

**PERANCANGAN MODEL SIMULASI SPBE (STASIUN
PENGISIAN *BULK* ELPIJI) STANDAR PERTAMINA
DENGAN MENGGUNAKAN PROMODEL**

SKRIPSI

**EKO PARDIANTO
04 04 070255**



**UNIVERSITAS INDONESIA
FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INDUSTRI
DEPOK
JULI 2008**

**PERANCANGAN MODEL SIMULASI SPBE (STASIUN PENGISIAN
BULK ELPIJI) STANDAR PERTAMINA DENGAN MENGGUNAKAN
PROMODEL**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

EKO PARDIANTO

04 04 070255



**DEPARTEMEN TEKNIK INDUSTRI
FAKULTAS TEKNIK
UNIVERSITAS INDONESIA
DEPOK
JULI 2008**

HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.**

Nama : Eko Pardianto

NPM : 04 04 070255

Tanda Tangan :

Tanggal : 9 Juli 2008

LEMBAR PENGESAHAN

Skripsi ini diajukan oleh

Nama : Eko Pardianto
NPM : 0404070255
Program Studi : Teknik Industri
Judul Skripsi : PERANCANGAN MODEL SIMULASI SPBE
STANDAR PERTAMINA DENGAN PROMODEL

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana pada Program Sarjana Fakultas Teknik Universitas Indonesia

DEWAN PENGUJI

Pembimbing : Armand Omar Moeis, ST., MSc
Penguji : Dr. Ir. Teuku Yuri MZ, M. Eng. Sc
Penguji : Ir. Boy Nurtjahyo M, MSIE

Depok, 9 Juli 2008

(Armand Omar Moeis, ST., MSc)

UCAPAN TERIMAKASIH

Segala puji hanya milik ALLAH sebagai Tuhan semesta alam yang telah memberikan pertolongan dan dukungan kepada penulis dalam menyelesaikan skripsi ini tepat pada waktunya. Penulis juga menyadari bahwa tanpa bimbingan, bantuan, motivasi, inspirasi, dan sumbangan pikiran dari berbagai pihak, baik secara langsung maupun tidak langsung sejak awal masa perkuliahan hingga penyusunan skripsi maka skripsi ini tidak akan dapat selesai. Oleh sebab itu, pada kesempatan ini penulis ingin mengungkapkan rasa terima kasih sebesar-besarnya kepada beberapa pihak:

1. Ibu (Alm.), Bapak, Mama, Priska, dan Aji yang telah mendoakan dan memberikan dukungan kepada penulis
2. Bapak Armand Omar Moeis, ST., MSc selaku dosen pembimbing skripsi yang telah banyak memberikan bantuan moral dan material kepada penulis
3. Bapak Ir. Akhmad Hidayatno., MBT selaku dosen pembimbing akademis
4. Bapak Hariyanto Salim (Ahau) selaku pembimbing pemodelan sistem
5. Bapak Dr. Ir. Teuku Yuri MZ, M. Eng. Sc dan bapak Ir. Boy Nurtjahyo M, MSIE selaku dosen penguji skripsi
6. Nastiti Harini (titi) yang telah mendoakan, memberi motivasi, dan inspirasi
7. Anggota “Jomblo Corner” (Aqqinaldo, Rian, Heri, dan Arie) atas persahabatan selama ini; Romad, Oka, Surip, Azis, Bejo, Ian, Ajeng, Adi, Nuri, Novan, Ipeh, Fahmi, serta semua teman-teman Teknik Industri angkatan 2004 atas motivasi dan dukungan kepada penulis
8. Semua dosen dan karyawan Teknik Industri Universitas Indonesia

Semoga ALLAH *azza wa jalla* membalas semua kebaikan kalian semua. Dan semoga skripsi ini memberikan manfaat bagi dunia ilmu pengetahuan. Aamiin.

Depok, 2 Juli 2008

Penulis

**LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS
(Hasil Karya Perorangan)**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Eko Pardianto
NPM/NIP : 0404070255
Program Studi : Teknik Industri
Fakultas : Teknik
Jenis karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Non-Eksklusif** (*Non-exclusive Royalty-Free Right*) atas karya ilmiah saya yang berjudul :

PERANCANGAN MODEL SIMULASI SPBE STANDAR PERTAMINA
DENGAN MENGGUNAKAN PROMODEL

beserta perangkat yang ada (bila diperlukan). Dengan Hak Bebas Royalti Non-Eksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelolanya dalam bentuk pangkalan data (*database*), mendistribusikannya, dan menampilkan/mempublikasikannya di Internet atau media lain untuk kepentingan akademis tanpa perlu meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta. Segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah ini menjadi tanggungjawab saya pribadi.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok
Pada tanggal : 2 Juli 2008
Yang menyatakan

(Eko Pardianto)

RIWAYAT HIDUP PENULIS

Nama : Eko Pardianto
Tempat, Tanggal Lahir : Jakarta, 5 Februari 1986
Alamat : Jl. Balai Rakyat 1 No. 23^a Rt.014 Rw.07
Kelurahan Utan Kayu Utara Kecamatan
Matraman Jakarta Timur 13120

Pendidikan:

1.	SD	SD Negeri 07 Pagi Utan Kayu Utara	1991 - 1997
2.	SLTP	SLTP Negeri 7 Jakarta	1997 - 2000
3.	SMU	SMU Negeri 31 Jakarta	2000 - 2003
4.	S-1	Departemen Teknik Mesin Universitas Indonesia	2003 - 2004
5.	S-1	Departemen Teknik Industri Universitas Indonesia	2004 - 2008

ABSTRAK

Nama : Eko Pardianto
Program studi : Teknik Industri
Judul : PERANCANGAN MODEL SIMULASI SPBE STANDAR
PERTAMINA DENGAN MENGGUNAKAN PROMODEL

Kenaikan harga minyak dunia hingga di atas USD 100 per *barrel* telah memaksa pemerintah untuk mempercepat program konversi minyak tanah ke gas LPG. Percepatan program konversi ini telah mendorong peluang investasi yang massif dalam bisnis pembangunan SPBE (Stasiun Pengisian *Bulk* Elpiji) khususnya untuk pengisian tabung gas LPG 3 kg. Pertamina pun telah mengeluarkan standar SPBE. Namun, para pengusaha perlu mengetahui animasi bisnis proses SPBE yang dapat menampilkan proses di SPBE yang sebenarnya secara jelas dan baik. *Output* di SPBE dipengaruhi oleh faktor-faktor eksternal seperti fluktuasi permintaan tabung gas, tingkat kedatangan truk agen, dan pasokan gas. Berbagai fluktuasi tersebut mendorong pengusaha untuk mengetahui komposisi penggunaan sumberdaya yang dimiliki SPBE yang dapat menghasilkan tingkat *output* tertinggi. Oleh sebab itu, tujuan dari penelitian ini adalah membuat model simulasi sistem SPBE standar Pertamina guna mengetahui penggunaan sumberdaya yang menghasilkan tingkat *output* tertinggi dalam menghadapi fluktuasi permintaan tabung dan pasokan gas; agar tidak terjadi kelangkaan distribusi tabung gas.

Penelitian ini dimulai dengan penyusunan konsep model simulasi SPBE dengan menggunakan *flowchart* dan diagram IDEFØ. Kemudian dilakukanlah pengumpulan data struktural dan data operasional SPBE. Sedangkan data kuantitatif SPBE dikumpulkan dengan *time study* untuk kemudian diuji kecukupan data dan akhirnya ditentukan persebaran distribusi data kuantitatif tersebut dengan aplikasi Stat::Fit dalam ProModel. Perancangan model simulasi SPBE dibuat dengan memasukkan semua data yang telah dikumpulkan sebelumnya. Perancangan model simulasi menggunakan *software* ProModel karena fitur ProModel yang *user friendly* sehingga mudah digunakan, kemampuan ProModel memodelkan sistem yang memiliki variabilitas dan komponen yang saling bergantung, serta ProModel dapat menampilkan animasi proses yang jelas secara visual. Setelah dirancang, model simulasi SPBE harus melalui uji verifikasi dan validasi agar menjamin bahwa model simulasi SPBE ini memang mewakili proses yang terjadi pada SPBE yang sebenarnya.

Dengan model simulasi SPBE yang telah lulus uji verifikasi dan validasi, maka akan dilakukan eksperimen dengan serangkaian skenario yang nantinya akan didapat penggunaan sumberdaya SPBE yang dapat menghasilkan tingkat *output* tertinggi.

Kata kunci:
Simulasi, SPBE, ProModel, *Output*

ABSTRACT

Name : Eko Pardianto
Study Program: Industrial Engineering
Title : SIMULATION MODEL DESIGN OF SPBE BASED ON
PERTAMINA STANDARDS USING PROMODEL

The increase of world petroleum price which is over USD 100 per barrel, has urged Indonesian government to accelerate the conversion program from petroleum to LPG. This conversion program acceleration has opened massive investment in building SPBE (LPG bulk Filling Station) business especially for 3 kg tubes filling. Pertamina has also made SPBE standards. Nevertheless, businessmen need to know SPBE business process animation which can display real SPBE process clearly and properly. Output in SPBE is influenced by many external factors which are like: fluctuation of gas tubes demand, arrival rate of agencies' trucks, and LPG bulk supplies. These fluctuations make businessmen to find resources owned by SPBE allocation which can yield highest output rate. To solve these problems, the purpose of this research is to design simulation model of SPBE based on Pertamina standards in order to find resources owned by SPBE allocation which can yield highest output rate in facing fluctuation of gas tubes demand, arrival rate of agencies' trucks, and LPG bulk supplies.

This research is initialized by designing simulation model concept of SPBE by using flowchart and IDEFØ diagram. Next, structural and operational data of SPBE are collected. While quantitative data are collected using time study method and then are tested to be adequate data and finally distribution probability of quantitative data are determined using Stat::Fit application in ProModel. Those data which are previously collected are using in building simulation model of SPBE. ProModel software is used in designing simulation model of SPBE because figures in ProModel are user friendly so that it's easy to use, ProModel also has capability to model a system which consists of variability and interdependency among its components, and also ProModel can display process animation visually. After simulation model of SPBE is built, it must pass verification and validation test to ensure that it represent real process in real SPBE.

By using simulation model of SPBE that has passed verification and validation test, experiment with several scenarios will be conducted. After the experiment is completed, we will find resources owned by SPBE allocation which can yield highest output rate.

Keywords:
Simulation, SPBE, ProModel, Output

DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN ORISINALITAS	ii
LEMBAR PENGESAHAN	iii
UCAPAN TERIMAKASIH	iv
LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS	v
RIWAYAT HIDUP PENULIS	vi
ABSTRAK	vii
ABSTRACT	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
DAFTAR LAMPIRAN	xiv
1. PENDAHULUAN	1
1.1. Latar Belakang Permasalahan	1
1.2. Diagram Keterkaitan Masalah	6
1.3. Perumusan Permasalahan	6
1.4. Tujuan Penelitian	7
1.5. Batasan Masalah	7
1.6. Metodologi Penelitian	7
1.7. Sistematika Penulisan	11
2. DASAR TEORI	12
2.1. Pemodelan Simulasi Sistem	12
2.1.1. Simulasi dengan <i>Software</i>	12
2.1.2. Dinamika Sistem	14
2.1.3. Pengumpulan Data Simulasi	16
2.1.4. Pembuatan Model Simulasi suatu Sistem	18
2.1.5. Verifikasi dan Validasi Model	19
2.1.6. Struktur Sistem Optimasi dengan Simulasi	20
2.2. Sistem Antrian	21
2.3. Diagram IDEF0	26
2.4. Statistik	29
2.4.1. Data dan Uji Kecukupan Data	29
2.4.2. <i>Sampling</i>	31
2.4.3. Probabilitas	33
2.4.4. Distribusi Probabilitas	34
2.4.5. Uji Kesesuaian Distribusi Probabilitas	42
3. PENGUMPULAN DAN PENGOLAHAN DATA	45
3.1. SPBE Dalam Jalur Distribusi Elpiji	45
3.2. Data Struktural	46
3.2.1. Layout SPBE	46
3.2.2. Fasilitas Operasional dan Biaya SPBE	47
3.3. Data Operasional	48
3.2.1. Bisnis Proses	48

3.3.2. Aliran Proses -----	49
3.4. Data Kuantitatif -----	53
3.4.1. Permintaan <i>Bulk</i> LPG -----	53
3.4.2. Jumlah dan Interval Kedatangan <i>Skid Tank</i> -----	54
3.4.3. Proses Pengisian <i>Storage Tank</i> -----	55
3.4.4. Permintaan Tabung LPG 3 Kg -----	57
3.4.5. Jumlah dan Interval Kedatangan Truk Agen -----	58
3.4.6. Proses Bongkar Tabung -----	60
3.4.7. Proses Penimbangan Tabung -----	61
3.4.8. Proses Pengisian Tabung -----	62
3.4.9. Proses Pemeriksaan Isi Tabung -----	62
3.4.10. Proses Koreksi Isi Tabung -----	63
3.4.11. Proses Uji Kebocoran Tabung -----	65
3.4.12. Proses Segel Tutup Tabung -----	66
3.4.13. Proses Muat Tabung -----	67
3.4.14. Proses Evakuasi Isi Tabung -----	68
4. PEMODELAN DAN EKSPERIMEN -----	70
4.1. Formulasi Model -----	70
4.1.1. Elemen Struktural -----	70
4.1.2. Elemen Operasional -----	72
4.2. Verifikasi Dan Validasi Model -----	87
4.2.1. Verifikasi Model -----	87
4.2.2. Validasi Model -----	88
4.3. Eksperimen -----	89
4.3.1. Kondisi Awal Model -----	89
4.3.2. Rancangan Skenario -----	90
4.3.3. Analisa Hasil Skenario 1 -----	94
4.3.4. Analisa Hasil Skenario 2 -----	95
4.3.5. Analisa Hasil Skenario 3 -----	97
4.3.6. Analisa Hasil Skenario 4 -----	99
4.3.7. Analisa Perilaku Jumlah Mesin pada <i>Carousel</i> terhadap Semua Skenario -----	101
5. KESIMPULAN DAN SARAN -----	103
5.1. Kesimpulan -----	103
5.2. Saran -----	105
DAFTAR PUSTAKA -----	106
LAMPIRAN	

DAFTAR GAMBAR

Gambar 1.1 Jalur Distribusi LPG 3 Kg	4
Gambar 1.2 Diagram Keterkaitan Masalah	6
Gambar 1.3 Diagram Alir Metodologi Penelitian	9
Gambar 2.2 Struktur Sistem Optimasi dengan Simulasi	20
Gambar 2.3 Tipe Sistem Antrian	22
Gambar 2.4 Model Keputusan Antrian Berbasis-Biaya	25
Gambar 2.5 Kotak IDEFØ secara sederhana	27
Gambar 2.6 Diagram IDEFØ dari suatu model yang kompleks	28
Gambar 2.7 <i>Sampling</i>	31
Gambar 2.8 Grafik Distribusi Binomial	36
Gambar 2.9 Grafik Distribusi Poisson	37
Gambar 2.10 Grafik Distribusi Normal	38
Gambar 2.11 Grafik Distribusi Gamma	39
Gambar 2.12 Grafik Distribusi Eksponensial	40
Gambar 2.13 Grafik Distribusi Lognormal	41
Gambar 2.14 Grafik Distribusi Erlang	41
Gambar 3.1 Jalur Distribusi Tabung Elpiji 3 Kg	45
Gambar 3.2 Layout SPBE keseluruhan	46
Gambar 3.3 Area Pengisian Tabung Elpiji dan <i>Storage tank</i>	47
Gambar 3.4 Diagram Alir Pengadaan Elpiji di SPBE	48
Gambar 3.5 Diagram Alir Distribusi LPG Oleh Agen dari SPBE	49
Gambar 3.6 Diagram A-0 Pelaksanaan Operasional SPBE	50
Gambar 3.7 Peta Aliran Proses di SPBE	52
Gambar 3.8 Permintaan Harian <i>Bulk</i> LPG	54
Gambar 3.9 Permintaan Harian Tabung LPG 3 Kg	57
Gambar 4.1 <i>Layout</i> SPBE yang digunakan pada model simulasi	70
Gambar 4.2 <i>Skid tank</i> (Truk Tangki)	71
Gambar 4.3 Truk Agen (Pengangkut Tabung 3 kg Kosong)	71
Gambar 4.4 Truk Agen (Pengangkut Tabung 3 kg Isi)	71

Gambar 4.5 Tabung Gas 3 kg	71
Gambar 4.6 Diagram Alir Operasi SPBE	73
Gambar 4.7 Hasil Skenario 1	94
Gambar 4.8 Hasil Skenario 2	96
Gambar 4.9 Hasil Skenario 3	97
Gambar 4.10 Hasil Skenario 4	99
Gambar 4.11 Perilaku Jumlah Mesin pada <i>Carousel</i> terhadap Semua Skenario	102



DAFTAR TABEL

Tabel 2.1 Notasi Waktu Antar-Kedatangan-----	24
Tabel 2.2 Perbedaan antara Populasi dengan Sampel-----	31
Tabel 3.1 Permintaan Harian <i>Bulk</i> LPG -----	53
Tabel 3.2 Jadwal Kedatangan <i>Skid Tank</i> -----	55
Tabel 3.3 <i>Time Study</i> Proses Pengisian <i>Storage Tank</i> -----	56
Tabel 3.4 Permintaan Harian Tabung LPG 3 Kg-----	57
Tabel 3.5 Interval Kedatangan Tiap Satu Unit Truk Agen-----	58
Tabel 3.6 <i>Time Study</i> Proses Bongkar Tabung -----	60
Tabel 3.7 <i>Time Study</i> Proses Penimbangan Tabung -----	61
Tabel 3.8 <i>Time Study</i> Proses Pemeriksaan Isi Tabung-----	62
Tabel 3.9 <i>Time Study</i> Proses Koreksi Isi Tabung-----	64
Tabel 3.10 <i>Time Study</i> Proses Uji Kebocoran Tabung-----	65
Tabel 3.11 <i>Time Study</i> Proses Segel Tutup Tabung-----	66
Tabel 3.12 <i>Time Study</i> Proses Muat Tabung -----	67
Tabel 3.13 <i>Time Study</i> Proses Evakuasi Isi Tabung-----	68
Tabel 4.1 Kondisi Awal Model -----	89
Tabel 4.2 Hasil Skenario 1 -----	94
Tabel 4.3 Hasil Skenario 2 -----	95
Tabel 4.4 Hasil Skenario 3 -----	97
Tabel 4.5 Hasil Skenario 4 -----	99
Tabel 4.6 Perilaku Jumlah Mesin pada <i>Carousel</i> terhadap Semua Skenario -----	101
Tabel 5.1 Kondisi Awal Model Simulasi SPBE Standar Pertamina -----	103

DAFTAR LAMPIRAN

Lampiran 3.4 Uji Penentuan Kesesuaian Distribusi

Lampiran 4.1 Formulasi Model



1. PENDAHULUAN

1.1. Latar Belakang Permasalahan

Penyediaan energi di masa depan merupakan permasalahan yang senantiasa menjadi perhatian semua bangsa karena bagaimanapun juga kesejahteraan manusia dalam kehidupan modern sangat terkait dengan jumlah dan mutu energi yang dimanfaatkan. Bagi Indonesia yang merupakan salah satu negara sedang berkembang, penyediaan energi merupakan faktor yang sangat penting dalam mendorong pembangunan. Seiring dengan meningkatnya pembangunan terutama pembangunan di sektor industri, pertumbuhan ekonomi dan pertumbuhan penduduk, kebutuhan akan energi terus meningkat. Sampai saat ini, minyak bumi masih merupakan sumber energi yang utama dalam memenuhi kebutuhan di dalam negeri. Selain untuk memenuhi kebutuhan energi di dalam negeri, minyak bumi juga berperan sebagai komoditi penghasil penerimaan negara dan devisa. Peranan minyak bumi yang besar tersebut terus berlanjut, sedangkan cadangan semakin menipis. Di lain pihak harga minyak bumi sangat sulit untuk diperkirakan, sebagai akibat banyaknya faktor tak menentu yang berpengaruh. Selain itu, produksi bahan bakar minyak (BBM) yang dilakukan melalui teknologi transformasi di dalam negeri, tidak mencukupi kebutuhannya.

Kenaikan harga minyak bumi tingkat dunia yang telah mencapai tingkat harga tertinggi sepanjang sejarah yaitu menembus angka 100 USD per barrel telah membawa dampak ekonomi yang sangat merugikan. Tak bisa dipungkiri bahwa ketergantungan kebutuhan energi kepada minyak bumi di Indonesia sangat tinggi di berbagai sektor. Ketergantungan kepada minyak bumi yang demikian tinggi di Indonesia ini terjadi karena kekeliruan kebijakan pemerintah yang telah terjadi selama puluhan tahun. Selain menetapkan minyak bumi sebagai sumber energi utama selama puluhan tahun, pemerintah juga menerapkan sistem subsidi minyak bumi yang memiliki dampak tidak sehat bagi pertumbuhan ekonomi di masa depan dan juga bagi anggaran keuangan negara apalagi jika terjadi kenaikan harga minyak bumi tingkat dunia. Kekeliruan pemanfaatan minyak bumi ini hanya terjadi pada dua negara di dunia yaitu Indonesia dan satu negara di benua Afrika¹.

Dengan kecenderungan meningkatnya harga minyak bumi tingkat dunia, maka mengakibatkan juga meningkatnya pengeluaran pemerintah untuk memberikan subsidi bagi lebih dari 40 juta kilo liter Bahan Bakar Minyak (BBM) non industri yang salah satunya adalah minyak tanah untuk kebutuhan rumah tangga. Masalah kenaikan harga dan kelangkaan minyak bumi ini mendorong pemerintah Indonesia untuk segera mencari dan menetapkan pemakaian energi alternatif guna mengurangi ketergantungan kepada minyak bumi sekaligus menyelamatkan perekonomian dan kesehatan lingkungan. Saat ini beban anggaran pemerintah 75% masih bertumpu pada sektor perminyakan dan hanya 5% pada sektor gas², diharapkan penggunaan gas bisa lebih ditingkatkan lagi menjadi 20% untuk penghematan anggaran. Oleh sebab itu, Pemerintah berusaha menjadikan LPG (Liquified Petroleum Gas) sebagai bahan bakar pengganti minyak tanah terutama untuk sektor rumah tangga.

