karakteristik Pengujian

$$H_0 \text{ diterima apabila } F_0 \leq 2,89$$

$$H_0 \text{ ditolak apabila } F_0 \geq 2,89$$

4. Hasil Perhitungan ANOVA

Tabel 4.9. Tabel ANOVA

Sumber variasi	dk	jk	kt	f
----------------	----	----	----	---



Rata-rata	1	18620,72	18620,72	
Sistem	3	86,17593	28,72531	4,197475
Kekeliruan	27	184,7738	6,843473	
	30	18891,67		

5. Kesimpulan

Pada tabel ANOVA setelah perhitungan ANOVA dilakukan didapatkan hasil bahwa f hitung = 4,197 yang lebih besar dari F tabel yaitu 2,89 maka kesimpulan yang dapat diambil adalah terjadi perbedaan yang signifikan antara sistem X, sistem Y, dan sistem Z.

Pengujian ANOVA dilakukan pada nilai IL rata-rata seluruh *spare part* dalam penelitian yang ditunjukkan pada tabel 4.11 berikut ini:

Tabel 4.10. Pengujian ANOVA nilai IL rata-rata *spare part* kode B sampai dengan *spare part* kode J.

Kode Spare Part	Nilai F hitung	Nilai F tabel	Keputusan	Kesimpulan
B	2,47	2,89	Diterima	Tidak Terdapat perbedaan
C	17,42	2,89	Ditolak	Terdapat perbedaan
D	55,65	2,89	Ditolak	Terdapat perbedaan
E	6,1	2,89	Ditolak	Terdapat perbedaan
F	1,25	2,89	Diterima	Tidak Terdapat perbedaan
G	4,55	2,89	Ditolak	Terdapat perbedaan
H	35,22	2,89	Ditolak	Terdapat perbedaan
I	0,06	2,89	Diterima	Tidak Terdapat perbedaan
J	4,57	2,89	Ditolak	Terdapat perbedaan

Setelah melakukan uji F atau Uji Anova dan didapatkan kesimpulan yang berbeda-beda pada setiap *spare part* yang diuji, maka langkah selanjutnya adalah melakukan uji BNT (Beda Nyata Terkecil) untuk mengetahui tingkat kesignifikan perbedaan rata-rata antara sistem X dan sistem Y, sistem X dan sistem Z, dan sistem Y dan sistem Z. Adapun langkah-langkah uji BNT adalah sebagai berikut:

1. Beda Nyata terkecil

$$Vard = 6561,39, n = 30, k=3, v = 30 - 3 = 27$$

2. Statistik Uji

$$t_{ijk} = \sqrt{Vard * 1/n}$$

$$t_{ijk} = \sqrt{6,843 * 0,3}$$

$$= 1,439$$

3. Kriteria Pengujian

$$\alpha = 0,05, \quad v = 27, \quad t(0,05,27) = 2,05$$

4. Keputusan

5. Apabila nilai $t < -2,05$ atau $t > 2,05$ maka pengujian yang dilakukan adalah signifikan.

6. Pengujian

- Perbedaan antara sistem X dan sistem Y

$$t = -2,79 / 1,439 = -1,94$$

- Perbedaan antara sistem X dan sistem Z

$$t = 3,965 / 1,439 = 2,76$$

- Perbedaan antara sistem Y dan sistem Z

$$t = 1,175 / 1,439 = 0,82$$

7. Kesimpulan

Dari hasil pengujian dapat diketahui nilai t pada perbedaan rata-rata antara sistem X dan sistem Y kurang dari 2,05 maka perbedaan yang ada tidak signifikan, perbedaan antara sistem X dan sistem Z nilai t lebih besar dari 2,05 maka perbedaan rata-rata anatar sistem X dan Z signifikan, dan untuk perbedaan antara sistem Y dan Z niali t yang dihasilkan kurang dari 2,05 maka perbedaan rata-rata antara sistem Y dan sistem Z tidak signifikan.

Dilakukan uji BNT atau uji setelah ANOVA pada seluruh *spare part* dalam penelitian ini dengan hasil yang ditunjukkan pada tabel 4.11.