LPG (Liquified Petroleum Gas) atau yang biasa dikenal dengan sebutan Elpiji kini memasuki babak baru dalam pemanfaatannya di dalam negeri Indonesia. Selama ini dari konsumsi sebesar 100.000 metrik ton (MT) per bulan di tahun 2004, Elpiji lebih banyak digunakan oleh sektor rumah tangga (69 persen), lalu hotel berbintang dan restoran (13 persen), dan industri sebanyak 18 persen³. Sekadar gambaran, produksi LPG nasional saat ini tercatat kurang lebih 3,5 juta MT dan dari jumlah tersebut sebagian besar diekspor. Utilisasi yang rendah dari Elpiji terlihat jelas dari persentase rata-rata konsumsi per kapita di Indonesia yang hanya sebesar 0,5 persen⁴ dari jumlah penduduk. Malaysia padahal sudah mencapai lima persen dan Thailand sebesar dua persen dari jumlah penduduknya. Pemerintah akhirnya mendorong peningkatan konsumsi Elpiji nasional hingga mencapai 3 juta metrik ton melalui program pengalihan minyak tanah ke LPG.

Selama ini, PT Pertamina (Persero) adalah pemain satu-satunya di bisnis LPG. Jadi, jika kini produsen migas lainnya atau KKS (Kontrak Kerja Sama) di bawah

¹ Yusuf Kalla, *Puluhan Tahun Indonesia Keliru Manfaatkan Minyak Tanah*, 2007.
<<http://www.antara.co.id/arc/2007/8/28/wapres-puluhan-tahun-indonesia-keliru-manfaatkan-minyak-tanah>>

² “*Statistik Ekonomi Energi Indonesia 2004*”, Pusat Informasi Energi dan Sumber Daya Mineral, Jakarta, 2004

³ *Ibid.*

⁴ *Ibid.*

payung BP Migas menyuplai LPG untuk PLTG (Pembangkit Listrik Tenaga Gas) Gresik milik PLN, ini tentu menjadi babak baru ekstensifikasi bisnis LPG. Sumber gas bumi di Indonesia sebenarnya cukup besar, mencapai lebih dari 100 TCF (triliun kaki kubik). Cadangan itu terpecah di seluruh Indonesia, terutama di Sumatera bagian Utara, Sumatera bagian Tengah, Sumatera Selatan, Kalimantan Timur, dan Natuna, Jawa sebelah tenggara, Jawa Timur, dan Sulawesi Selatan. Sekitar 22 cekungan yang sebagian terbesar terletak di kawasan timur Indonesia belum dibor. Angka cadangan itu akan terus meningkat, karena bagian terbesar kawasan timur Indonesia baik darat maupun di lepas pantai, relatif masih banyak yang belum diselidiki. Dari sisi permintaan, gas bumi cukup potensial bahkan cenderung meningkat karena program pengalihan penggunaan minyak bumi ke gas bumi akibat kenaikan harga minyak bumi⁵.

Utilisasi LPG ternyata sangat luas, mulai dari kalangan rumah tangga, industri (pupuk, petrokimia, semen, pabrik baja), listrik, hingga transportasi. LPG tak hanya digunakan untuk memasak, tetapi juga untuk keperluan penerangan, *water heater*, *gas stove*, *rice cooker*, setrika, dan lain-lain. Dan secara umum, LPG ini digunakan untuk restoran, rumah sakit, laboratorium, industri, pabrik-pabrik, perusahaan keramik, dok, perkapalan, bengkel, dan lain-lain. LPG juga digunakan sebagai alat penekan pada industri yang menghasilkan produk seperti deodoran, minyak wangi, alat kosmetik, dan sebagainya. Selain itu, pada industri keramik, LPG digunakan sebagai alat bantu penyemprot cat keramik serta bahan bakar pemanas. Di bidang industri, produk LPG digunakan sebagai pengganti freon, aerosol, *refrigerant/cooling agent*, kosmetik, dan dapat pula dijadikan bahan baku produk khusus. LPG pun digunakan sebagai tenaga penggerak atau bahan bakar bagi kendaraan bermotor. Jadi singkatnya, industri LPG di Indonesia tetap memiliki prospek yang cerah⁶.

Elpiji dapat dihasilkan dari kilang Unit Pengolahan (UP) Direktorat Hilir atau kilang swasta, yaitu pihak swasta yang bekerjasama dengan Daerah Operasi Hulu (DOH) Direktorat Hulu. LPG juga dihasilkan dari gas bumi yang dihasilkan

⁵ Rudy Victor Sinaga, *Prospek Bisnis LPG menjanjikan*, 2007. < <http://www.sinarharapan.co.id> >

⁶ *Ibid.*

lapangan Attaka (Kalimantan Timur) dan lapangan Ardjuna (laut Jawa). Mengikuti meningkatnya konsumsi bahan bakar serta munculnya teknologi yang lebih canggih, LPG telah dihasilkan bukan saja oleh kilang minyak tetapi juga oleh kilang LPG. Kilang LPG tersebut dibangun di beberapa lapangan minyak antara lain di lapangan Rantau (Aceh), Tuga Barat dan Mundu (Jawa Barat), Arar (Irian Jaya), lepas pantai laut Jawa (di daerah kerja ARCO), dan di Tanjung Santan (Kalimantan Timur, dioperasikan oleh Union Oil). Pangsa pasar LPG sekitar 68 persennya dipegang oleh bagian Hilir Pertamina, 11 persen oleh Hulu Pertamina, dan 21 persen sisanya adalah total market produser swasta. LPG juga diekspor ke beberapa negara sahabat, seperti Jepang (pembeli terbesar LPG Indonesia dengan jumlah seluruhnya di tahun 1996 saja sebesar 2.100.000 ton per tahun), Singapura, Hongkong, dan negara lainnya. Di antara pembeli itu di antaranya Mitsubishi Corp, Marubeni, Sumitomo, Mitsui, Devon Energy, dan Petrede vv⁷.

Dalam pelaksanaan program konversi minyak tanah ke LPG ini, ternyata masalah distribusi merupakan hambatan terbesar. Seringkali terjadi saat minyak tanah ditarik, tetapi gas LPG belum tersedia. Memang pada tahap awal hanya Pertamina yang memiliki kemampuan untuk pendistribusian LPG. Akan tetapi, pemerintah juga memberikan peluang kepada swasta untuk berpartisipasi dalam pendistribusian LPG. Walaupun begitu, swasta yang berpartisipasi dalam pendistribusian LPG wajib memiliki persyaratan seperti harus memenuhi berbagai persyaratan teknis yang sesuai standar Pertamina, harus bisa memanfaatkan kilang-kilang dalam negeri, harus memiliki *storage*, dan harus memiliki *coverage day* agar tidak terjadi kelangkaan⁸.



Gambar 1.1 Jalur Distribusi LPG 3 Kg

⁷ *Ibid.*

⁸ Purnomo Yusgiantoro, *Ratas Bahas Distribusi LPG dan Kegiatan Usaha Panas Bumi*, 2007. <<http://www.presidensby.info/index.php/fokus/2007/10/04/2289.html>>

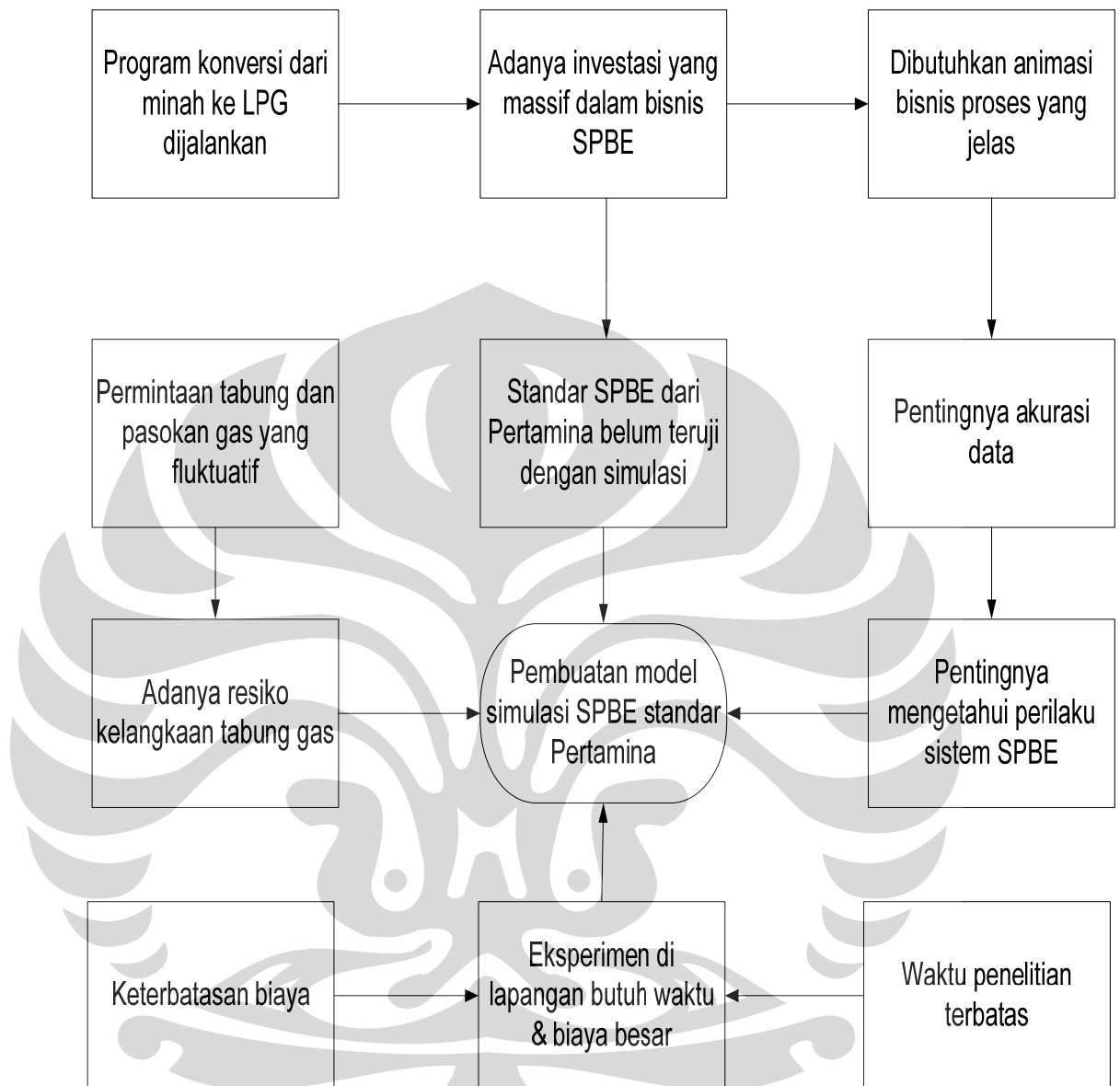
(Sumber: "Pedoman Pencacahan dan Distribusi Elpiji 3 kg" , No. 1688/F10000/2007-S3)

SPBE (Stasiun Pengisian *Bulk* Elpiji) yang berada di bagian hulu pendistribusian LPG memegang peranan yang menentukan kelancaran pendistribusian LPG ke konsumen. SPBE A merupakan salah satu swasta yang belum beroperasi dan ingin berpartisipasi dalam pendistribusian LPG. Agar proses pendistribusian LPG dapat berjalan lancar maka perlu ditentukan beberapa keputusan operasional dalam tahap desain SPBE A. Dalam SPBE A, akan terdapat dua proses utama yaitu proses pengisian *storage tank* dan proses pengisian tabung gas LPG 3 Kg. SPBE A harus mematuhi berbagai persyaratan teknis standar Pertamina. Akan tetapi, SPBE standar Pertamina belum teruji secara simulasi dan belum ada informasi apakah standar Pertamina telah memenuhi standar Internasional.

Dalam permasalahan ini, pendekatan simulasi dapat membantu SPBE A untuk mengetahui variabel-variabel yang mempengaruhi kinerja sistem SPBE dalam menentukan kapasitas sistem SPBE dengan animasi yang baik dan bisnis proses yang menggambarkan keadaan yang terjadi di lapangan. Pelaksanaan uji skenario dengan simulasi untuk berbagai kondisi yang mungkin terjadi akan memberikan gambaran mengenai kondisi sistem dalam SPBE A bila nantinya skenario tersebut benar-benar terjadi. Penelitian ini yang melibatkan banyak variabel yang saling memiliki ketergantungan dengan tingkat variabilitas yang tinggi seperti pada model proses pengisian gas LPG 3 Kg di SPBE A dapat dilakukan dengan lebih mudah menggunakan simulasi. Selain itu, penelitian ini merupakan kesempatan baik untuk menguji model SPBE standar Pertamina dengan pendekatan simulasi.

Sebagai tambahan, penggunaan simulasi lebih menghemat waktu dan biaya serta memiliki resiko yang lebih kecil karena simulasi hanya terjadi di dalam komputer dibandingkan dengan penerapan eksperimen langsung di sistem nyata yang sebenarnya. *Software* simulasi ProModel untuk simulasi jenis diskrit yang memiliki kelebihan diantaranya desain yang lebih *user friendly* dan memiliki animasi yang mewakili pergerakan sistem di alam nyata menjadi pilihan utama untuk digunakan dalam penelitian ini.

1.2. Diagram Keterkaitan Masalah



Gambar 1.2 Diagram Keterkaitan Masalah

1.3. Perumusan Masalah

Berdasarkan latar belakang dan diagram keterkaitan masalah yang telah dikemukakan pada bagian sebelumnya, perumusan masalah pada penelitian ini adalah menciptakan simulasi sistem SPBE standar Pertamina dengan animasi yang baik dan proses operasi yang mendekati operasional sebenarnya.

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah membuat model simulasi sistem SPBE standar Pertamina guna mengetahui penggunaan sumberdaya yang menghasilkan tingkat *output* tertinggi dalam menghadapi fluktuasi permintaan tabung dan pasokan gas; agar tidak terjadi kelangkaan distribusi tabung gas.

1.5. Ruang Lingkup Penelitian

Dalam mencapai tujuan penelitian, maka ruang lingkup penelitian ditentukan sebagai berikut:

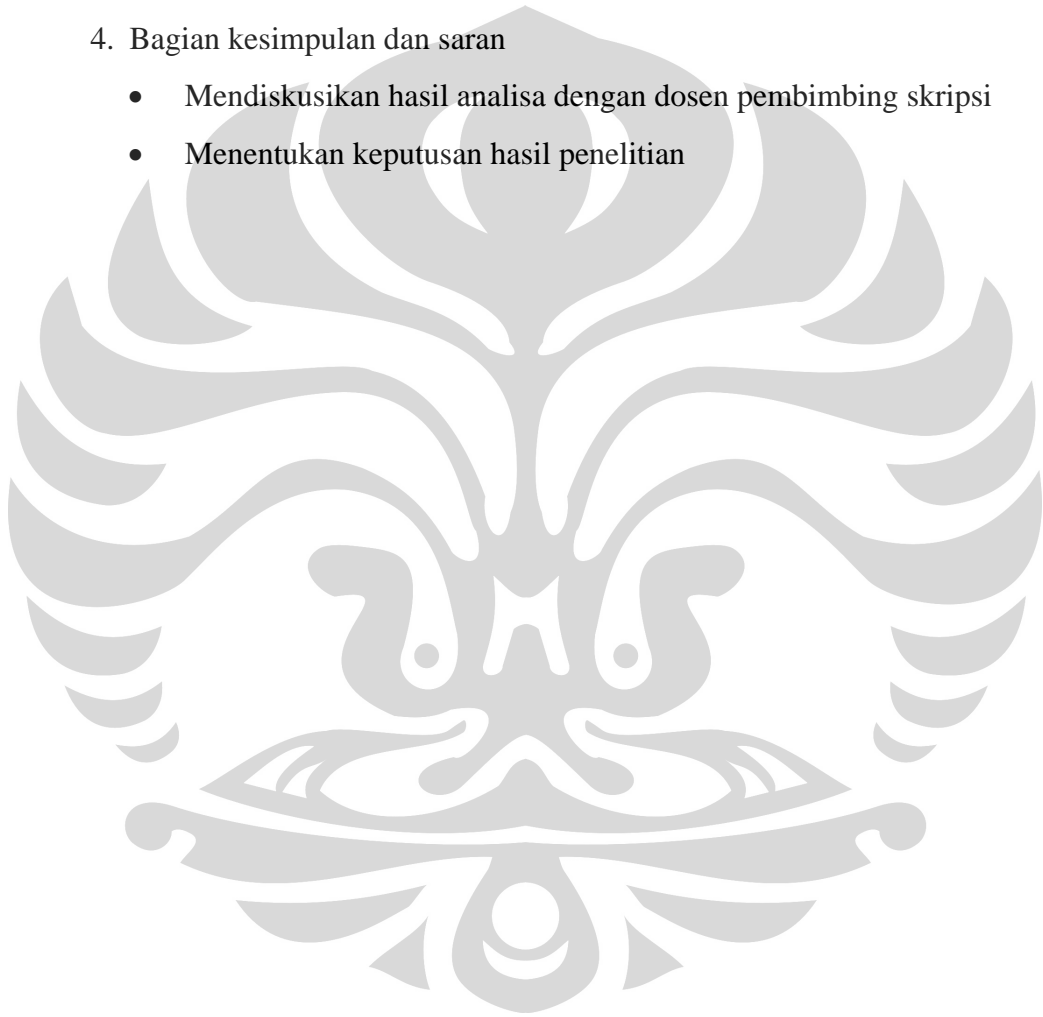
1. Proses yang akan disimulasikan ialah proses pengisian *storage tank* dan semua proses yang berkaitan dengan pengisian tabung LPG 3 Kg hingga keluar dari SPBE A
2. Tidak melibatkan perhitungan biaya yang terkait langsung dengan aktivitas operasional yang disimulasikan dan *resource* yang dicari optimalitasnya
3. Simulasi hanya terjadi dalam komputer tanpa diimplementasikan di lapangan

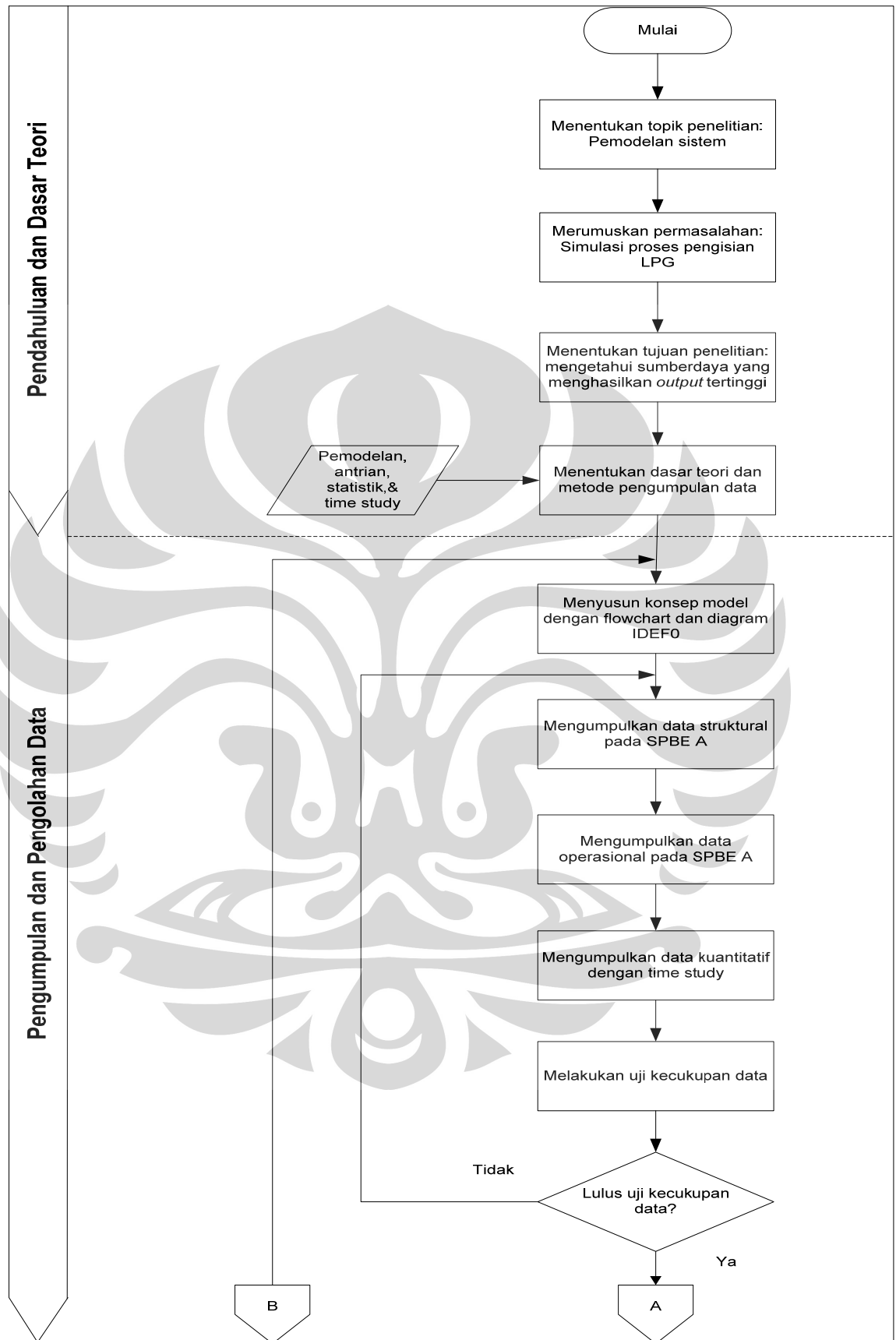
1.6. Metodologi Penelitian

Beberapa tahapan metodologi dalam penelitian ini sebagai berikut:

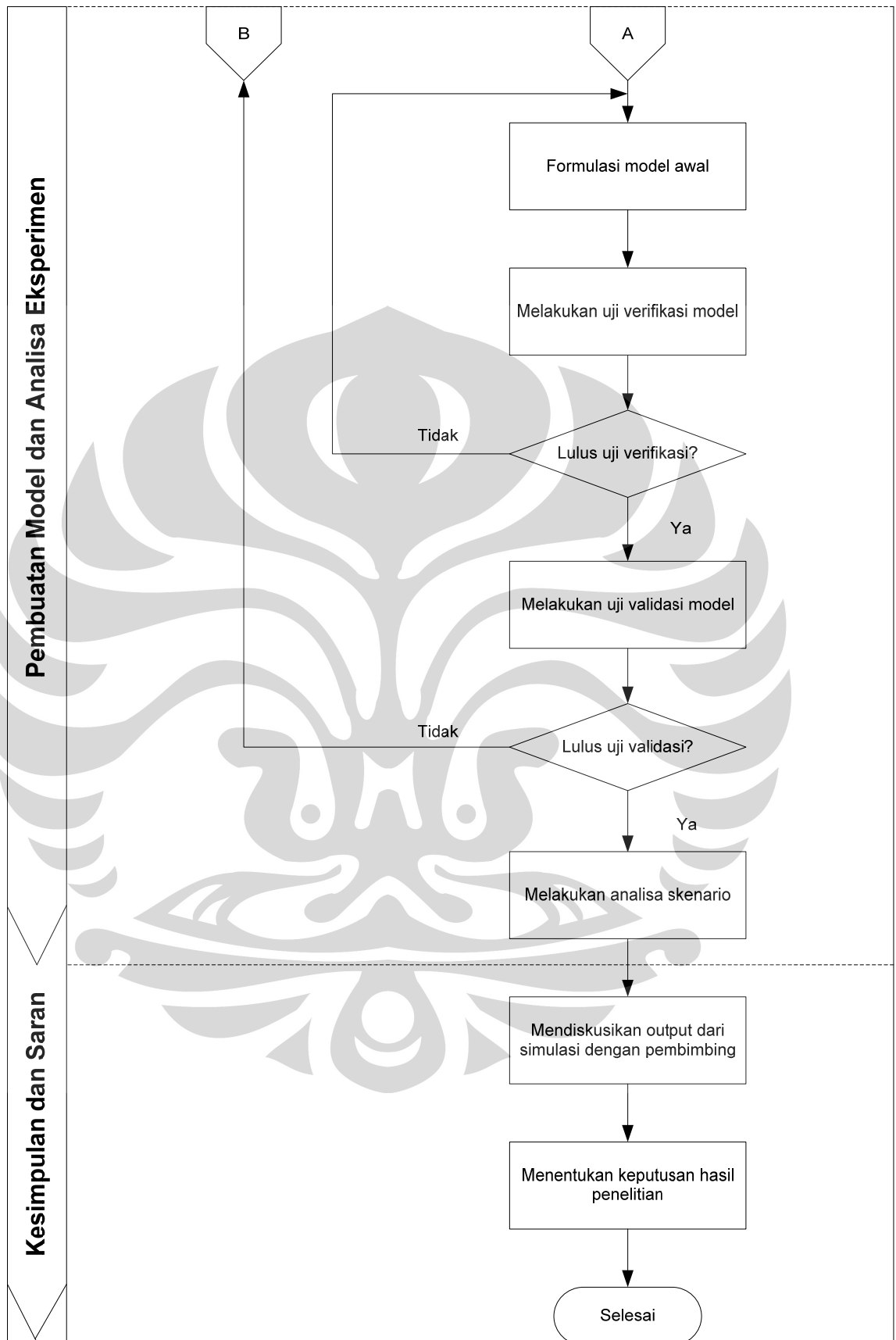
1. Bagian pendahuluan dan dasar teori
 - Menentukan topik penelitian yaitu pemodelan sistem
 - Merumuskan permasalahan yaitu simulasi proses pengisian LPG
 - Menentukan tujuan penelitian yaitu mengetahui penggunaan sumberdaya yang menghasilkan tingkat *output* tertinggi dalam menghadapi fluktuasi permintaan tabung dan pasokan gas
 - Menentukan dasar teori dan metode pengumpulan data. Dasar teori yang akan digunakan ialah pemodelan sistem, teori antrian, statistik, diagram IDEF0, dan *time study*.
2. Bagian pengumpulan dan pengolahan data
 - Menyusun konsep model dengan *flowchart* dan diagram IDEF0
 - Mengumpulkan data struktural dan data operasional pada SPBE A dan depo Pertamina

- Mengumpulkan data kuantitatif dengan *time study*
 - Melakukan uji kecukupan data
3. Bagian pembuatan model dan analisa eksperimen
- Membuat formulasi model simulasi
 - Melakukan uji verifikasi model
 - Melakukan uji validasi model
 - Melakukan analisa skenario simulasi
4. Bagian kesimpulan dan saran
- Mendiskusikan hasil analisa dengan dosen pembimbing skripsi
 - Menentukan keputusan hasil penelitian





Gambar 1.3 Diagram Alir Metodologi Penelitian



Gambar 1.3 Diagram Alir Penelitian (sambungan)

1.7. Sistematika Penulisan

Penelitian ini terdiri atas lima bab secara berurutan berdasarkan urutan sistematis dan kronologis proses pengerjaan penelitian yaitu bagian pendahuluan, bagian dasar teori, bagian pengumpulan dan pengolahan data, bagian pembuatan model dan analisa eksperimen, dan akhirnya bagian kesimpulan dan saran.

Bagian pendahuluan memberikan penjelasan mengenai latar belakang pemilihan topik penelitian. Kemudian diberikanlah diagram mengenai berbagai masalah yang saling berkaitan dengan topik penelitian. Setelah itu, ditetapkanlah tujuan penelitian ini dan juga ruang lingkup penelitian ini. Bagian ini juga menjelaskan metodologi penelitian dan sistematika penulisan.

Bagian dasar teori memberikan landasan ilmiah untuk proses pengerjaan penelitian ini. Landasan ilmiah yang mendukung proses penelitian ini adalah teori antrian, diagram IDEF0, statistik, *time study*, dan pemodelan sistem. Landasan ilmiah ini diambil dari berbagai sumber seperti buku, jurnal, artikel, dan tulisan dari internet.

Bagian pengumpulan dan pengolahan data memberikan pembahasan mengenai pengumpulan data yaitu jenis-jenis data yang dibutuhkan beserta sumber-sumber yang akan memberikan data-data yang dibutuhkan, cara-cara pengumpulan data-data tersebut, dan akhirnya proses pengolahan data-data tersebut diantaranya seperti proses penentuan distribusi untuk data-data yang telah dikumpulkan.

Bagian pembuatan model dan analisa eksperimen akan menggunakan *software ProModel* dalam formulasi, verifikasi, dan validasi model proses yang dijadikan objek penelitian. Hasil simulasi dalam lima skenario kemudian akan dianalisa untuk mendapatkan hasil simulasi yang sesuai dengan tujuan penelitian.

Bagian kesimpulan dan saran mengungkapkan hasil yang diperoleh dari analisa skenario terhadap simulasi eksperimen yang telah dilakukan. Berbagai saran akan diberikan berkaitan dengan usulan yang didapat dari kesimpulan sesuai tujuan penelitian.



2. DASAR TEORI

2.1. Pemodelan Simulasi Sistem

2.1.1. Simulasi dengan *Software*

Simulasi adalah imitasi atau tiruan dari suatu sistem dinamis dengan menggunakan suatu model komputer dengan tujuan untuk melakukan evaluasi dan peningkatan pada performa sistem tersebut¹. Kegunaan simulasi sangat banyak, salah satunya ialah simulasi membantu meminimalkan resiko biaya dan kadangkala meminimalkan kesalahan fatal dalam dunia nyata. Teknologi simulasi mengalami peningkatan dalam jumlah aplikasi produknya yang bervariasi mulai dari pelatihan untuk pilot pesawat terbang hingga pengujian prototipe produk baru. Satu hal yang dimiliki oleh semua aplikasi produk simulasi pada umumnya yaitu menyediakan suatu lingkungan virtual yang membantu menyiapkan (tiruan) situasi kehidupan-nyata sehingga menghasilkan penghematan waktu, biaya, dan bahkan nyawa. Satu daerah dimana simulasi mengalami peningkatan aplikasi ialah dalam desain dan peningkatan kinerja sistem pelayanan dan manufaktur. Kemampuan simulasi yang unik untuk memprediksi kinerja sistem yang kompleks secara akurat, menjadikan simulasi cocok untuk perencanaan sistem. Model-model simulasi umumnya didefinisikan dengan menggunakan *software* simulasi komersial yang menyediakan konstruksi pemodelan yang nyaman dan peralatan analisa. Data pada model diubah menjadi data simulasi, yang diproses selama simulasi berlangsung. Di akhir proses simulasi, statistik disimpulkan dalam database hasil yang ditabulasikan atau digrafikkan dalam berbagai bentuk.

Fishwick² menyimpulkan keuntungan menggunakan model-model dan metodologi-metodologi simulasi untuk memodelkan sistem pelayanan dan manufaktur sebagai berikut:

- Model-model simulasi lebih mudah disimpulkan daripada model-model analitik

¹ Charles Harrel, Biman K.Gosh, dan Royce Bowden. *Simulation Using Promodel*. New York : McGraw-Hill Higher Education. 2000: h 5.

² P.A. Fishwick. *Computer simulation: growth through extension*, Transactions of the Society for Computer Simulation International 14. 1997: h 13–23.

- Model-model yang realistis (mewakili sistem yang kompleks) dapat dikonstruksi, sedangkan model-model analitik membutuhkan asumsi-asumsi yang disederhanakan
- Pengukuran performa untuk menilai perilaku sistem adalah mudah didapat, dan hasil visual memberikan suatu bantuan yang baik untuk pengguna-akhir
- Eksperimen alternatif dapat dengan cepat dilakukan secara bebas terhadap sistem nyata

Keuntungan-keuntungan tersebut menyimpulkan³ bahwa simulasi digunakan dalam sistem manufaktur dan pelayanan yang fleksibel⁴ untuk meningkatkan kontrolnya.

Aplikasi produk simulasi telah berkembang untuk menyediakan lebih dari sekedar kemampuan simulasi yang berdiri-sendiri. Aplikasi produk simulasi modern telah membuka arsitektur berbasis teknologi komponen dan metode-metode akses data (seperti SQL) untuk menyediakan kapabilitas *interface* dengan aplikasi produk lainnya seperti program-program CAD dan peralatan perencanaan perusahaan yang lainnya. Survei yang dilaporkan tiap tahun dalam *Industrial Engineering Solutions* menunjukkan bahwa sebagian besar aplikasi produk simulasi memiliki fitur-fitur sebagai berikut⁵:

- Analisa data *input* dengan *distribution fitting*
- Tampilan grafis dengan *point-and-click*
- *Templates* dan komponen yang dapat digunakan berulang kali
- Animasi dua-dimensi dan tiga-dimensi
- Pengajaran dan bantuan secara *online*
- Proses *debug* yang interaktif
- Generasi model yang otomatis
- Peralatan analisa hasil
- Optimasi

³ P. Pujo. *Formal DEVS modelling and simulation of a flow-shop relocation method without interrupting the production*. Simulation Modelling Practice and Theory 14. 2006: h 817–842.

⁴ J. Kosturiak dan M. Gregor. *FMS simulation: some experience and recommendations*. Simulation Practice and Theory 6. 1998: h 423–442.

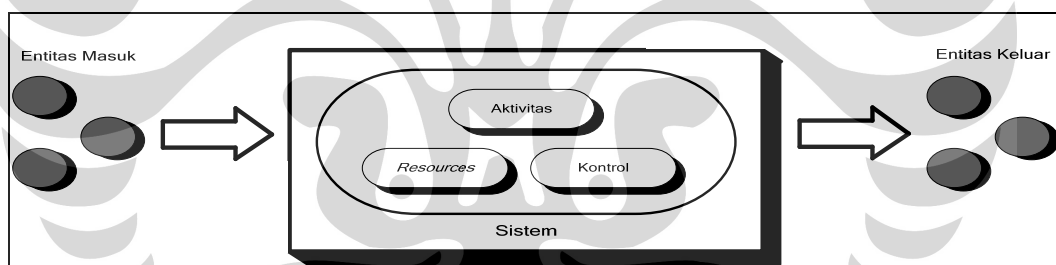
⁵ Charles Harrel, Biman K.Gosh, dan Royce Bowden. *Op. Cit.* h 72.

- Konektivitas arsitektur dan database yang terbuka

Simulasi adalah suatu teknologi yang akan berlanjut berkembang seiring peningkatan teknologi dan lebih banyak waktu digunakan untuk pengembangan *software*. Aplikasi produk simulasi akan menjadi lebih mudah digunakan dengan kecerdasan yang dimasukkan ke dalam *software* itu sendiri. Bukti dari tren ini dapat dilihat dengan optimasi dan fungsi penghemat-waktu lainnya yang muncul pada aplikasi produk simulasi. Animasi dan teknik visualisasi grafis lainnya akan terus memainkan peran penting dalam simulasi. Seperti 3-D (tiga dimensi) dan pengembangan teknologi grafis lainnya, fitur-fitur ini juga akan dimasukkan dalam produk simulasi.

2.1.2. Dinamika Sistem

Sistem merupakan kumpulan dari elemen-elemen yang saling berhubungan dan bekerja sama untuk mencapai tujuan yang spesifik⁶. Berdasarkan sudut pandang simulasi, sistem terdiri atas entitas, aktivitas, sumberdaya, dan kontrol⁷.



Gambar 2.1 Komponen dalam suatu sistem

(Sumber: Charles Harrel, Biman K.Gosh dan Royce Bowden. 2000: h 49)

Entitas adalah objek yang diproses melalui sistem seperti dokumen dan pelanggan. Entitas dapat dikategorikan menjadi tiga yaitu :

- a. Makhluk hidup : pelanggan, pasien
- b. Benda mati : *part*, dokumen, dan bisnis
- c. Benda yang tak berwujud : panggilan, *email*

⁶ Charles Harrel, Biman K.Gosh dan Royce Bowden. *Ibid.* h 25.

⁷ Charles Harrel, Biman K.Gosh dan Royce Bowden. *Ibid.* h 25.

Aktivitas merupakan pekerjaan untuk memproses entitas di dalam sistem. Pekerjaan ini dapat berhubungan secara langsung ataupun tidak dalam pemrosesan entitas. Aktivitas dapat dikategorikan :

- a. Proses entitas : masuk dan inspeksi
- b. Entitas dan pergerakan sumberdaya : pengoperasian *forklift*, penggunaan *elevator*
- c. Pengaturan sumber daya, pemeliharaan, dan perbaikan : *set-up* mesin, perbaikan mesin

Sumberdaya adalah agen yang digunakan untuk memproses entitas dalam sistem. Dapat dikategorikan menjadi tiga, yaitu :

- a. Mahluk hidup : pekerja, dokter
- b. Benda mati : peralatan
- c. Benda yang tidak berwujud : informasi

Yang terakhir adalah kontrol, yaitu hal yang memperlihatkan bagaimana, kapan dan dimana aktivitas dijalankan. Pada tingkat tertinggi, kontrol terdiri atas perencanaan, penjadwalan, dan kebijakan. Sedangkan pada tingkat terendah terdiri atas prosedur tertulis. Dan di semua tingkat kontrol terdiri atas informasi dan logika keputusan untuk mengetahui bagaimana seharusnya sistem beroperasi.

Kompleksitas dari suatu sistem merupakan gabungan dari dua faktor, yaitu elemen yang tidak tetap dan elemen yang saling bergantung. Saling ketergantungan antara elemen dari sistem menyebabkan setiap elemen dapat mempengaruhi elemen lainnya. Sedangkan elemen yang tidak tetap akan menghasilkan ketidakpastian dalam suatu sistem. Elemen-elemen tersebut dapat disebut juga dengan variabel-variabel. Ada tiga macam variabel pada sistem yaitu:

- a. Variabel Keputusan

Merupakan variabel yang tidak bergantung pada variabel lain pada suatu sistem. Mengubah nilai dari variabel ini akan mempengaruhi karakteristik dari suatu sistem.

- b. Variabel Responsif

Merupakan variabel yang memperlihatkan hasil dari dimasukkannya variabel keputusan. Jadi di dalam sistem, nilai variabel ini bergantung pada variabel keputusan.

c. Variabel Status

Merupakan variabel yang mengindikasikan status dari suatu sistem pada waktu yang tertentu.

2.1.3. Pengumpulan Data Simulasi

Pengumpulan data merupakan tahapan yang paling menantang dan menghabiskan banyak waktu dalam simulasi. Untuk sistem yang baru (belum pernah diterapkan), informasi biasanya sangat kasar dan hanya diperkirakan. Untuk sistem yang telah diterapkan, mungkin terdapat data mentah yang tak tersusun selama bertahun-tahun. Informasi biasanya jarang tersedia dalam bentuk yang langsung dapat dipakai dalam pembuatan suatu model simulasi. Oleh sebab itu, informasi tersebut harus dikumpulkan dan diseleksi secara tepat. Ada enam cara mengumpulkan dan menyeleksi data secara tepat, yaitu:

1. Identifikasikan data yang menimbulkan suatu kejadian
2. Fokuskan data hanya pada faktor yang berpengaruh
3. Fokuskan pada waktu aktivitas yang sebenarnya
4. Kelompokkan data dalam kelompok tertentu
5. Fokuskan pada esensi (hubungan sebab-akibat) daripada substansi
6. Pisahkan variabel keputusan (*input*) dari variabel responsif

Data pada sistem dari sudut pandang simulasi, dapat dibagi menjadi tiga kategori yaitu

1. Data Struktural

Data struktural meliputi semua objek dalam sistem yang akan dimodelkan. Objek tersebut meliputi entitas (produk, pelanggan, dst.), sumberdaya (operator, mesin, dst.), dan lokasi (ruang tunggu, stasiun kerja, dst.)

2. Data Operasional

Data operasional menjelaskan bagaimana sistem beroperasi, yaitu: kapan, dimana, dan bagaimana kejadian dan aktivitas berlangsung. Data

operasional terdiri dari semua informasi perilaku atau logis tentang sistem seperti rute trayek, penjadwalan, perilaku *downtime*, dan alokasi sumberdaya.

3. Data Numerikal

Data numerikal menyediakan informasi kuantitatif tentang suatu sistem. Contoh dari data numerikal meliputi kapasitas, tingkat kedatangan, dan waktu antara kegagalan.

Sumber pengumpulan data bervariasi tergantung keadaan sistem. Untuk sistem yang telah diterapkan, terdapat banyak sumber pengumpulan data seperti catatan dan pengetahuan orang tentang sistem tersebut. Untuk sistem yang baru, sumber pengumpulan data biasanya terbatas pada orang yang langsung terlibat dalam desain sistem tersebut. Sumber pengumpulan data yang baik meliputi:

- Catatan historis—produksi, penjualan, dst.
- Dokumentasi sistem—rancangan proses, *layout* fasilitas, dst.
- Observasi—*time study*, dst.
- Wawancara—operator (metode kerja), staf pemeliharaan (prosedur perbaikan), insiyur (rute trayek), manajer (penjadwalan dan peramalan)
- Perbandingan dengan sistem sejenis—dalam perusahaan, dalam industri yang sama, dalam industri yang lain
- Klaim supplier—waktu proses, reliabilitas dari mesin baru
- Estimasi desain—waktu proses, waktu pindah, dsb untuk sistem baru
- Literatur penelitian—penelitian *time study*, dst

Model simulasi dapat dijalankan dengan data yang salah tetapi tidak dapat dijalankan dengan data yang tidak lengkap. Asumsi dimunculkan karena ketidaklengkapan data (terlebih lagi untuk sistem yang baru). Alasan melakukan simulasi ialah untuk memprediksi performa sistem berdasarkan asumsi yang diberikan. Lebih baik membuat keputusan dengan mengetahui implikasi dari asumsi kita daripada menggabungkan kesalahan keputusan dengan memasukkan kesalahan dari asumsi tersebut. Memasukkan asumsi ke dalam suatu model dapat membantu validasi asumsi kita dengan melihat apakah asumsi tersebut memiliki

dampak pada keseluruhan operasi pada model. Salah satu cara menilai pengaruh asumsi terhadap validitas suatu model ialah dengan analisa kepekaan.

2.1.4. Pembuatan Model Simulasi suatu Sistem

Willemain⁸ mendata lima kualitas dari suatu model yang efektif: validitas, nyaman digunakan, bernilai untuk klien, dapat / mungkin dijalankan, dan sesuai dengan permasalahan klien. Sementara itu, Brooks dan Tobias⁹ mengidentifikasi sebelas kriteria performa untuk suatu model yang baik. Akan tetapi, ada empat syarat utama¹⁰ untuk suatu model yang baik:

- Validitas: model cukup akurat sesuai tujuannya
- Kredibilitas: model dan hasilnya dapat dipercaya
- Utilitas: model berguna sesuai spesifikasi aplikasinya
- Feasibilitas: punya kemungkinan untuk mengembangkan dan menggunakan model dengan waktu dan sumberdaya yang tersedia

Persyaratan kelima mungkin ditambahkan, yaitu akredibilitas (model sesuai dengan akreditasi), untuk model yang menjadi subjek dari penelitian independen.

Karena permintaan terhadap kecepatan akhir-akhir ini, waktu untuk menyelesaikan suatu studi simulasi dianggap lama, bahkan dengan perkembangan *hardware* dan *software* simulasi. Oleh sebab itu, metode penyederhanaan model simulasi¹¹ memainkan peran utama. Tujuan utama penyederhanaan model ialah untuk menemukan suatu model konsep yang lebih sederhana yang masih valid berdasarkan persyaratan simulasi. Persyaratan simulasi merupakan tujuan dari studi simulasi, yang secara utama disebut sebagai pengukuran performa yang diinginkan. Ada dua pendekatan untuk mendapatkan suatu model yang lebih sederhana: konstruktif atau evolutif. Untuk

⁸ T. R. Willemain. *Insights on modeling from a dozen experts*. Operations Research 42 (2): 1994: h 213-22.