Tabel 4.11. Uji BNT Untuk *Spare Part* Kode B sampai dengan J

NO	Kode Spare Part	Nilai t Perbedaan sistem X dan Y	Kesimpulan	Nilai t Perbedaan sistem X dan Z	Kesimpulan	Nilai t Perbedaan sistem Y dan Z	Kesimpulan
1	B	-2,2139	Signifikan	0,93676	Tidak Signifikan	1,27715213	Tidak signifikan
2	C	-5,5171	Signifikan	-4,579	Signifikan	0,93806764	Tidak signifikan
3	D	-2,3387	Signifikan	7,74007	Signifikan	10,07879622	Signifikan
4	E	-0,4323	Tidak signifikan	2,7867	Signifikan	3,218959599	Signifikan
5	F	1,4972	Tidak signifikan	-0,303	Tidak signifikan	1,194277324	Tidak signifikan
6	G	-1,1995	Tidak signifikan	1,79936	Tidak signifikan	2,998837391	Signifikan
7	H	-0,462	Tidak signifikan	7,02738	Signifikan	7,489364313	Signifikan
8	I	0,1874	Tidak signifikan	-0,1722	Tidak signifikan	0,35961658	Tidak signifikan
9	J	1,89073	Tidak signifikan	0,43144	Tidak signifikan	2,322172906	Signifikan

4.10 Analisa Hasil Eksperimen.

Setelah dilakukan perancangan eksperimen terhadap sistem dengan menjalankan simulasi selama 1460 hari dan dilakukan replikasi selama 10 kali, hasil yang didapatkan adalah:

1. Nilai *service level* dari penerapan sistem X, sistem Y dan sistem Z dalam perancangan eksperimen menunjukkan nilai 1 % yang berarti bahwa *service level* yang ada mencapai nilai 100 %. Sistem Q yang diterapkan tidak mengalami *shortage* hal ini menunjukkan bahwa ketersediaan *spare part* memiliki performa yang bagus atau memiliki kehandalan sistem. Dengan adanya ketersediaan *spare part* yang handal maka hal ini dapat mengurangi *down time* mesin.
2. Nilai IL rata-rata yang dihasilkan terjadi perbedaan antara sistem X, sistem Y dan sistem Z. Untuk menguji apakah terdapat perbedaan diantara sistem, maka

dilakukan uji ANOVA, dengan uji ANOVA menunjukkan bahwa sistem X, Y dan Z memiliki perbedaan. Untuk mengetahui tingkat kesiginifikan perbedaan yang ada dilakukan uji BNT, dan dari hasil uji BNT yang dilakukan menunjukkan bahwa perbedaan yang ada signifikan.

3. Didapatkan usulan sistem persediaan *spare part* untuk menentukan nilai Q, nilai ROP. Dengan menjalankan simulasi didapatkan nilai IL rata-rata yang dapat dijadikan sebagai usulan perbaikan untuk mengurangi *Inventory Level* di perusahaan. Sistem Persediaan yang dapat dijadikan sebagai usulan perbaikan pada perusahaan dapat terlihat pada tabel 4.12.

Tabel 4.12. Sistem Persediaan Usulan dengan menggunakan “*what if analysis*”

No	Kode <i>Spare part</i>	Nilai Q	Nilai ROP	Nilai IL rata-rata
1	A	20	12	25
2	B	50	30	72
3	C	52	25	66
4	D	37	19	42
5	E	58	38	75
6	F	20	10	20
7	G	23	14	31
8	H	26	13	31
9	I	60	29	76
10	J	34	19	36

Pada tabel 4.12. didapatkan sistem persediaan yang dapat dijadikan sebagai usulan terhadap perusahaan untuk mengefisiensikan *inventory value spare part*. Dari nilai usulan perbaikan sistem persediaan yang diberikan, selanjutnya adalah membandingkan *inventory value* antara sistem yang ada di perusahaan dengan nilai yang diusulkan. Perbandingan *inventory value* antara sistem yang ada di perusahaan dengan sistem persediaan yang dijadikan sebagai usulan ditunjukkan pada tabel 4.13.

Pada tabel 4.13. terlihat bahwa sistem yang ada saat ini memberikan *inventory value* yang tinggi, dibandingkan dengan sistem persediaan usulan, PT. Petrokimia dapat meningkatkan efisiensi terhadap *inventory value* yang ada. Hal ini disebabkan IL rata-rata yang ada lebih kecil dibandingkan dengan nilai IL yang ada di perusahaan.