⁹ R. J. Brooks dan A. M. Tobias. *Choosing the best model: Level of detail, complexity and model performance*. Mathematical and Computer Modelling 24 (4): 1996: h 1-14.

¹⁰ S. Robinson. *Distributed Simulation and Simulation Practice*. SIMULATION, Vol. 81, Issue 1, January 2005 h 5-13 (2005) The Society for Modeling and Simulation International.

¹¹ L. Chwif. *Discrete event simulation model reduction: A causal approach*. Simulation Modelling Practice and Theory 14: 2006: h 930-944.

pendekatan “konstruktif”, suatu model yang lebih sederhana akan muncul langsung dari potongan model yang lebih besar. Untuk pendekatan “evolutif”, suatu model yang lebih sederhana muncul dari model awal. Untuk pendekatan “evolutif”, masih terdapat dua kemungkinan:

- 1) Model awal lebih kompleks sehingga kita mencoba untuk menyederhanakannya. Ini disebut pendekatan evolutif-reduksi.
- 2) Model awal terlalu sederhana sehingga kita menambahkan detail yang diperlukan. Ini disebut pendekatan evolutif-ekspansi.

2.1.5. Verifikasi dan Validasi Model

Verifikasi merupakan proses untuk menentukan apakah model hasil simulasi telah beroperasi sesuai yang diinginkan. Beberapa cara yang dapat dilakukan dalam verifikasi adalah:

- a. Melakukan pengecekan pada kode model
- b. Memeriksa masuk akal nya output
- c. Mengamati tingkah laku sistem dalam animasi
- d. Menggunakan fungsi *trace* dan *debug* pada *software* untuk mendapatkan keadaan model yang tidak dapat terlihat melalui animasi

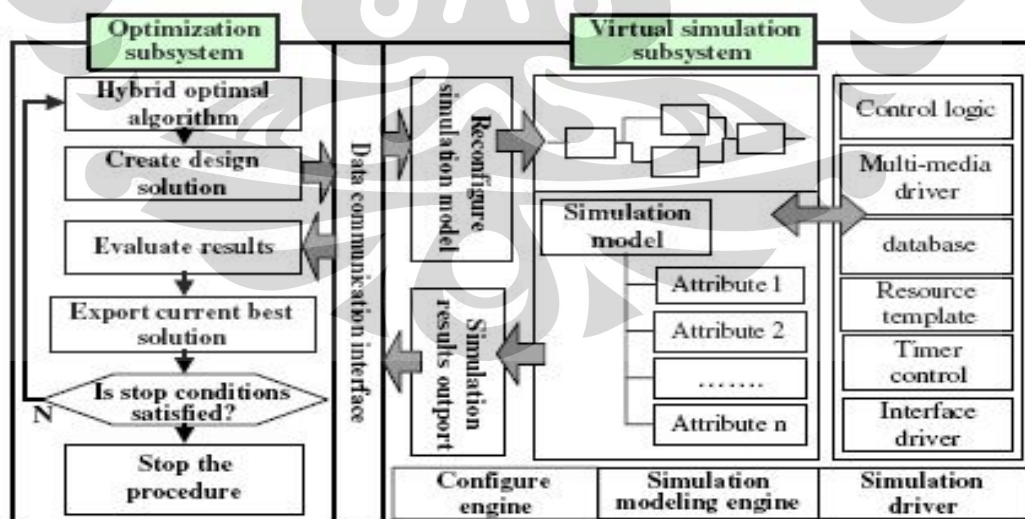
Validasi merupakan proses untuk menentukan apakah model telah mencerminkan keadaan nyata secara akurat (Hoover dan Perry 1990). Beberapa cara yang dilakukan dalam validasi model adalah:

- a. Melakukan pengamatan pada animasi untuk dibandingkan dengan pengetahuan seseorang mengenai keadaan sebenarnya.
- b. Membandingkan dengan sistem aktual
- c. Membandingkan dengan model lain yang telah tervalidasi
- d. Melakukan uji degenerasi untuk melihat apakah sifat output dari model akan berubah ketika salah satu variabelnya mencapai titik tertentu
- e. Menggunakan uji kondisi ekstrim seperti dengan menghalau kedatangan pada suatu sistem untuk melihat apakah sistem masih akan berjalan
- f. Melakukan pemeriksaan dengan *face validity*, yaitu dengan menanyakan seseorang yang memiliki pengetahuan mengenai sistem mengenai masuk akal nya model yang dibuat

- g. Melakukan pengujian terhadap data historis
- h. Melakukan analisis sensitivitas
- i. Melakukan penelitian terhadap kejadian-kejadian yang terjadi pada model untuk dibandingkan dengan tingkah laku sistem aktual
- j. Melakukan *turing test* yaitu sebuah uji untuk melihat apakah orang-orang yang berpengalaman mengenai sistem aktual dapat membedakan output dari sistem dan output dari model.

2.1.6. Struktur Sistem Optimasi dengan Simulasi

Berdasarkan suatu prinsip yang memisahkan model simulasi dengan algoritma optimasi, dan menjadikan sistem manufaktur sebagai bahan investigasi, tulisan¹² ini menjelaskan suatu struktur sistem optimasi dengan simulasi berbasis lingkungan pemodelan simulasi virtual, yang terdiri atas dua sub-sistem yang relatif tidak bersentuhan: sub-sistem optimasi dan sub-sistem simulasi virtual, keduanya tidak dapat hanya menyelesaikan optimasi dan fungsi simulasi secara terpisah satu sama lain, tetapi keduanya harus bekerjasama dalam suatu proses optimasi dalam model simulasi. Suatu *interface* data gabungan yang dapat memindahkan informasi data, informasi kontrol, dan informasi sinyal digunakan sebagai penghubung antara kedua sub-sistem tersebut (lihat gambar).



Gambar 2.2 Struktur Sistem Optimasi dengan Simulasi

(Sumber: Y. Guo. 2006: h 578)

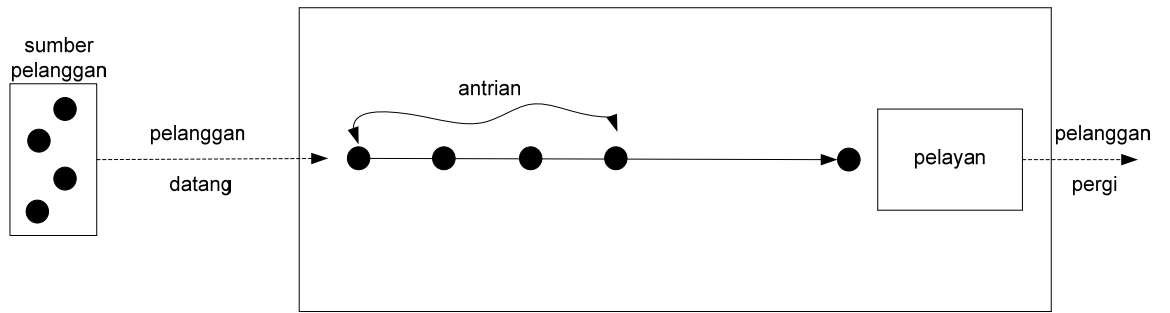
¹² Y. Guo. *SimOpt: A new simulation optimization system based virtual simulation for manufacturing system*. Simulation Modelling Practice and Theory 14: 2006: h 577-585.

Ketika suatu sistem manufaktur dinamik perlu dioptimasi, pertama kali, model simulasi dari sistem tersebut harus dibuat dalam sub-sistem simulasi virtual. Dalam model tersebut, semua variabel desain dan objek optimasi merupakan atribut dari berbagai objek atau fungsi dari atribut tersebut, seperti kecepatan *conveyor*, utilisasi mesin, dan seterusnya, pengguna *software* simulasi dapat menggunakan *configure engine* untuk memetakan variabel dan objek tersebut ke dalam suatu *parameter manager* dari sub-sistem optimasi untuk membuat model (atau fungsi) optimal. *Simulation driver* dapat menjalankan model simulasi dan hasil simulasi *output*; secara bersamaan, suatu model optimal primer akan otomatis dibuat oleh *parameter manager*. Pengguna *software* simulasi mungkin menambahkan hubungan kendala dan menetapkan batasan pada variabel desain tersebut dalam sub-sistem optimasi. Ketika sub-sistem optimasi berjalan, algoritma optimasi akan membuat satu paket variabel desain awal yang mungkin, dan memindahkannya ke sub-sistem simulasi virtual melalui data *interface*; sub-sistem simulasi virtual menerapkan variabel tersebut untuk menyusun ulang model simulasi saat itu juga, dan menjalankan model simulasi, kemudian hasil simulasi akan dikembalikan otomatis pada sub-sistem optimasi; berdasarkan hasil, algoritma optimasi akan menyesuaikan arah pencarian nilai optimal dan membuat satu paket variabel solusi baru yang mungkin. Proses tersebut terus terulang hingga kondisi yang diinginkan terjadi. Akhirnya, variabel desain optimal dan nilai terbaik dari objek dikeluarkan dan direkam.

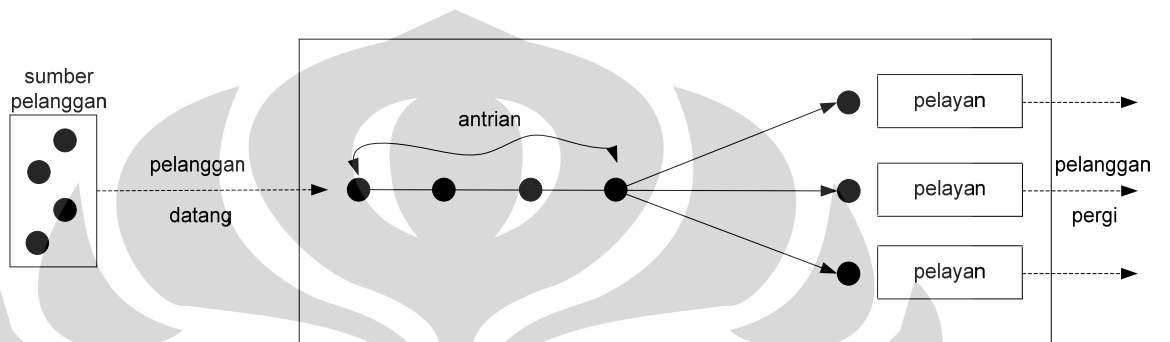
2.2. Sistem Antrian

Suatu proses antrian adalah suatu proses yang berhubungan dengan kedatangan seorang pelanggan pada suatu fasilitas pelayanan, kemudian menunggu dalam suatu baris (antrian) jika semua pelayannya sibuk, dan akhirnya meninggalkan fasilitas tersebut. Sebuah sistem antrian adalah suatu himpunan pelanggan, pelayan, dan suatu aturan yang mengatur kedatangan pada pelanggan dan pemrosesan masalahnya¹³. Skema beberapa sistem antrian (lihat gambar 2.3).

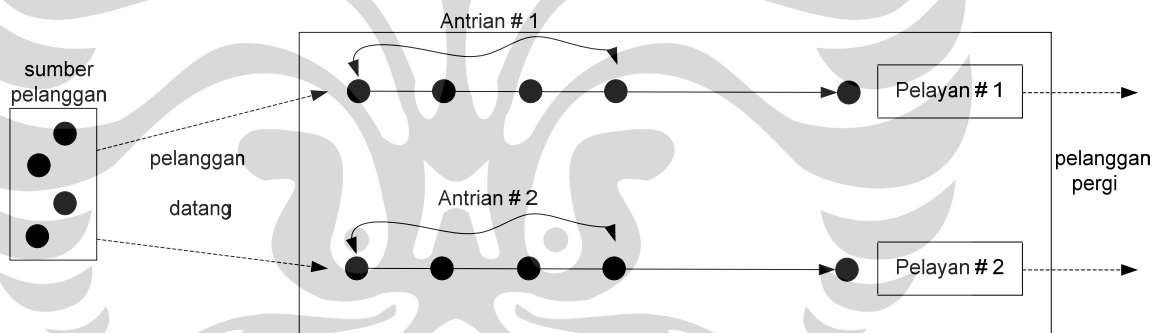
¹³ Richard Bronson. *Theory and Problems of Operation Research*. McGraw-Hill, Inc. 1993: h 308.



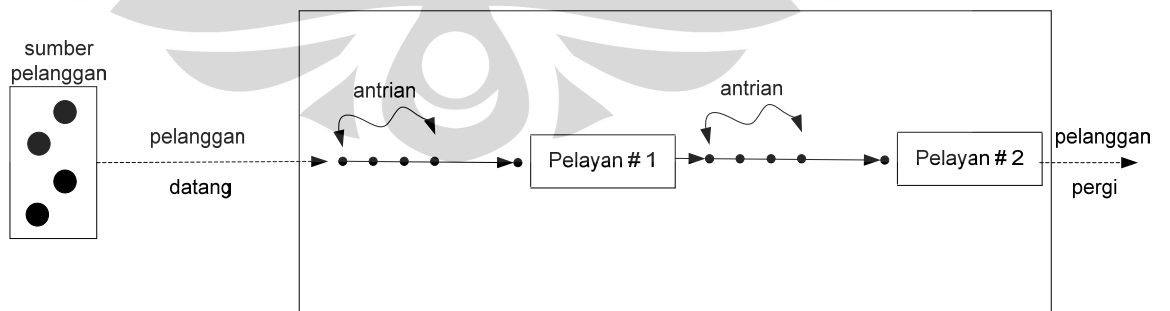
(a) Sistem Antrian: Satu antrian, satu pelayan



(b) Sistem Antrian: Satu antrian, pelayan paralel



(c) Sistem Antrian: Beberapa antrian, beberapa pelayan paralel



(d) Sistem Antrian: Satu antrian, beberapa pelayan

Gambar 2.3 Tipe Sistem Antrian

(Sumber: Richard Bronson. McGraw-Hill, Inc. 1993: h 309)

Sistem-sistem antrian dicirikan oleh lima buah komponen: pola kedatangan para pelanggan, pola pelayanan, jumlah pelayan, kapasitas fasilitas untuk menampung para pelanggan dan aturan dalam mana para pelanggan dilayani. Pola kedatangan para pelanggan biasanya dicirikan oleh *waktu antar-kedatangan*, yakni waktu antara kedatangan dua pelanggan yang berurutan pada suatu fasilitas pelayanan. Pola ini dapat deterministik (yakni, diketahui secara pasti), atau berupa suatu variabel acak yang distribusi probabilitasnya dianggap telah diketahui. Pola ini dapat bergantung pada jumlah pelanggan yang berada dalam sistem, atau tidak-bergantung pada keadaan sistem antrian ini.

Juga yang menarik adalah apakah para pelanggan datang satu per satu atau secara berombongan dan apakah penolakan (*balking*) atau pembatalan (*reneging*) diperkenankan. *Penolakan* terjadi apabila seorang pelanggan menolak untuk memasuki suatu fasilitas pelayanan karena antriannya terlalu panjang. *Pembatalan* terjadi apabila seorang pelanggan yang telah berada dalam suatu antrian meninggalkan antrian dan fasilitas pelayanan yang dituju karena ia menunggu terlalu lama. Bila tidak disebutkan secara khusus, maka anggapan standarnya adalah bahwa semua pelanggan tiba satu per satu dan juga bahwa tidak terjadi penolakan dan pembatalan.

Pola pelayanan biasanya dicirikan oleh *waktu pelayanan (service time)*, yaitu waktu yang dibutuhkan seorang pelayan untuk melayani seorang pelanggan. Waktu pelayanan ini dapat bersifat deterministik, atau berupa suatu variabel acak yang distribusi probabilitasnya dianggap telah diketahui. Besaran ini dapat bergantung pada jumlah pelanggan yang telah berada di dalam fasilitas pelayanan, atau tidak bergantung pada keadaannya. Juga yang menarik adalah apakah seorang pelanggan hanya dilayani oleh satu pelayan atau, seperti dalam gambar 2.3(d), pelanggan ini membutuhkan suatu barisan pelayan. Bila tidak disebutkan secara khusus, maka anggapan dasarnya adalah bahwa satu pelayan saja dapat melayani secara tuntas urusan seorang pelanggan.

Kapasitas sistem adalah jumlah maksimum pelanggan, mencakup yang sedang dilayani dan yang berada dalam antrian, yang dapat ditampung oleh fasilitas pelayanan pada saat yang sama. Apabila seorang pelanggan datang pada suatu

Universitas Indonesia

tempat fasilitas yang telah penuh, maka pelanggan ini ditolak untuk memasukinya. Ia juga tidak diperkenankan untuk menunggu di luar (karena akan memperbesar kapasitas) dan dipaksa untuk meninggalkan tempat pelayanan tersebut tanpa mendapatkan pelayanan. Sebuah sistem yang tidak membatasi jumlah pelanggan di dalam fasilitas pelayanannya memiliki *kapasitas tak terhingga*, sedangkan suatu sistem yang membatasi jumlah pelanggan memiliki *kapasitas berhingga*.

Disiplin antrian adalah aturan dalam mana para pelanggan dilayani. Aturan ini dapat didasarkan pada yang pertama-masuk, pertama-keluar (PMPK, dalam bahasa Inggris disingkat FIFO) (yakni, pelayanan menurut urutan kedatangan), yang terakhir-masuk pertama-keluar (TMPK, dalam bahasa Inggris disingkat LIFO) (contohnya, pelanggan yang datang paling akhir mendapat pelayanan yang berikutnya), secara acak, atau berdasarkan prioritas.

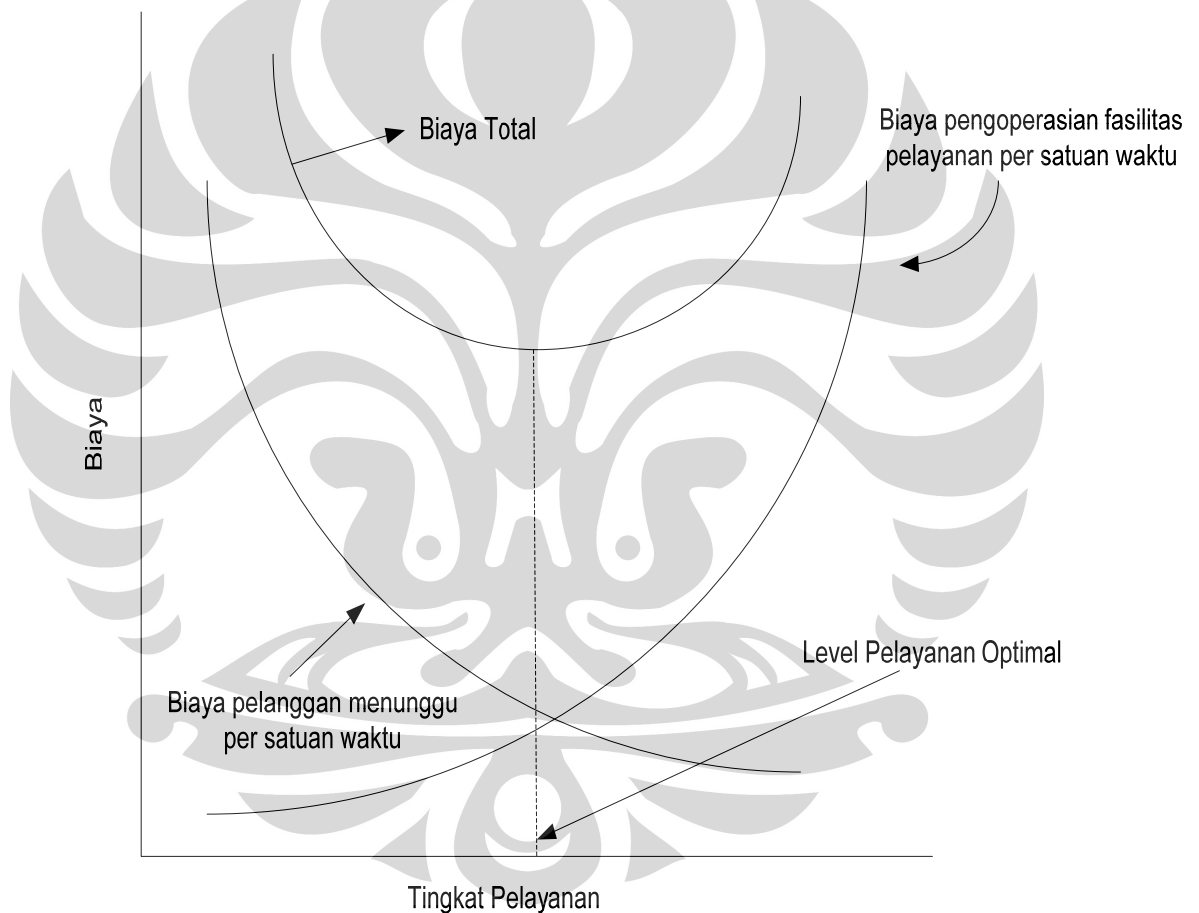
Notasi Kendall untuk merinci ciri dari suatu antrian adalah $v/w/x/y/z$, dimana v menunjukkan pola kedatangan, w menunjukkan pola pelayanan, x menyatakan jumlah pelayan yang ada, y menunjukkan kapasitas sistem dan z menunjukkan disiplin antrian. Dalam Tabel 2.1 diperkenalkan berbagai notasi untuk waktu antar-kedatangan atau waktu pelayanan dan disiplin antrian. Jika y dan z tidak ditentukan, maka dimaksudkan bahwa y adalah ∞ dan z adalah PMPK/FIFO.