Tabel 4.13. Perbandingan *Inventory Value* antara sistem nyata dengan hasil simulasi

N o	Kode Spare part	IL data	<i>Inventory Value</i>	Total <i>Inventory Value</i> Awal	IL Usulan	Total <i>Inventory Value</i> usulan	Efisiensi
1	A	41	543670	Rp 22.290.470,00	25	Rp 13.591.750,00	Rp 8.698.720,00
2	B	152	88068,8	Rp 13.386.450,00	72	Rp 6.340.950,00	Rp 7.045.500,00
3	C	184	162061	Rp 29.819.224,00	66	Rp 10.696.026,00	Rp 19.123.198,00
4	D	102	463000	Rp 47.226.000,00	42	Rp 19.446.000,00	Rp 27.780.000,00
5	E	168	265183	Rp 44.550.703,68	75	Rp 19.888.707,00	Rp 24.661.996,68
6	F	32	265183	Rp 8.485.848,32	20	Rp 5.303.655,20	Rp 3.182.193,12
7	G	64	175000	Rp 11.200.000,00	31	Rp 5.425.000,00	Rp 5.775.000,00
8	H	110	6500	Rp 715.000,00	31	Rp 201.500,00	Rp 513.500,00
9	I	144	617700	Rp 88.948.800,00	76	Rp 46.945.200,00	Rp 42.003.600,00
10	J	128	360250	Rp 46.112.000,00	36	Rp 12.969.000,00	Rp 33.143.000,00

BAB V PENUTUP

5.1 Kesimpulan

Dari penelitian yang dilakukan didapatkan kesimpulan sebagai berikut :

1. Perencanaan Pengendalian Sistem Persediaan *Spare Part* dengan menggunakan simulasi program komputer dengan pendekatan *compound poisson process*. Sistem Persediaan yang ada dimodelkan dengan menggunakan model matematis, dimana dalam model matematis terdapat beberapa komponen yaitu :

4. *Input System*

Dalam sistem input data terdiri dari *decision input* yaitu nilai Q nilai jumlah kuantitas yang harus dipesan dan nilai ROP nilai yang menunjukkan jumlah persediaan barang di gudang yang menyebabkan harus melakukan pesan kembali , untuk input data adalah nilai IL awal (*stok on hand*) dan nilai probabilistik *demand*.

5. *Component System*

Komponen yang terdapat didalam Sistem *Persediaan Spare part* dengan Pendekatan *Compound Poisson Process* yaitu :

- *Compound Poisson Process* yaitu Fungsi $x = -\beta \ln(1 - u)$ dan Fungsi $x = \ln(u \cdot e^{-x} \cdot x!) / \lambda$
- *Incidental Demand* = Demand + N, $N \approx$ *Compound Poisson Process* (fungsi β dan fungsi λ)
- Pengadaan Barang Datang = Q + *Lead Time*
- Pemesanan kembali = ROP
- *Served* = *Served* - IL
- *Shortage* = *Shortage* + (N - IL)