Tabel 2.1 Notasi Waktu Antar-Kedatangan atau Waktu Pelayanan dan Disiplin Antrian

Ciri Antrian	Simbol	Arti
Waktu antar-kedatangan atau Waktu pelayanan	D	Deterministik
	M	Terdistribusi secara eksponensial
	E_k	Distribusi Erlang, tipe k ($k = 1, 2, \dots$)
	G	Distribusi yang lain
Disiplin Antrian	FIFO (PMPK)	Pertama Masuk, pertama keluar
	LIFO (TMPK)	Terakhir masuk, terakhir keluar
	SIRO	Pelayanan dalam urutan acak
	PRI	Urutan Prioritas
	GD	Urutan khusus yang lain

(Sumber: Richard Bronson. McGraw-Hill, Inc. 1993: h 311)

Hasil dari analisa antrian dapat digunakan dalam konteks model optimasi biaya¹⁴, dimana jumlah biaya pelayanan yang ditawarkan dan biaya menunggu diminimalkan. Gambar 2.4 menampilkan suatu model biaya umum (dalam dollar per satuan waktu) dimana biaya pelayanan meningkat seiring peningkatan tingkat pelayanan. Pada saat yang bersamaan, biaya menunggu berkurang seiring dengan peningkatan tingkat pelayanan. Hambatan utama dalam pelaksanaan model biaya ialah mungkin sulit untuk mendapatkan perkiraan satuan biaya menunggu yang dapat dipercaya, khususnya saat perilaku manusia berdampak pada operasi dari situasi.



Gambar 2.4 Model Keputusan Antrian Berbasis-Biaya
(Sumber: Hamdy A.Taha. Pearson Education, Inc. 2003: h 580)

¹⁴ Hamdy A.Taha. *Operation Research: An Introduction*. Pearson Education, Inc. 2003

2.3. Diagram IDEFØ

Diagram IDEF (*Integrated Definition*) diperkenalkan pada tahun 1981 sebagai bagian dari proyek ICAM (*Integrated Computer-Aided Manufacturing*) milik Angkatan Udara Amerika Serikat. IDEFØ diturunkan dari bahasa grafis yaitu SADT (*Structured Analysis and Design Technique*). IDEFØ ialah suatu metode yang didesain untuk memodelkan keputusan-keputusan, tindakan-tindakan, dan aktivitas-aktivitas dari suatu organisasi ataupun sistem dan ditujukan untuk mengomunikasikan dan menganalisa perspektif fungsional dari suatu sistem¹⁵. IDEFØ berguna dalam pembuatan ruang lingkup dari suatu analisa, khususnya untuk suatu analisa fungsional. Sebagai suatu alat komunikasi, IDEFØ meningkatkan keterlibatan ahli dalam bidang yang berkaitan dan konsensus pengambilan-keputusan melalui alat grafis yang disederhanakan. Sebagai suatu alat analisa, IDEFØ membantu pemodel dalam mengidentifikasi fungsi apa yang diperlihatkan, apa yang diperlukan untuk memperlihatkan fungsi-fungsi tersebut, apa yang sistem sekarang lakukan dengan benar, dan apa yang sistem sekarang lakukan dengan salah. Hasilnya, model IDEFØ sering dibuat sebagai satu dari tahapan awal dalam usaha pengembangan suatu sistem¹⁶.

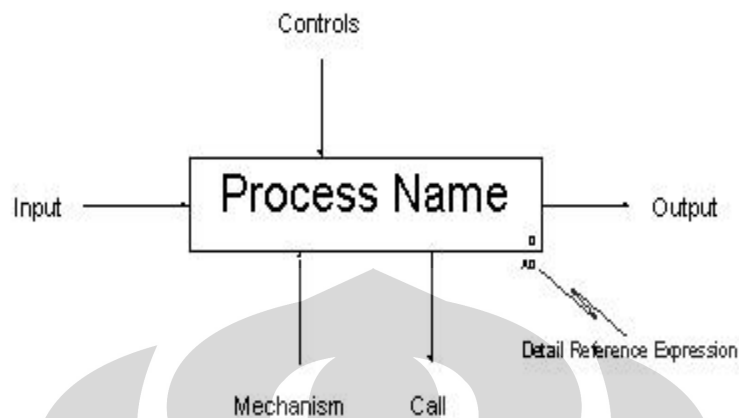
Sebagai suatu bahasa pemodelan fungsional, IDEFØ memiliki karakteristik sebagai berikut:

1. Komprehensif dan ekspresif; mampu mempresentasikan secara grafis yaitu variasi luas dari bisnis, manufaktur, dan operasi lainnya pada berbagai tingkat detail
2. Bahasa yang koheren dan sederhana; menyediakan ekspresi yang tepat dan singkat, dan mempromosikan konsistensi kegunaan dan interpretasi
3. Meningkatkan komunikasi antara analis sistem, pengembang, dan pemakai melalui kemudahan pembelajaran dan kejelasan penggambaran detail yang bertingkat

¹⁵ "Integration Definition for Function Modeling (IDEF0)". dalam *Draft Federal Information Processing Standards Publication 183*. National Technical Information Service, U.S. Department of Commerce. Springfield. 1994. h 18.

¹⁶ www.idef.com

Diagram IDEFØ secara sederhana terdiri atas kumpulan kotak IDEFØ digambarkan sebagai berikut dan terdiri dari komponen-komponen¹⁷:



Gambar 2.5 Kotak IDEFØ secara sederhana

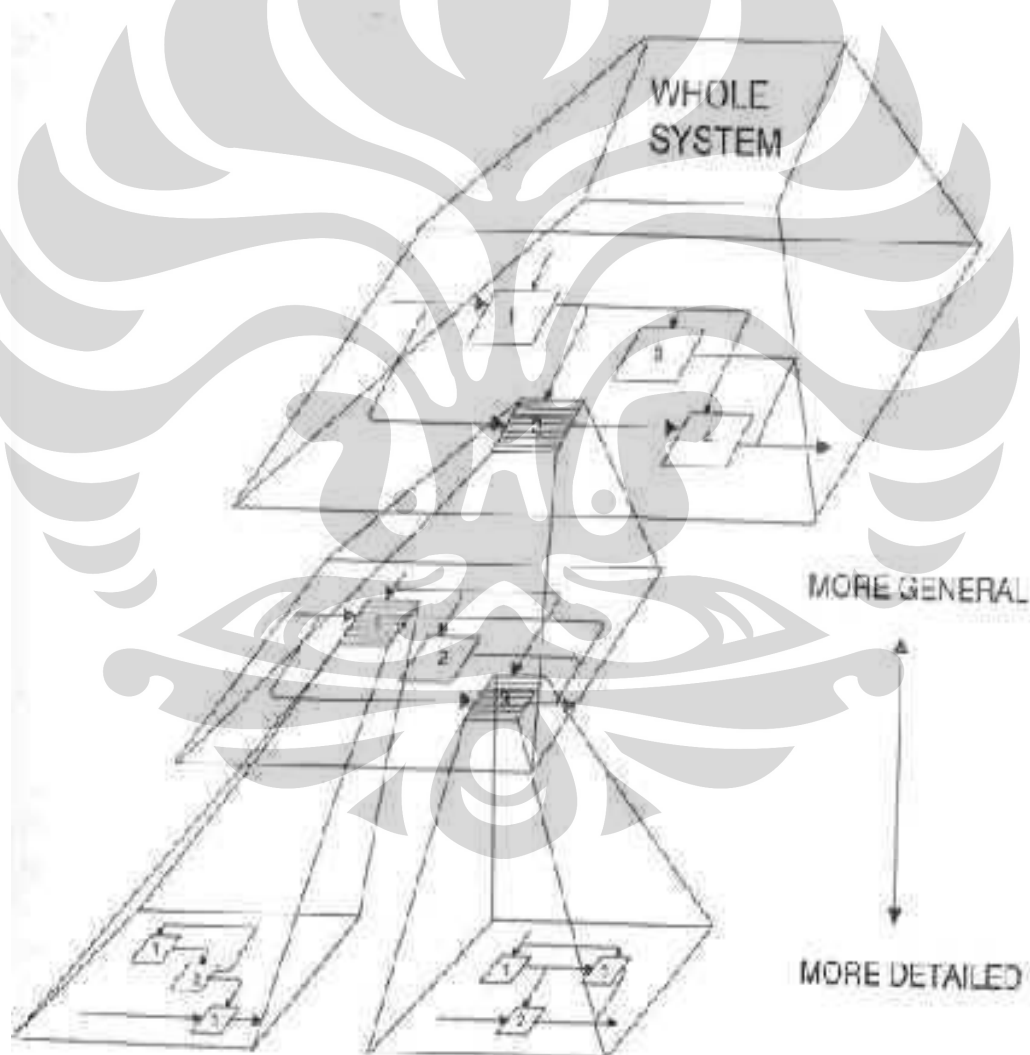
(sumber: Clarence G.Feldmann. Dorset House Publishing Co., Inc. 1998: h 63)

- Nama Proses (fungsi) – nama suatu proses (fungsi) yang akan mengubah input menjadi output dengan control dan mechanism tertentu. Nama proses (fungsi) menggunakan kata kerja
- *Inputs* – data atau objek yang diubah oleh proses (fungsi) menjadi *output*. Panah untuk Input digambarkan di sebelah kiri kotak IDEFØ. Nama *input* menggunakan kata benda
- *Controls* – kondisi tertentu yang dibutuhkan untuk menghasilkan output yang benar. Panah untuk *Control* digambarkan di sebelah atas kotak IDEFØ. Nama control menggunakan kata benda
- *Outputs* – data atau objek yang dihasilkan oleh proses (fungsi). Panah untuk *Output* digambarkan di sebelah kanan kotak IDEFØ. Nama output menggunakan kata benda
- *Mechanisms* – sarana yang dibutuhkan untuk menjalankan proses (fungsi). Panah untuk *Output* digambarkan di sebelah bawah kotak IDEFØ. Nama *mechanism* menggunakan kata benda

¹⁷Clarence G. Feldmann. *The Practical Guide to Business Process Reengineering Using IDEFØ*, Dorset House Publishing Co., Inc. 1998. h 63.

- *Calls* – bagian dari *Mechanisms* yang memungkinkan proses yang lebih detail pada kotak lainnya sebagai informasi tambahan. Nama *call* menggunakan kata benda

Dalam gambar diagram IDEFØ yang menjelaskan suatu model sistem yang kompleks, maka akan ditemukan kotak induk dengan kotak anak sebagaimana ditemukannya diagram induk dengan diagram anak. Istilah “induk” merujuk pada sistem utama yang lebih besar; sedangkan istilah “anak” merujuk kepada komponen sistem yang lebih kecil. Hubungan “induk-anak” ini digambarkan dengan gambar sebagai berikut:



Gambar 2.6 Diagram IDEFØ dari suatu model yang kompleks
(sumber: Clarence G.Feldmann. Dorset House Publishing Co., Inc. 1998: h 63)

2.4. Statistik

Statistik adalah pengetahuan yang berkaitan dengan metode, teknik, atau cara untuk mengumpulkan data, mengolah data, menyajikan data, menganalisa data, dan menarik kesimpulan atau menginterpretasikan data¹⁸.

2.4.1. Data dan Uji Kecukupan Data

Data adalah kumpulan angka-angka yang belum mempunyai makna atau arti apa-apa¹⁹. Dalam statistik, jenis data tergantung pada konteksnya. Data yang berupa angka disebut data kuantitatif, yang nilainya bisa berubah-ubah (variabel). Berdasarkan nilainya dikenal dua jenis data kuantitatif, yaitu data diskrit dan data kontinyu. Data diskrit diperoleh dari hasil perhitungan. Menurut sumbernya, data dibedakan menjadi dua jenis, yaitu data internal dan data eksternal. Data internal adalah data yang diperoleh atau bersumber dari dalam suatu instansi (lembaga, organisasi). Data eksternal adalah data yang diperoleh atau bersumber dari luar instansi. Data eksternal dibagi menjadi dua jenis, yaitu data primer dan data sekunder. Data primer adalah data yang langsung dikumpulkan oleh orang yang berkepentingan atau yang memakai data tersebut. Data primer dapat diperoleh dari wawancara atau kuesioner. Sedangkan data sekunder adalah data yang tidak langsung dikumpulkan oleh orang yang berkepentingan dengan data tersebut. Contoh data sekunder adalah laporan tahunan perusahaan untuk keperluan menulis skripsi dan begitu juga data kepustakaan.

Sebelum diolah lebih lanjut, data yang dikumpulkan harus melalui uji kecukupan untuk mengetahui kesalahan dalam menentukan nilai waktu rata-rata sebuah elemen untuk jumlah observasi tertentu. Jika dalam uji ini tingkat kepercayaan yang digunakan besarnya 95% dengan tingkat ketepatan 10%, berarti bahwa kemungkinan kesalahan pada 95 dari 100 observasi rata-rata yang diambil, besarnya tidak lebih dari $\pm 10\%$. Karena itu, digunakan rumusan berikut:

$$N' = \left[\frac{20\sqrt{N \sum x^2 - (\sum x)^2}}{\sum x} \right]^2 \dots\dots\dots(2.1)$$

¹⁸ Boediono dan Wayan Koster. *Teori dan Aplikasi Statistika dan Probabilitas*. Bandung: Remaja Rosdakarya. 2002: h 6.

¹⁹ Boediono dan Wayan Koster. *Ibid*. h 6.

dengan: N' = jumlah observasi yang diperlukan

N = jumlah observasi aktual yang dilakukan

Jika $N' < N$ maka jumlah observasi aktual yang dilakukan sudah memenuhi syarat kecukupan data. Jumlah kecukupan data yang dikumpulkan dapat ditentukan secara lebih akurat menggunakan metode statistik. Karena pengumpulan data dengan *time study* merupakan prosedur *sampling*, maka observasi dapat diasumsikan terdistribusi secara normal. Dengan rata-rata sampel, \bar{x} dan standar deviasi sampel, s , distribusi normal untuk ukuran sampel besar ($n \geq 30$) maka *tingkat kepercayaan*-nya:

$$\bar{x} \pm z \sqrt{\frac{s}{n}} \dots\dots\dots(2.2)$$

dengan:

$$s = \sqrt{\frac{\sum_{i=1}^{i=n} (x_i - \bar{x})^2}{n - 1}} \dots\dots\dots(2.3)$$

Sedang untuk sampel berukuran kecil ($n < 30$), digunakan distribusi t , maka *tingkat kepercayaan*-nya:

$$\bar{x} \pm t \frac{s}{\sqrt{n}} \dots\dots\dots(2.4)$$

jadi jumlah data observasi yang dibutuhkan dihitung menurut rumus berikut:

$$n = \left\{ \frac{ts}{k\bar{x}} \right\}^2 \dots\dots\dots(2.5)$$

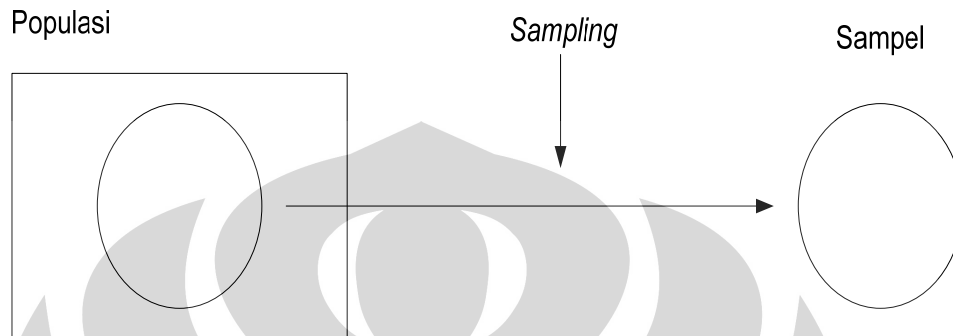
dengan t adalah probabilitas n pada k tertentu.²⁰

2.4.2. *Sampling*

Insiyur dan ilmuwan seringkali harus mengkaji suatu sistem fisik dimana fungsi distribusi sebuah karakteristik populasi tidak diketahui. Dalam situasi seperti itu, mereka harus mempercayakan data dari sampel dalam membantu memahami sifat-sifat distribusi tersebut. Karena informasi yang diperoleh dari data sampel tidak mungkin lebih baik daripada informasi yang sesungguhnya pada populasi (yang bisa diperoleh melalui sensus), maka teknik *sampling* menjadi sangat

²⁰ Benjamin Niebel dan Andris Freivalds. *Methods, Standards, and Work Design*. ed. ke-11. McGraw Hill. h 393-340.

berguna dalam upaya penarikan kesimpulan (*inference*) yang valid dan dapat dipercaya. *Sampling* adalah cara mengumpulkan data dari sampel. Tentu saja ada perbedaan antara populasi dengan sampel yang dapat dilihat dari gambar dan tabel berikut:



Gambar 2.7 *Sampling* (Sumber: Boediono dan Wayan Koster. 2002: h 365)

Tabel 2.2 Perbedaan antara Populasi dengan Sampel

Karakteristik Populasi	Karakteristik Sampel
Ukuran N	Ukuran n
Parameter	Statistik
<i>Mean</i> μ	<i>Mean</i> \bar{x}
Simpangan baku σ	Simpangan baku S
Populasi berhingga atau tak berhingga	Sampel besar atau kecil

(Sumber: Boediono dan Wayan Koster. 2002: h 365)

Dengan dilakukannya *sampling* yang baik akan diperoleh keuntungan berupa penghematan biaya dan waktu tanpa harus mengorbankan keakuratan hasil-hasilnya. Ada empat teknik *sampling* yang terkenal, yaitu:

1. Teknik *Sampling* Acak Sederhana

Teknik ini adalah pengambilan sampel sebanyak n sedemikian rupa sehingga; (1) setiap unit dalam populasi mempunyai kesempatan yang sama untuk terambil, dan (2) setiap ukuran sampel n juga mempunyai kesempatan yang sama untuk terambil. Teknik pengambilan acak sederhana ini dapat dilakukan dengan memakai tabel bilangan acak dan dengan memakai undian (lotere). Selain dengan tabel, bilangan acak juga dapat diperoleh dengan kalkulator dan komputer. Untuk populasi berhingga atau populasi terbatas yang kecil, teknik pengambilan sampel ini dengan mudah dapat dilakukan.

2. Teknik *Sampling* Acak Sistematis

Teknik ini dilakukan dengan mengambil setiap unsur ke-k dalam populasi, untuk dijadikan sampel dengan titik awal ditentukan secara acak di antara k unsur yang pertama. Teknik ini sangat mudah dilakukan sehingga banyak dipakai dengan menganggap seolah-olah sampel yang diperoleh merupakan sampel acak. Sesungguhnya, sampel yang diperoleh dengan teknik ini dapat menghasilkan kesimpulan yang lebih tepat mengenai parameter populasi, karena nilai-nilai pengamatan sampel menyebar secara merata di seluruh populasi.

3. Teknik *Sampling* Acak Stratifikasi

Teknik ini dilakukan dengan membagi-bagi populasi menjadi beberapa kelompok (strata) sehingga setiap kelompok akan menjadi seragam atau homogen dan kemudian unit sampel dipilih secara acak dengan atau tanpa pengembalian pada setiap kelompok tersebut. Teknik ini sangat baik dipakai terutama untuk populasi yang diketahui tidak homogen. Dengan teknik ini, maka pada setiap kelompok unit-unit populasi akan menjadi lebih homogen sehingga menghasilkan statistik yang lebih tepat untuk menyimpulkan parameter populasi. Banyaknya bagian populasi (sub populasi) pada setiap kelompok tidak perlu sama. Dengan demikian kita harus memperhatikan banyaknya sampel yang akan diambil pada setiap kelompok tersebut. Salah satu cara yang banyak dipakai adalah dengan alokasi sebanding atau alokasi proporsional, yaitu mengambil sampel pada setiap kelompok populasi yang sebanding dengan ukuran populasi pada setiap kelompok tersebut. Bila suatu populasi berukuran N dibagi menjadi k kelompok (strata) yang masing-masing berukuran $N_1, N_2, N_3, \dots, N_k$ dan pada setiap kelompok itu diambil sampel acak sederhana berupa $n_1, n_2, n_3, \dots, n_k$, maka alokasinya akan sebanding jika:

$$n_i = n \times \frac{N_i}{N} \dots \dots \dots (2.6)$$

Dimana $i = 1, 2, 3, \dots, k$ dan $n =$ ukuran sampel keseluruhan $= n_1 + n_2 + n_3 + \dots + n_k$.

4. Teknik *Sampling* Acak Kluster

Teknik ini dilakukan dengan mengambil beberapa kluster (kelompok) dan setelah kluster itu diambil, semua atau sebagian unit dalam setiap kluster diambil secara acak sebagai sampel. Teknik ini seringkali dianggap sebagai cara yang lebih efisien dari segi biaya bilamana populasinya tersebar luas.

2.4.3. Probabilitas

Perumusan konsep dasar probabilitas dilakukan dengan dua cara yaitu dengan cara klasik dan cara frekuensi relatif. Bila semua kejadian dilambangkan dengan huruf besar E, maka probabilitas kejadian E dirumuskan dengan P (E).

1) Perumusan Klasik

Bila kejadian E terjadi dalam m cara dari seluruh n cara yang mungkin terjadi dan setiap n cara itu mempunyai kesempatan atau kemungkinan yang sama untuk muncul, maka probabilitas kejadian E yang ditulis P (E) dirumuskan sebagai berikut: $P(E) = \frac{m}{n}$(2.7)

2) Perumusan dengan Frekuensi Relatif

Perumusan dengan cara klasik mempunyai kelemahan karena menuntut syarat semua hasil mempunyai kemungkinan yang sama untuk muncul. Pengertian ini mengaburkan adanya probabilitas yang sama. Maka dikembangkan konsep probabilitas berdasarkan statistik, yaitu dengan pendekatan empiris. Probabilitas empiris dari suatu kejadian dirumuskan dengan memakai frekuensi relatif dari terjadinya suatu kejadian dengan syarat banyaknya pengamatan atau banyaknya sampel n adalah sangat besar. Bila n bertambah besar sampai tak berhingga ($n \rightarrow \infty$), maka probabilitas dari kejadian E adalah sama dengan nilai limit dari frekuensi relatif dari kejadian E tersebut. Dengan demikian, jika kejadian E terjadi sebanyak f kali dari keseluruhan pengamatan sebanyak n, dimana n mendekati tak berhingga ($n \rightarrow \infty$), maka probabilitas kejadian E dirumuskan sebagai : $P(E) = \lim_{n \rightarrow \infty} \frac{f}{n}$(2.8)

Walaupun mudah dan berguna dalam praktek, tetapi secara matematis perumusan konsep probabilitas dengan frekuensi relatif ini juga mempunyai kelemahan, karena suatu nilai limit yang benar-benar mungkin

sebenarnya tidak ada. Oleh sebab itu, konsep probabilitas modern dikembangkan dengan memakai pendekatan aksiomatis, yaitu suatu kebenaran yang diterima secara apa adanya tanpa memerlukan bukti matematis, dimana konsep probabilitas tidak didefinisikan, seperti konsep titik dan konsep garis yang tidak didefinisikan dalam ilmu geometri.