6. *Output System*

Output dari sistem persediaan *Spare part* adalah

- nilai IL
$$IL = IL / T_{\text{Simulasi}}$$
- nilai service level

$$\text{Service Level} = \text{Served} / \text{Demand}$$

2. Dengan menggunakan “*what if analysis*” ditentukan sistem untuk menentukan nilai ROP, nilai Q dan IL rata-rata yang dihasilkan sebagai berikut:
Untuk *spare part* Kode A nilai Q 20 unit, ROP 12 unit dan IL rata-rata 25unit.
Spare part kode B nilai Q 50 unit, nilai ROP 30 unit, nilai IL rata-rata 72 unit.
Spare part kode C nilai Q 52 unit, nilai ROP 25 unit, nilai IL rata-rata 66 unit.
Spare part kode D nilai Q 37 unit, nilai ROP 19 unit, nilai IL rata-rata 42 unit.
Spare part kode E nilai Q 58 unit, nilai ROP 38 unit, nilai IL rata-rata 75 unit.
Spare part kode F nilai Q 20 unit, nilai ROP 10 unit, nilai IL rata-rata 20 unit.
Spare part kode G nilai Q 23 unit, nilai ROP 14 unit, nilai IL rata-rata 31 unit.
Spare part kode H nilai Q 26 unit, nilai ROP 13 unit, nilai IL rata-rata 31 unit.
Spare part kode I nilai Q 60 unit, nilai ROP 29 unit, nilai IL rata-rata 76 unit.
Spare part kode J nilai Q 34 unit, nilai ROP 19 unit, nilai IL rata-rata 36 unit.
3. Dengan metode persediaan sistem Q yang disimulasikan dengan menggunakan program komputer maka dapat mengurangi waktu *down time* mesin, hal ini disebabkan dari hasil simulasi menyatakan bahwa sistem persediaan yang telah dibuat telah memiliki nilai *service level* 100% yang artinya tidak terjadinya *shortage*, sehingga persediaan yang ada mampu melayani permintaan *spare part* dari pihak *maintenance*.

5.2 Saran

Setelah melakukan penelitian di P.T. Petrokimia Gresik, penulis menyarankan hal-hal sebagai berikut :

1. Dalam merencanakan sistem persediaan pihak pengadaan dapat menggunakan metode simulasi untuk merencanakan sistem persediaan yang akan datang. Hal ini disebabkan simulasi merupakan suatu eksperimen semu yang dapat menggambarkan keadaan nyata yang terjadi.
2. Untuk penelitian selanjutnya disarankan untuk membandingkan sistem persediaan antara sistem P dan sistem Q
3. Untuk penelitian selanjutnya disarankan untuk menambahkan animasi dari sistem yang telah dibuat..

4. DAFTAR PUSTAKA

- 5.
6. Alifen, Ratna. 2000. *Analisa "what if" sebagai Metode Antisipasi Keterlambatan durasi proyek*. Jurnal Teknik Sipil Universitas Kristen Petra.
- 7.
8. Alkaff, Abdullah. 1992. *Teknologi Keandalan Sistem*. Penerbit Fakultas Teknik Industri, ITS. Surabaya.
- 9.
10. Asmungi. 2006. *Simulasi Komputer Sistem Diskrit*. Andi. Yogyakarta
11. Assauri, Sofyan. 1999. *Manajemen Produksi*. Penerbit FE UI. Jakarta.
12. Daellenbach, Hans G. 1995. *System and Decision Making*. Willey. Singapura.
13. Dhillon, BS. 2002. *Engineering Maintenance*. CRC Press. New York
14. Fabrycky, W.J., Thuesen, G.J. and Verma, D. 1998. *"Economic Decision Analysis"*. Prentice Hall Int. Inc.
- 15.
16. Hakim, Arman. 1999. *Perencanaan dan Pengendalian Produksi*. Guna Widjaya. Jakarta
17. Halvorson, Michael. 2000. *Step by Step Microsoft Visual Basic 6.0*. PT. Elex Media Komputindo. Jakarta
- 18.
19. Hasan, M. Iqbal. 2000. *Statistik 2*. Bumi Aksara . Jakarta
20. Kelton, W. David. 2003. *Simulation Modeling and Analysis*. McGraw – Hill, Inc. New Jersey.
- 21.
22. Kandeep, Sumar. 2004. *Spare Parts Management An IT Automation Perspective*. Jurnal Teknologi.
- 23.
24. O'Connor, D.T. Patrick. 2000. *Practical Reliability Engineering*, Penerbit John Wiley and Sons Ltd, London.
- 25.
26. Rangkuti, Fredy. 1996. *Manajemen Persediaan*. Raja Grafindo Persada. Jakarta

27. Ross, Sheldon. 1996. *Stochastic Processes*. John Wiley & Sons Inc. New York
28. Sudjana. 1994. *Desain dan Analisis Eksperimen*. Tarsito. Bandung
29. Supranto, J. 1993. *Statistik Aplikasi dan Teori*. PT Gelora Aksara Pratama. Jakarta.
30. Setiawan, Sandi. 1991. *Simulasi*. ANDI OFFSET. Yogyakarta.

31.

32.

33.

34.

35.

36.

37.

38.

39.