2.4.4. Distribusi Probabilitas

Banyak eksperimen probabilitas memiliki keluaran (*outcomes*) yang berupa pengamatan numerik (angka/bilangan), cacah/hitungan (*counts*) atau pengukuran (*measurement*). Sebuah variabel acak (*random variable*), biasa didefinisikan dengan sebuah simbol, misalnya X , adalah variabel yang memiliki sebuah nilai numerik tunggal untuk setiap keluaran dari sebuah eksperimen probabilitas. Jadi X dapat bernilai angka berapapun yang dikaitkan dengan keluaran dari sebuah eksperimen probabilitas. Jadi X dapat bernilai angka berapapun yang dikaitkan dengan keluaran yang mungkin dari eksperimen tersebut. Dengan kata lain, nilai tertentu dari X dalam sebuah eksperimen adalah kemungkinan keluaran yang acak. Sebuah variabel acak diskrit adalah variabel acak yang dapat dicacah (*countable*). Sementara sebuah variabel acak kontinu memiliki nilai yang tak terhingga banyaknya sepanjang sebuah interval yang tidak terputus. Sebuah variabel acak kontinu biasanya diperoleh dari hasil pengukuran.

Berdasarkan jenis variabel acak tersebut, maka distribusi probabilitas terbagi dua, yaitu distribusi probabilitas diskrit dan distribusi probabilitas kontinu. Distribusi probabilitas diskrit memiliki beberapa jenis seperti distribusi Bernoulli, distribusi Binomial, distribusi Binomial negatif, distribusi Geometrik, distribusi Hipergeometrik, dan distribusi Poisson. Sedangkan distribusi probabilitas kontinu memiliki beberapa jenis seperti distribusi Normal (Gaussian), distribusi Gamma, distribusi Chi-Kuadrat, distribusi Eksponensial, distribusi Weibull, dan distribusi Lognormal. Di sini hanya akan dijelaskan mengenai distribusi Binomial dan distribusi Poisson sebagai perwakilan dari distribusi probabilitas diskrit; dan juga dijelaskan distribusi Normal (Gaussian), distribusi Gamma, distribusi Eksponensial, dan distribusi Lognormal sebagai perwakilan dari distribusi probabilitas kontinu.

1. Distribusi Binomial

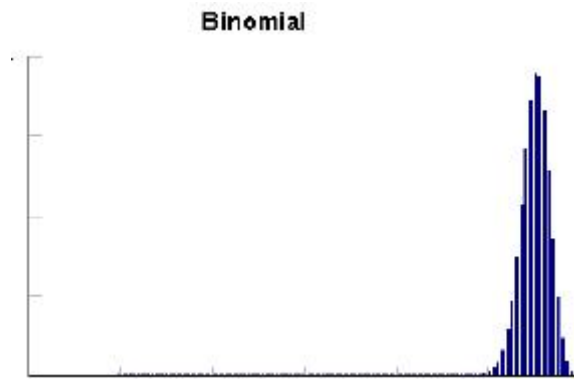
Distribusi binomial adalah salah satu distribusi probabilitas diskrit yang paling sering digunakan dalam analisa statistik modern. Di bidang teknik, distribusi ini erat kaitannya dengan pengendalian kualitas (*quality control*). Suatu distribusi binomial dibentuk oleh suatu eksperimen binomial. Eksperimen ini merupakan n kali percobaan Bernoulli, sehingga harus memenuhi kondisi-kondisi berikut:

- Jumlah percobaan n adalah konstanta yang telah ditentukan sebelumnya (dinyatakan sebelum eksperimen dimulai)
- Setiap pengulangan eksperimen, yang biasa disebut percobaan (trial), hanya dapat menghasilkan satu dari dua keluaran yang mungkin: sukses atau gagal
- Probabilitas sukses p , dan demikian pula probabilitas gagal $q = 1-p$ selalu konstan dalam setiap percobaan (trial)
- Setiap percobaan (trial) saling bebas secara statistik, yang berarti keluaran suatu percobaan tidak berpengaruh pada keluaran percobaan lainnya

$$p(x) = \binom{n}{x} p^x (1-p)^{n-x} \quad \dots\dots\dots (2.9)$$

$$\binom{n}{x} = \frac{n!}{x!(n-x)!} \quad \dots\dots\dots (2.10)$$

Dimana : $x = 0, 1, \dots, n$
 $n =$ jumlah percobaan
 $p =$ probabilitas kejadian



Gambar 2.8 Grafik Distribusi Binomial

(Sumber: Stat::Fit User Guide. h 68. ProModel Documentation)

2. Distribusi Poisson

Distribusi ini digunakan untuk mengamati jumlah kejadian-kejadian khusus yang terjadi dalam satuan waktu atau ruang. Suatu distribusi Poisson dapat digunakan dengan tepat dalam suatu eksperimen Poisson yang memenuhi kondisi-kondisi berikut:

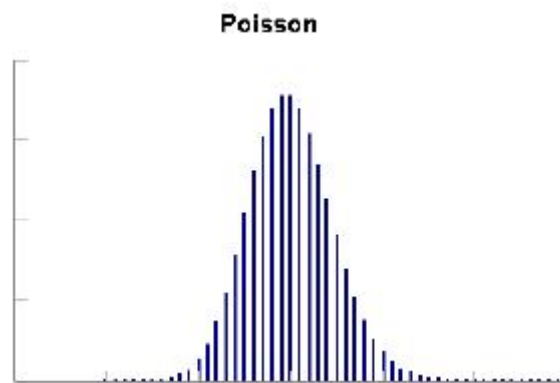
- Suatu eksperimen yang meliputi pencacahan banyaknya suatu peristiwa terjadi dalam setiap satuan unit yang ditentukan. Unit yang ditentukan ini biasanya adalah unit waktu atau ruang
- Probabilitas peristiwa tersebut adalah sama untuk setiap satuan unit
- Banyaknya peristiwa yang terjadi dalam setiap satuan unit saling bebas terhadap banyaknya peristiwa yang terjadi pada setiap satuan unit lainnya

$$p(x) = \frac{e^{-\lambda} \lambda^x}{x!} \dots\dots\dots(2.11)$$

Dimana : $x = 0,1,\dots,n$

λ = tingkat kedatangan

e = konstanta dasar logaritma natural = 2,71828.....



Gambar 2.9 Grafik Distribusi Poisson

(Sumber: Stat::Fit User Guide. hal 89. ProModel Documentation)

3. Distribusi Normal (Gaussian)

Distribusi Normal (Gaussian) mungkin merupakan distribusi probabilitas paling penting baik dalam teori maupun aplikasi statistik. Terminologi normal itu sendiri bukannya berarti tidak pada tempatnya, karena memang distribusi ini adalah yang paling banyak digunakan sebagai model bagi data riil di berbagai bidang yang meliputi antara lain karakteristik fisik makhluk hidup (berat, tinggi badan manusia, hewan, dll), kesalahan-kesalahan pengukuran dalam eksperimen ilmiah, pengukuran-pengukuran intelegensi dan perilaku, nilai skor berbagai pengujian, dan berbagai ukuran-ukuran dan indikator ekonomi. Dan bahkan meskipun variabel yang ditangani dalam distribusi adalah variabel diskrit, kurva distribusi normal sering juga digunakan sebagai pendekatan. Sekurang-kurangnya terdapat empat alasan mengapa distribusi normal menjadi paling penting:

- Distribusi normal terjadi secara alamiah. Seperti diuraikan sebelumnya banyak peristiwa di dunia nyata yang terdistribusi secara normal
- Beberapa variabel acak yang bahkan tidak terdistribusi secara normal dapat dengan mudah ditransformasi menjadi suatu distribusi variabel acak yang normal

- Banyak hasil dan teknik analisa yang berguna dalam pekerjaan statistik hanya bisa berfungsi dengan benar jika model distribusinya merupakan distribusi normal.
- Beberapa variabel acak yang tidak menunjukkan distribusi normal pada populasinya namun distribusi dari rata-rata sampel yang diambil secara acak dari populasi tersebut menunjukkan distribusi normal

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{[x - \mu]^2}{2\sigma^2}\right) \dots\dots\dots(2.12)$$

Dimana: μ = parameter pergeseran=*mean*
 σ = parameter skala = *standard deviation*

Normal



Gambar 2.10 Grafik Distribusi Normal

(Sumber: Stat::Fit User Guide. h 84. ProModel Documentation)

4. Distribusi Gamma

Meskipun distribusi normal mempunyai penerapan yang luas di berbagai bidang, dalam kenyataannya terdapat situasi dimana hasil-hasil eksperimen menunjukkan distribusi yang tidak simetris ataupun tidak menunjukkan kecenderungan simetris. Untuk itu, model distribusi normal tidak dapat memberikan hasil yang tepat jika digunakan. Untuk eksperimen-eksperimen probabilitas yang hasilnya menunjukkan suatu bentuk distribusi yang mempunyai variasi ukuran kemiringan yang cukup

signifikan, distribusi gamma merupakan salah satu alternatif model yang banyak digunakan. Terdapat gambaran yang berbeda dari distribusi ini yang bergantung dari nilai α .

- $\alpha=1$ distribusi gamma akan menjadi distribusi Eksponensial (gambar 2.11 paling kiri)
- α dibatasi untuk nilai bulat positif, maka distribusi gamma akan menjadi distribusi Erlang.
- $\alpha < 1$, grafik cenderung bernilai tidak terhingga pada nilai x minimum dan ketika x naik maka grafik akan menurun (gambar 2.11 tengah)
- $\alpha > 1$, grafik bernilai 0 pada x minimum kemudian meningkat bergantung pada nilai alpha dan beta nya selantutnya akan menurun (gambar 2.11 paling kanan)

Distribusi ini digunakan untuk menggambarkan umur benda, *lead time*, data pendapatan seseorang, populasi dengan keseimbangan yang stabil, waktu antar kedatangan dan waktu pelayanan. Persamaan yang digunakan :

$$f(x) = \frac{(x - \min)^{\alpha-1}}{\beta^{\alpha}\Gamma(\alpha)} \exp\left(-\frac{[x - \min]}{\beta}\right) \dots\dots\dots(2.13)$$

Dimana: min = minimum x
 α = parameter bentuk > 0
 β = parameter skala > 0



Gambar 2.11 Grafik Distribusi Gamma

(Sumber: Stat.:Fit User Guide. h 68. ProModel Documentation)

5. Distribusi Eksponensial

Distribusi eksponensial merupakan kasus khusus dari distribusi gamma dengan faktor bentuk $\alpha = 1$ dan $\beta = 1/\lambda$. distribusi ini banyak digunakan sebagai model di bidang teknik dan sains.

$$f(x) = \frac{1}{\beta} \exp\left(-\frac{[x - \text{min}]}{\beta}\right) \dots\dots\dots(2.14)$$

Dimana: min = minimum nilai x
 β = parameter skala = *mean*

Exponential



Gambar 2.12 Grafik Distribusi Eksponensial

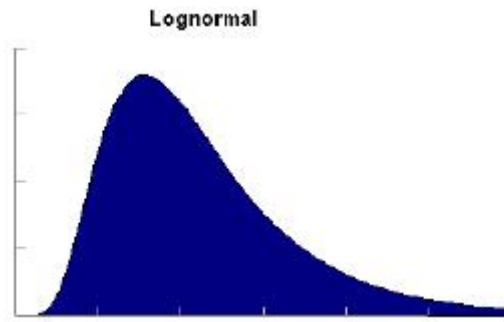
(Sumber: Stat::Fit User Guide. h 66. ProModel Documentation)

6. Distribusi Lognormal

Distribusi lognormal merupakan distribusi teoritis yang banyak digunakan di bidang teknik, khususnya sebagai model untuk berbagai jenis sifat material.

$$f(x) = \frac{1}{(x - \text{min})\sqrt{2\pi\sigma^2}} \exp\left(-\frac{[\ln(x - \text{min}) - \mu]^2}{2\sigma^2}\right) \dots\dots\dots(2.15)$$

Dimana: min = minimum x
 μ = *mean* keadaan Normal
 σ = standar deviasi keadaan Normal



Gambar 2.13 Grafik Distribusi Lognormal

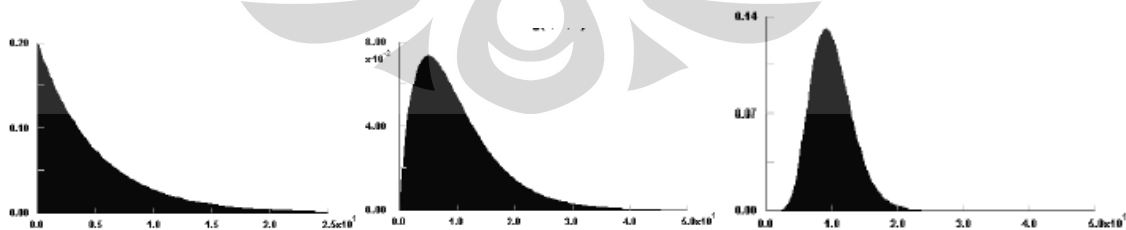
(Sumber: Stat::Fit User Guide. h 80. ProModel Documentation)

7. Distribusi Erlang

Distribusi Erlang merupakan distribusi kontiniu yang dibatasi pada sisi bawahnya. Notasi yang digunakan distribusi ini adalah (min, m, beta). Distribusi ini merupakan bentuk khusus dari distribusi gamma untuk parameter, m , dibatasi pada bilangan bulat positif. Ketika m=1, distribusi ini berubah menjadi distribusi eksponensial. Distribusi ini telah secara luas digunakan dalam teori *reliability* dan teori antrian. Persamaan yang digunakan

$$f(x) = \frac{(x - \min)^{m-1}}{\beta^m \Gamma(m)} \exp\left(-\frac{[x - \min]}{\beta}\right) \dots\dots(2.16)$$

Di mana: min = minimum x
 m = faktor bentuk = bulat positif
 β = faktor skala > 0



Gambar 2.14 Grafik Distribusi Erlang

(Sumber: Stat::Fit User Guide. h 63. ProModel Documentation)

2.4.5. Uji Kesesuaian Distribusi Probabilitas

Uji kesesuaian distribusi probabilitas pada hakikatnya merupakan perbandingan antara data masukan (*input*) dengan distribusi-distribusi yang bersesuaian dengan suatu cara statistik yang signifikan. Setiap pengujian membuat hipotesa bahwa kesesuaian telah baik dan menghitung suatu pengujian statistik sebagai perbandingan terhadap distribusi yang standar. Uji kesesuaian distribusi probabilitas meliputi pengujian Chi-kuadrat, pengujian Kolmogorov Smirnov, dan pengujian Anderson Darling. Jika pemilihan pengujian tidak pasti, bahkan setelah penjelasan terhadap setiap metode pengujian yang nanti akan dijelaskan, maka penggunaan pengujian Kolmogorov Smirnov dapat dipercaya terhadap kisaran data yang sangat luas dan parameter yang bersesuaian.

1. Pengujian Chi-kuadrat

Pengujian Chi-kuadrat adalah suatu pengujian kesesuaian dari kepadatan yang sesuai ke dalam data masukan (*input*) pada tabel data, dengan mana data yang berdekatan dipisahkan ke dalam interval (untuk data kontinyu) atau kelas (untuk data diskrit)²¹. Pengujian bermula dari data observasi yang dikelompokkan ke dalam kelas (interval). Karena jumlah kelas untuk data diskrit ditentukan oleh kisaran bilangan bulat, pemilihan terhadap jumlah interval yang berdekatan menjadi tidak ditentukan dengan baik. Pengujian kemudian menghitung nilai yang diharapkan (*expected value*) untuk setiap interval yang bersesuaian, dimana *expected value* tersebut mencakup jumlah atau integral (+/-) ke suatu ketidakterbatasan, atau ke nilai yang terdekat. Untuk membuat pengujian menjadi valid, interval (kelas) yang memiliki data kurang dari lima buah dimasukkan ke dalam tetangganya sampai interval tersebut memiliki sedikitnya data berjumlah lima buah. Kemudian data tersebut dihitung dengan persamaan Chi-kuadrat:

$$\chi^2 = \sum_{i=1}^k \frac{(n_i - np_i)^2}{np_i} \dots\dots\dots(2.17)$$

²¹ H.D. Brunk. *An Introduction to Mathematical Statistics*, Ginn&Co. 1960: h 261.

Dimana: X^2 adalah lambang persamaan Chi-kuadrat, n adalah total jumlah data, n_i adalah jumlah data pada interval (kelas) ke- i , k adalah jumlah interval (kelas), dan p_i adalah probabilitas kejadian yang diharapkan dari interval (kelas) untuk kesesuaian distribusi. Hasil pengujian Chi-kuadrat kemudian dibandingkan dengan nilai Chi-kuadrat yang standar pada angka derajat kebebasan yang berdekatan dan tingkat kepercayaan, biasanya disimbolkan dengan α . Meskipun pengujian Chi-kuadrat adalah suatu pengujian asimtotis yang hanya valid untuk jumlah data yang bertambah banyak, pengujian ini masih dapat digunakan untuk metode perbandingan²².

2. Pengujian Kolmogorov Smirnov

Pengujian Kolmogorov Smirnov (KS) adalah suatu pengujian kesesuaian yang menyesuaikan distribusi kumulatif yang bersesuaian ke dalam data masukan (input) pada tabel, nilai dengan nilai²³. Pengujian KS menghitung perbedaan absolut terbesar antara distribusi kumulatif pada data masukan (input) dengan distribusi yang bersesuaian menurut

$$D = \max(D^+, D^-)$$

$$D^+ = \max\left(\frac{i}{n} - F(x)\right), \quad i=1, \dots, n$$

persamaan: $D^- = \max\left(F(x) - \frac{(i-1)}{n}\right), \quad i=1, \dots, n$ (2.18)

Dimana: D adalah lambang persamaan KS, x adalah nilai dari data ke- i dari n jumlah total data, dan $F(x)$ adalah distribusi kumulatif yang bersesuaian. Perhatikan bahwa perbedaan positif dengan negatif didasarkan pada nilai dasar yang ditentukan sebelumnya. Hasil pengujian KS kemudian dibandingkan dengan nilai KS yang standar. Meskipun pengujian KS hanya valid jika tiada parameter dalam pengujian yang telah diperkirakan dari data, pengujian ini dapat digunakan untuk penyesuaian distribusi karena pengujian ini merupakan pengujian yang

²² Alan Stuart dan J. Keith Ord. *Kendall's Advanced Theory of Statistics*. Volume 2, Oxford University Press. 1991: h 1159.

²³ H.D. Brunk, *Loc. Cit.*

paling konservatif, yaitu, minimal menolak penyesuaian dalam kesalahan²⁴.

3. Pengujian Anderson Darling

Pengujian Anderson Darling (AD) adalah suatu pengujian kesesuaian yang menyesuaikan distribusi kumulatif yang bersesuaian ke dalam data masukan (input) pada tabel, dengan penekanan pada ujung distribusi. Pengujian ini menghitung integral dari perbedaan kuadrat antara data masukan (input) dengan distribusi yang bersesuaian menurut persamaan:

$$W^2 = n \int_{-\infty}^{\infty} \frac{[F_n(x) - F(x)]^2}{F(x)[1 - F(x)]} dF(x) \quad \dots\dots\dots(2.19)$$

Dimana: W^2 adalah lambang persamaan AD, n adalah jumlah data, $F(x)$ adalah distribusi kumulatif yang bersesuaian, dan $F_n(x)$ adalah distribusi kumulatif data masukan (*input*). Persamaan di atas dapat disederhanakan:

$$W_n^2 = n - \frac{1}{n} \sum_{i=1}^n (2i - 1) [\log \mu_i + \log(1 - \mu_{n-i+1})] \quad \dots\dots\dots(2.20)$$

Dimana: μ_i adalah nilai distribusi kumulatif yang bersesuaian, $F(x_i)$ adalah nilai data ke- i ²⁵. Hasil pengujian AD kemudian dibandingkan dengan nilai AD yang standar. Keterbatasan pengujian AD sama dengan keterbatasan pengujian KS dengan pengecualian bahwa pengujian AD dapat digunakan untuk berbagai ukuran sampel. Meskipun pengujian AD hanya valid jika tiada parameter dalam pengujian yang telah diperkirakan dari data, dapat digunakan untuk penyesuaian distribusi karena merupakan pengujian yang paling konservatif, yaitu, minimal menolak penyesuaian dalam kesalahan²⁶.

²⁴ Alan Stuart dan J. Keith Ord. *Loc. Cit.*

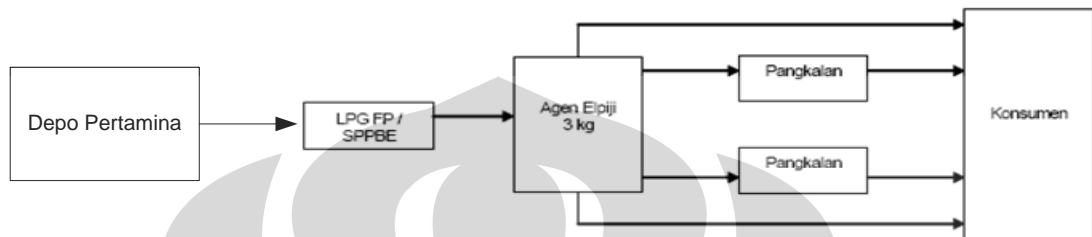
²⁵ T.W. Anderson dan D.A. Darling. *A Test of Goodness of Fit*. J. Am.Stat.Assoc. 1954: h 765.

²⁶ T.W. Anderson dan D.A. Darling. *Ibid.*

3. PENGUMPULAN DAN PENGOLAHAN DATA

3.1. SPBE Dalam Jalur Distribusi Elpiji

Jalur Distribusi LPG 3 kg berdasarkan "Pedoman Pencacahan dan Distribusi Elpiji 3 kg" , No. 1688/F10000/ 2007-S3 berlaku tmt. 1 Agustus 2007 adalah sebagai berikut:



Gambar 3.1 Jalur Distribusi Tabung Elpiji 3 Kg

(Sumber: "Pedoman Pencacahan dan Distribusi Elpiji 3 kg" , No. 1688/F10000/ 2007-S3)

Pada gambar di atas, terlihat bahwa posisi SPBE terletak di antara Depo Pertamina dengan Agen Elpiji 3 Kg. Depo Pertamina berfungsi sebagai pemasok *bulk* elpiji yang merupakan sumber pasokan gas untuk pengisian ke tabung-tabung elpiji. Agen elpiji 3 Kg sebagai distributor primer tabung-tabung elpiji yang nantinya akan dibeli oleh konsumen pengguna gas elpiji 3 Kg. Posisi SPBE yang strategis ini memainkan peranan penting sebagai penghubung antara penyuplai dengan distributor.

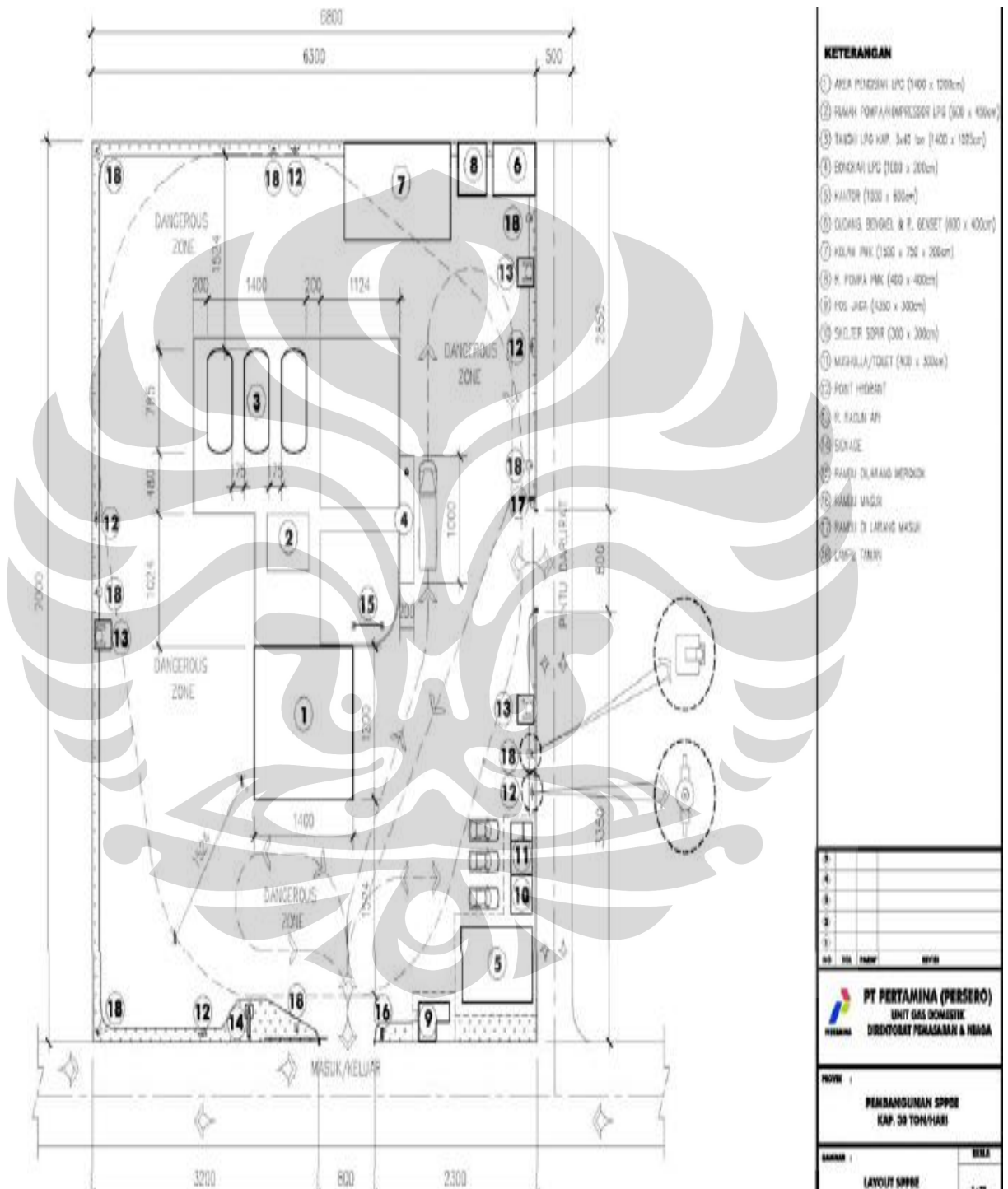
3.1.1. Sumber Pendapatan SPBE

Pendapatan yang diperoleh SPBE dari Pertamina ialah jasa pengisian dari *storage tank* milik SPBE ke tabung-tabung gas milik agen (berdasarkan ketentuan Pertamina) ialah sebesar Rp. 300 per kg.

3.2. Data Struktural

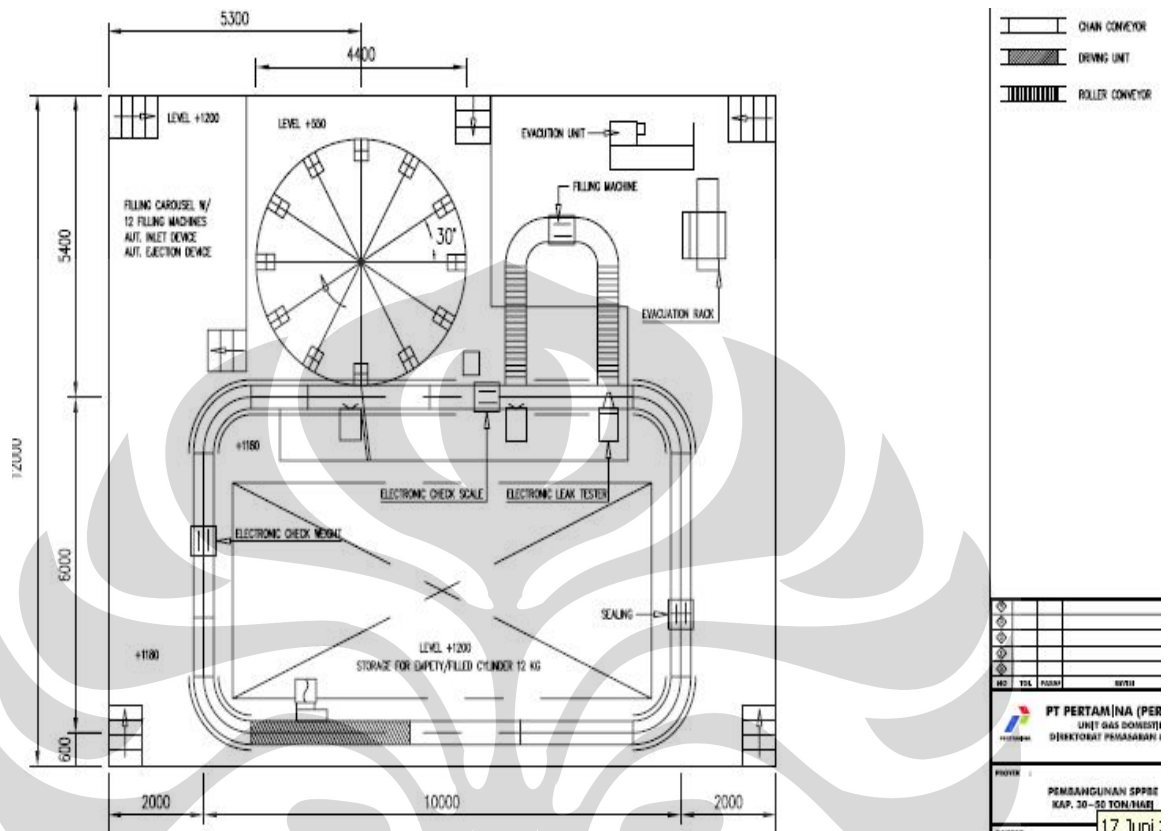
3.2.1. Layout SPBE

Gambar di bawah ini merupakan layout SPBE dengan kapasitas 30 ton per hari.



Gambar 3.2 Layout SPBE keseluruhan
 (Sumber: PT. Pertamina Unit Gas Domestik)

Sedangkan gambar di bawah ini ialah gambar area pengisian tabung gas dan *storage tank* (bagian dari layout SPBE keseluruhan).



Gambar 3.3 Area Pengisian Tabung Elpiji dan *Storage tank*

(Sumber: PT. Pertamina Unit Gas Domestik)

3.2.2. Fasilitas Operasional di SPBE

Fasilitas operasional yang dimiliki SPBE adalah

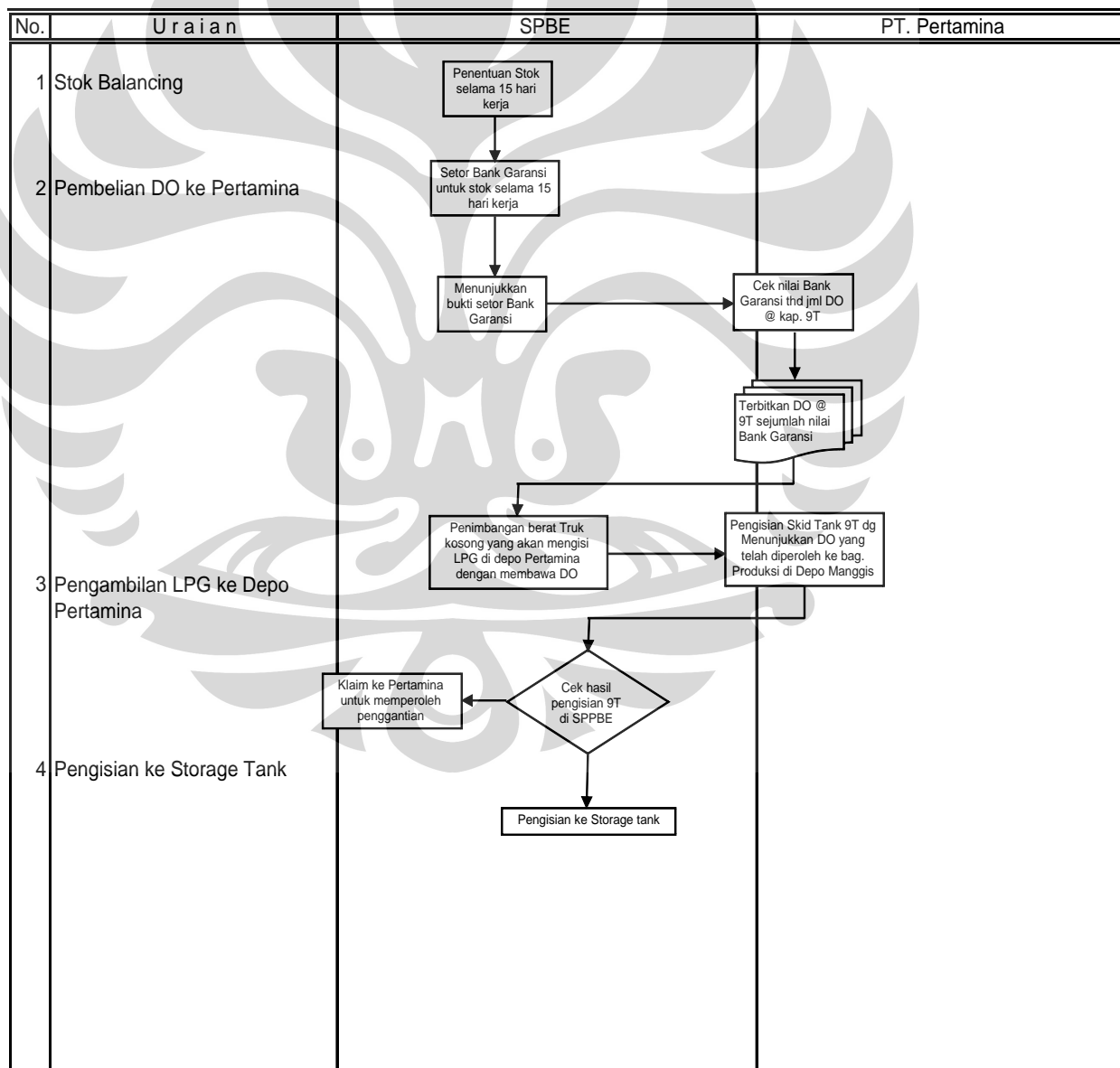
- 1) Tangki penyimpanan (*storage tank*) kapasitas 30 ton sebanyak tiga unit
- 2) Satu set CFM (*Carousel Filling Machine*) untuk pengisian tabung gas 3 kg yang terdiri atas 12 unit mesin tunggal (*filling point*)
- 3) Satu unit alat timbangan tabung (*electronic check weight*), alat periksa isi tabung (*electronic check scale*), mesin koreksi isi tabung (*correction scale*), alat uji kebocoran tabung (*electronic leak tester*), dan mesin evakuasi (*evacuation unit*).

3.3. Data Operasional

3.3.1. Bisnis Proses

Secara garis besar, bisnis proses di SPBE terdiri atas dua bagian yaitu pengisian *storage tank* dan pengisian tabung gas elpiji. Pengisian *storage tank* di SPBE baru dapat dilakukan setelah pengambilan *bulk* elpiji dari depo Pertamina. Sedangkan proses pengisian tabung gas elpiji baru dapat dilakukan setelah agen selesai mengurus pembayaran DO. Berikut ini disajikan diagram alir sebelum pengisian *storage tank* dan diagram alir sebelum pengisian tabung gas elpiji.

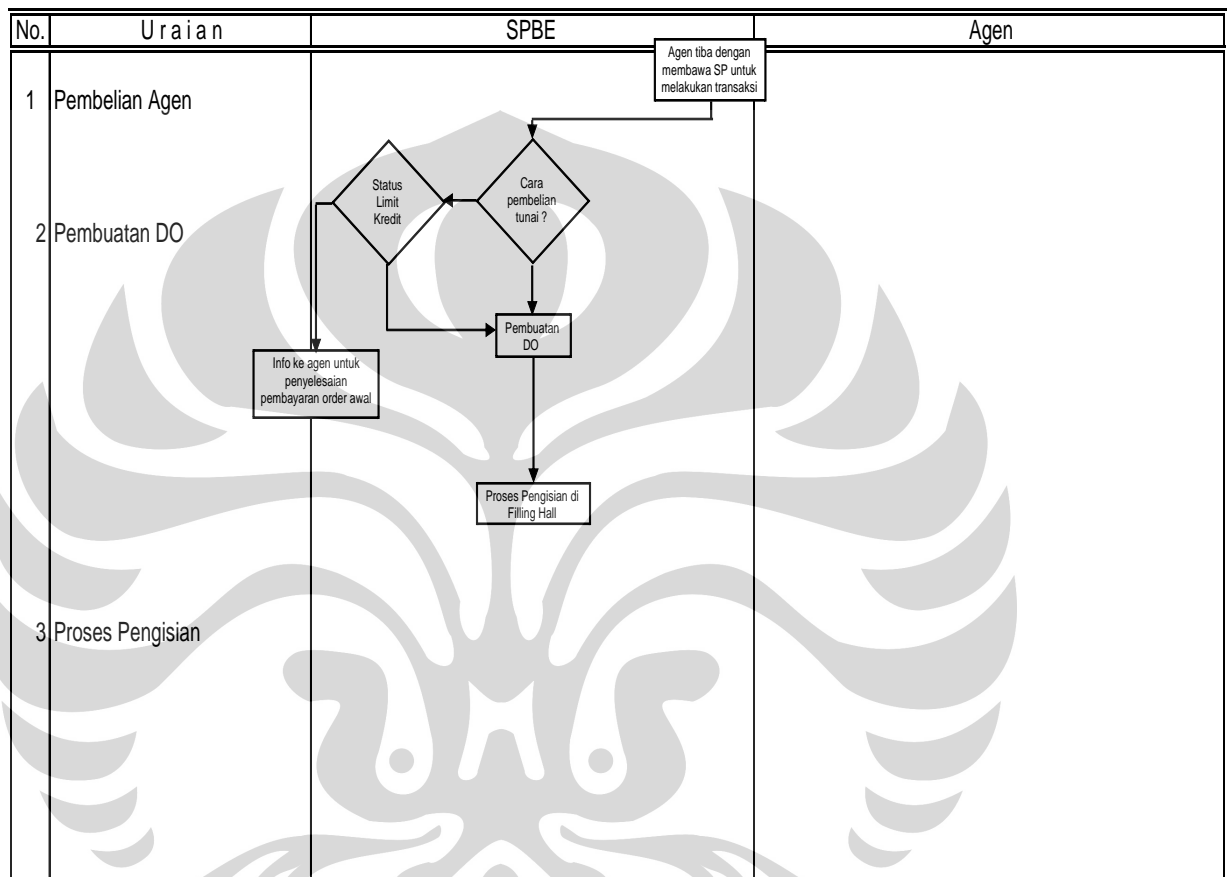
Diagram Alir Pengadaan LPG di SPBE



Gambar 3.4 Diagram Alir Pengadaan Elpiji di SPBE

Pada diagram alir di atas, disebutkan beberapa tahapan aliran proses yang harus dilakukan hingga dapat dilakukannya proses pengisian *bulk* elpiji ke *storage tank* di SPBE.

Diagram Alir Distribusi LPG oleh Agen dari SPBE

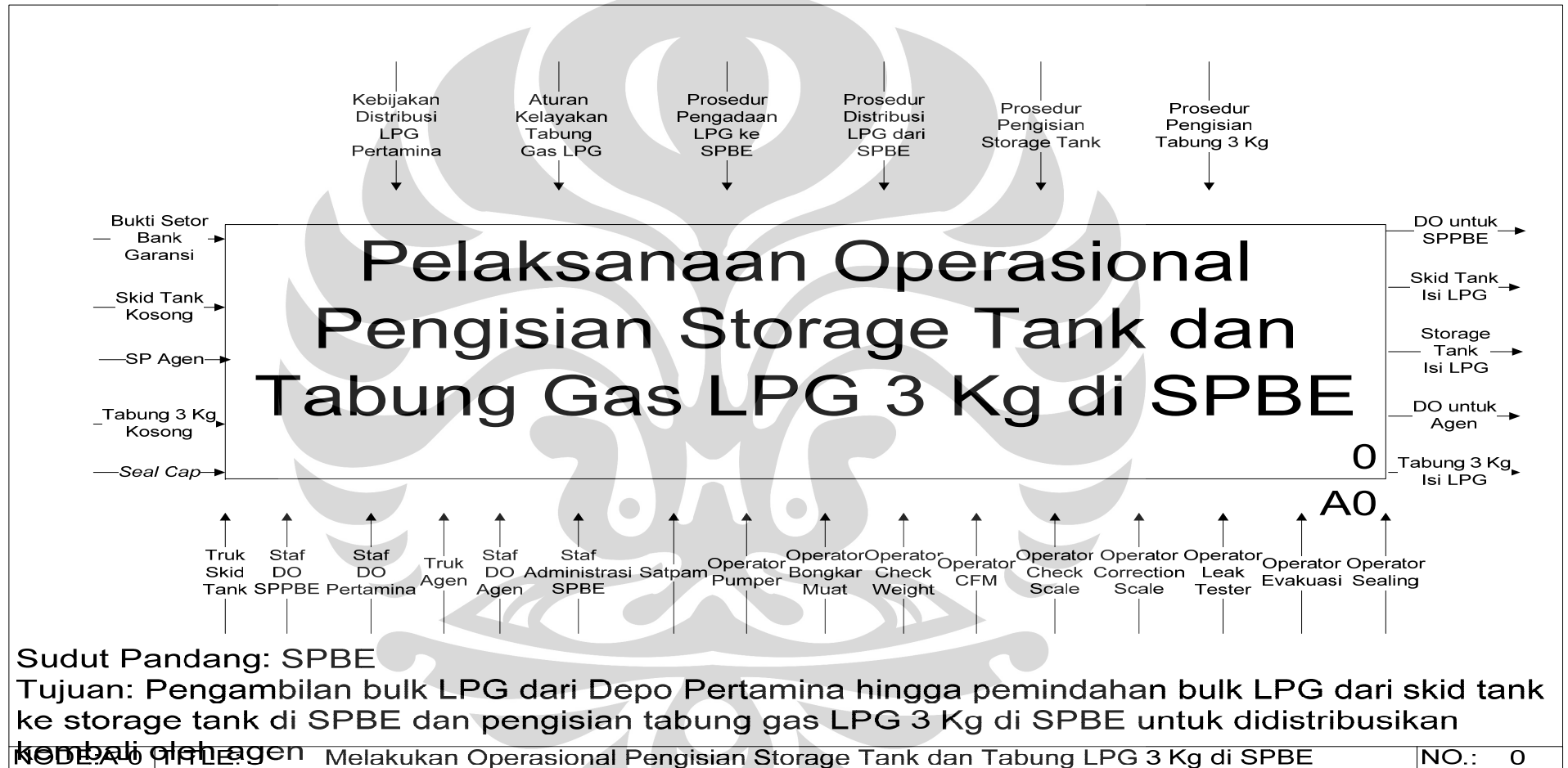


Gambar 3.5 Diagram Alir Distribusi LPG Oleh Agen dari SPBE

Pada diagram alir di atas, disebutkan beberapa tahapan aliran proses yang harus dilakukan hingga dapat dilakukannya proses pengisian tabung gas elpiji milik agen di SPBE.

3.3.2. Aliran Proses

Secara ringkas, kegiatan-kegiatan yang berhubungan dengan proses pengisian *storage tank* dan pengisian tabung gas elpiji di SPBE dapat dilihat pada diagram IDEF0 berikut.



Gambar 3.6 Diagram A-0 Pelaksanaan Operasional SPBE

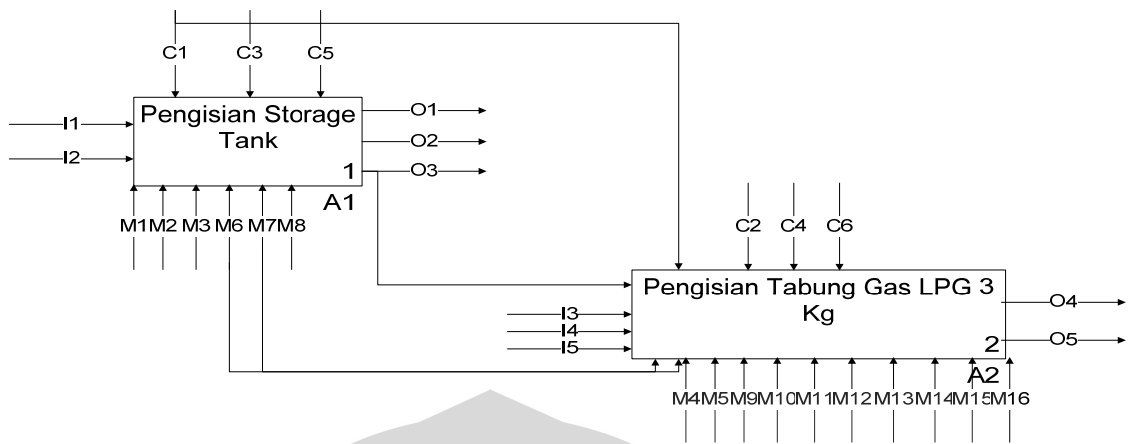


Diagram A0 Proses Pengisian *Storage tank* dan Tabung Gas

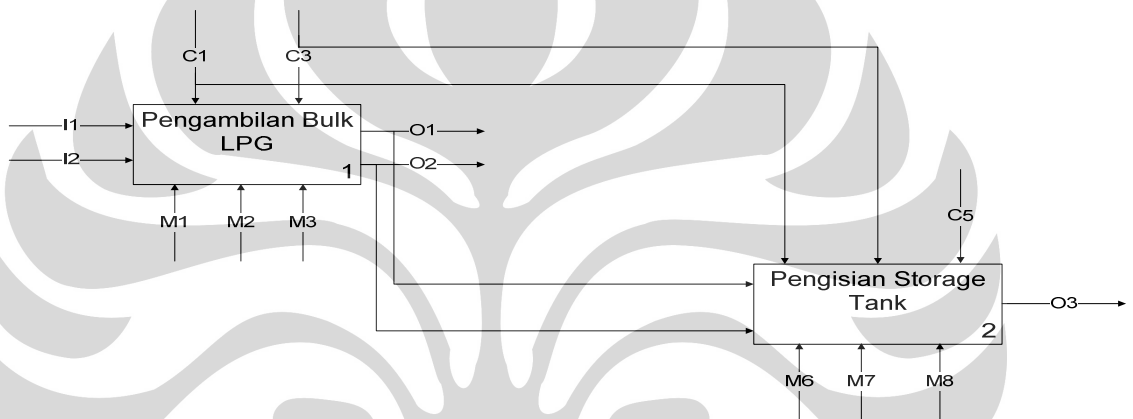


Diagram A1 Pengisian *Storage tank*

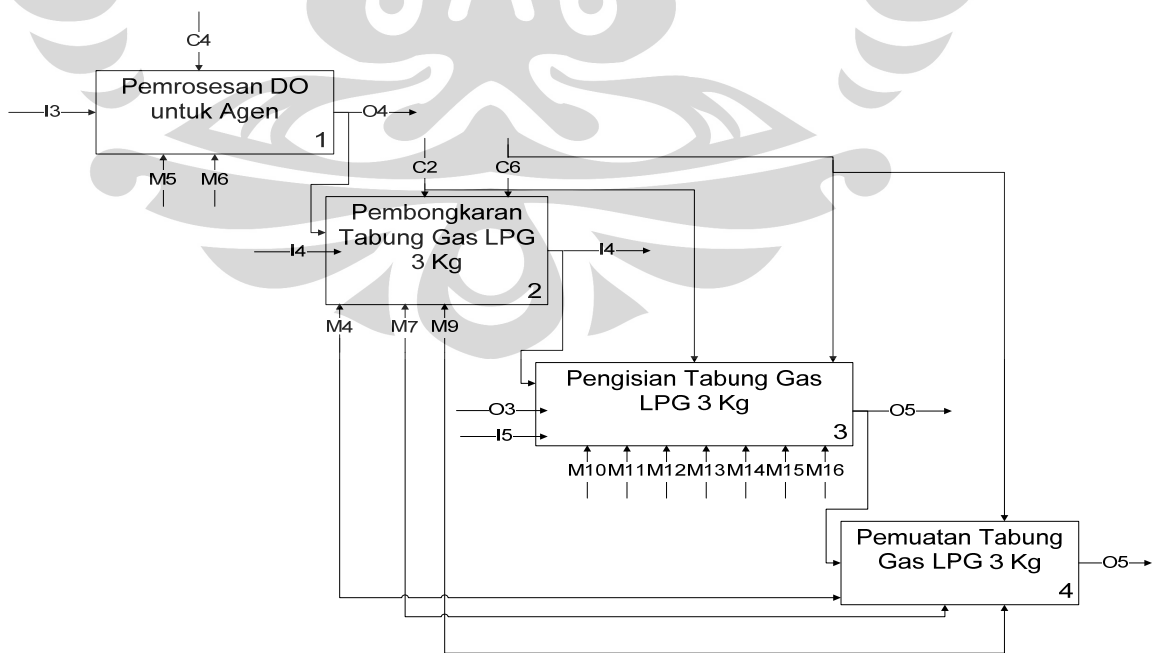


Diagram A2 Pengisian Tabung Gas

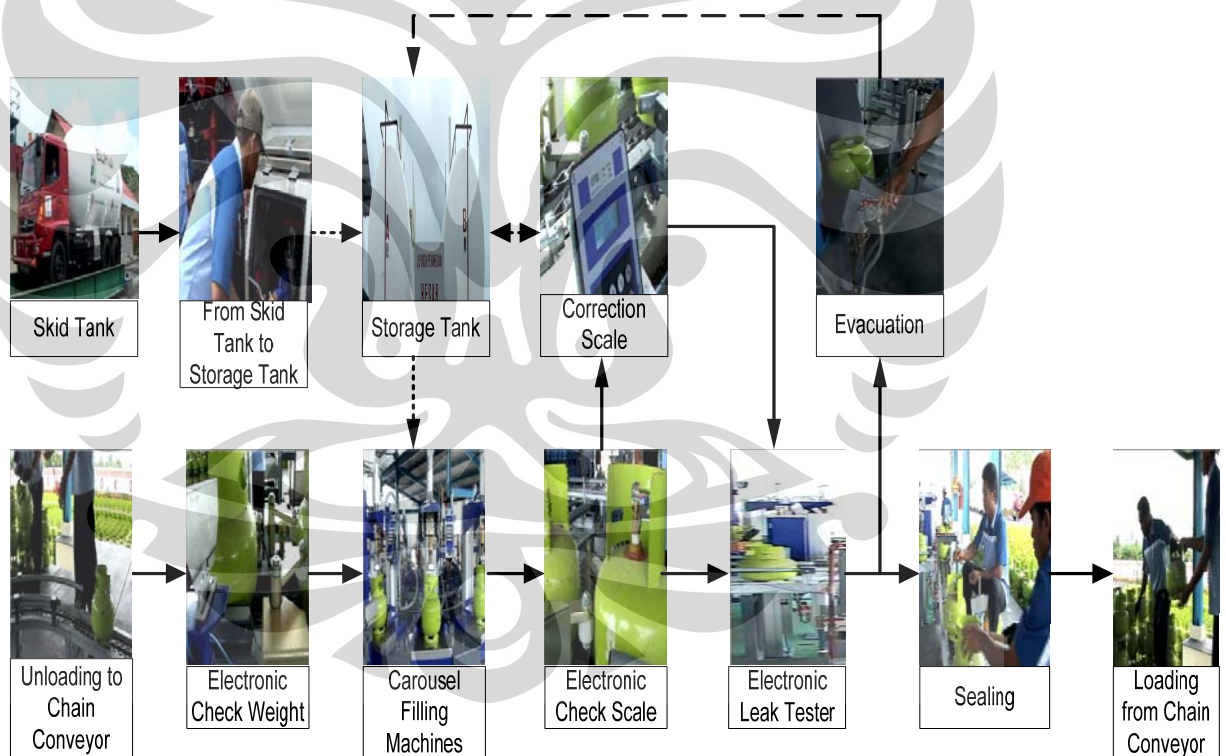
I1: Bukti Setor Bank Garansi
 I2: Skid Tank Kosong
 I3: SP Agen
 I4: Tabung 3 Kg Kosong
 I5: Seal Cap

C1: Kebijakan Distribusi LPG Pertamina
 C2: Aturan Kelayakan Tabung Gas LPG
 C3: Prosedur Pengadaan LPG ke SPBE
 C4: Prosedur Distribusi LPG dari SPBE
 C5: Prosedur Pengisian Storage Tank
 C6: Prosedur Pengisian Tabung 3 Kg

M1: Truk Skid Tank
 M2: Staf DO SPBE
 M3: Staf DO Pertamina
 M4: Truk Agen
 M5: Staf DO Agen
 M6: Staf Administrasi SPBE
 M7: Satpam
 M8: Operator Pumper
 M9: Operator Bongkar Muat
 M10: Operator Check Weight
 M11: Operator CFM
 M12: Operator Check Scale
 M13: Operator Correction Scale
 M14: Operator Leak Tester
 M15: Operator Evakuasi
 M16: Operator Sealing

O1: DO untuk SPBE
 O2: Skid Tank Isi LPG
 O3: Storage Tank Isi LPG
 O4: DO untuk Agen
 O5: Tabung 3 Kg Isi LPG

Peta aliran proses berikut ini dapat membantu memperjelas dua aliran proses utama di SPBE.



Gambar 3.7 Peta Aliran Proses di SPBE

Proses utama yang pertama ialah pengisian dari *skid tank* ke *storage tank* di SPBE. Nantinya, dengan bantuan kompresor, gas di *storage tank* dialirkan ke *Carousel Filling Machines* dan *Correction Scale*. Proses utama yang kedua ialah pengisian tabung gas elpiji 3 kg. Dimulai dengan pemindahan tabung gas elpiji 3

kg kosong dari truk agen ke *chain conveyor*. Kemudian tabung gas kosong akan ditimbang beratnya di *electronic check weight*. Kemudian tabung gas kosong diisi dengan gas oleh *carousel filling machines*. Setelah itu, tabung gas isi akan diperiksa isinya di *electronic check scale*. Jika isi tabung sesuai standar, maka tabung gas isi akan dikirim ke *leak tester*. Tetapi jika isi tabung tidak sesuai standar (kelebihan atau kekurangan gas), maka akan disesuaikan isinya di *correction scale*. Setelah isi tabung gas disesuaikan di *correction scale*, maka tabung gas isi dikirim ke *leak tester*. Di *leak tester*, dilakukan pengujian kebocoran isi tabung gas. Jika tabung tidak bocor, maka tabung gas dikirim ke *sealing*. Tetapi jika ada kebocoran, maka tabung gas akan dievakuasi oleh mesin evakuasi. Di area evakuasi, isi tabung gas yang bocor dialirkan kembali ke *storage tank*. Di *sealing*, dilakukan penyegelan tutup tabung. Kemudian tabung gas yang telah disegel diangkut ke atas truk agen. Garis panah yang utuh pada gambar menunjukkan aliran benda yang bersifat diskrit, sedangkan garis panah putus-putus menandakan aliran gas yang bersifat kontinyu.

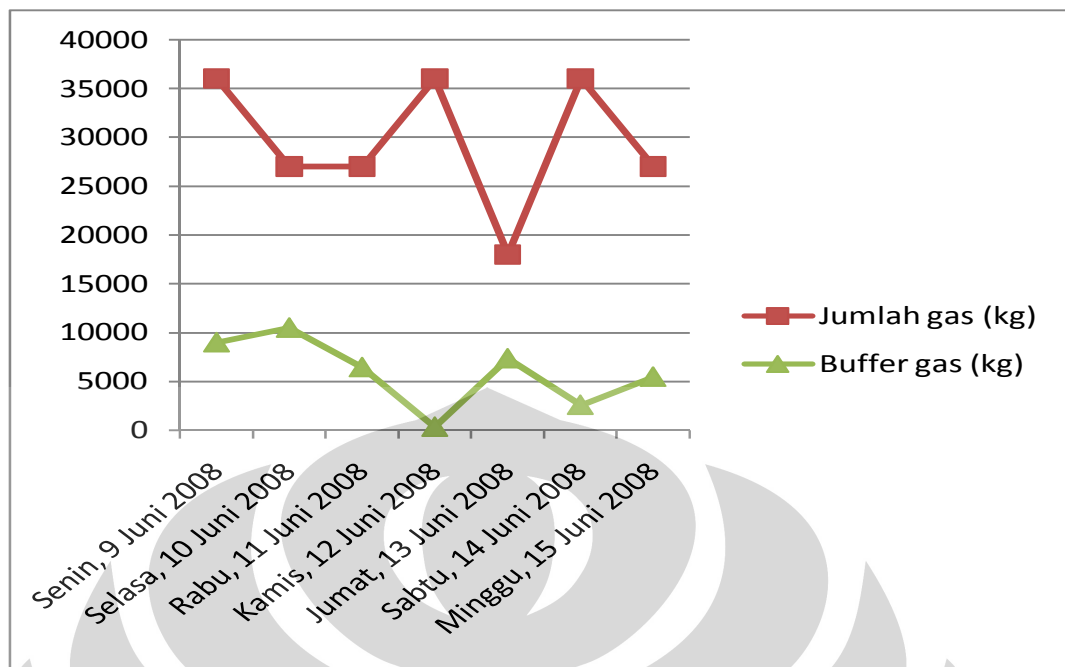
3.4. Data Kuantitatif

3.4.1. Permintaan *Bulk* LPG

Tabel 3.1 dan gambar 3.8 menunjukkan permintaan harian *bulk* elpiji di SPBE selama satu pekan di bulan Juni 2008 untuk diambil dari depo Pertamina. Jumlah hari kerja per bulan di SPBE rata-rata selama 30 hari. Sedangkan dalam satu hari kerja, SPBE beroperasi dari 06.00 WIB hingga 18.00 WIB dengan waktu istirahat pertama (12.00 – 13.00 WIB) dan waktu istirahat kedua (15.00-15.30 WIB).

Tabel 3.1 Permintaan Harian *Bulk* LPG

Permintaan harian <i>bulk</i> LPG dalam satu pekan			
Tanggal	Jumlah skidtank (unit)	Jumlah gas (kg)	Buffer gas (kg)
Senin, 9 Juni 2008	4	36000	9000
Selasa, 10 Juni 2008	3	27000	10500
Rabu, 11 Juni 2008	3	27000	6450
Kamis, 12 Juni 2008	4	36000	330
Jumat, 13 Juni 2008	2	18000	7350
Sabtu, 14 Juni 2008	4	36000	2580
Minggu, 15 Juni 2008	3	27000	5460



Gambar 3.8 Permintaan Harian *Bulk* LPG

3.4.2. Jumlah dan Interval Kedatangan *Skid Tank*

Interval kedatangan pertama setiap harinya dihitung dari dimulainya kegiatan operasional SPBE jam 06.00 WIB sampai *skid tank* pertama datang melapor ke penjaga gerbang. Sedangkan untuk kedatangan *skid tank* yang kedua dan seterusnya pada hari yang sama, interval kedatangan ditentukan dengan menghitung selisih waktu antara *skid tank* ke- (n) dan *skid tank* ke- ($n-1$) melapor kepada penjaga gerbang.

Tabel 3.2 Jadwal Kedatangan *Skid Tank*

Senin, 9 Juni 2008		
No.	Jam Masuk	Interval (Jam)
1	7:00	1
2	8:00	1
3	9:00	1
4	10:00	1
Selasa, 10 Juni 2008		
No.	Jam Masuk	Interval (Jam)
1	6:00	0
2	8:00	2
3	8:00	0
Rabu, 11 Juni 2008		
No.	Jam Masuk	Interval (Jam)
1	7:00	1
2	8:00	1
3	12:00	4
Kamis, 12 Juni 2008		
No.	Jam Masuk	Interval (Jam)
1	7:00	1
2	10:00	3
3	11:00	1
4	11:00	0
Jumat, 13 Juni 2008		
No.	Jam Masuk	Interval (Jam)
1	10:00	4
2	10:00	0
Sabtu, 14 Juni 2008		
No.	Jam Masuk	Interval (Jam)
1	8:00	2
2	11:00	3
3	12:00	1
4	14:00	2
Minggu, 15 Juni 2008		
No.	Jam Masuk	Interval (Jam)
1	9:00	3
2	11:00	2
3	12:00	1

Waktu antar kedatangan *skid tank* terdistribusi *Poisson (1.1) HR* yaitu distribusi *Poisson* dengan interval sebesar 1.1 jam.

3.4.3. Proses Pengisian *Storage Tank*

Definisi proses pengisian *storage tank* ialah dimulai dari pemasangan saringan hingga mesin kompresor dimatikan. Berikut ini ditampilkan hasil pencatatan *time study* untuk proses pengisian *storage tank*.

Tabel 3.3 *Time Study* Proses Pengisian *Storage Tank*

<i>n</i>	<i>t</i> (menit)
1	30
2	31
3	30
4	29
5	29
6	30
7	31
8	29
9	30
10	29
11	31
12	29
13	30
14	31
15	29
16	30
17	31
18	31
19	29
20	30
21	29
22	30
23	31
24	29
25	30
26	29
27	30
28	31
29	29
30	30
<i>N'</i>	0,3

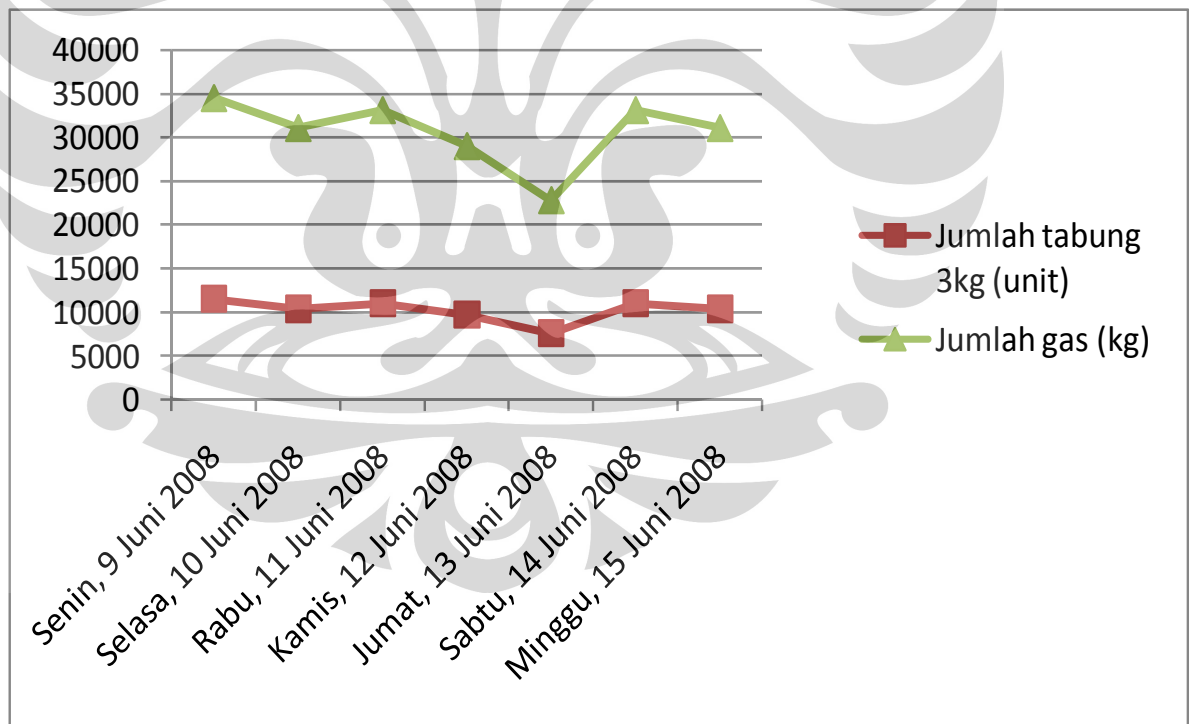
Dari uji kecukupan data didapat jumlah observasi untuk proses pengisian *storage tank* yang harus dilakukan ialah $N' = 3$; sedangkan observasi dengan *time study* telah melebihi jumlah observasi ($N' = 3$) yaitu sebanyak 30. Kemudian dengan aplikasi Stat::Fit, didapat distribusi proses pengisian *storage tank* mengikuti distribusi *Normal* (25., 5.) *MIN* yaitu distribusi Normal dengan nilai rata-rata 25 menit dan standar deviasi 5 menit.

3.4.4. Permintaan Tabung LPG 3 Kg

Tabel 3.4 dan gambar 3.9 menunjukkan daftar permintaan harian tabung KPG 3 Kg di SPBE selama satu pekan di bulan Juni 2008. Jumlah hari kerja per bulan di SPBE rata-rata selama 30 hari. Sedangkan dalam satu hari kerja, SPBE beroperasi dari 06.00 WIB hingga 18.00 WIB dengan waktu istirahat pertama (12.00 – 13.00 WIB) dan waktu istirahat kedua (15.00-15.30 WIB).

Tabel 3.4 Permintaan Harian Tabung LPG 3 Kg

Permintaan harian LPG 3 kg dalam satu pekan			
Tanggal	Jumlah truk agen (unit)	Jumlah tabung 3kg (unit)	Jumlah gas (kg)
Senin, 9 Juni 2008	50	11500	34500
Selasa, 10 Juni 2008	45	10350	31050
Rabu, 11 Juni 2008	48	11040	33120
Kamis, 12 Juni 2008	42	9660	28980
Jumat, 13 Juni 2008	33	7590	22770
Sabtu, 14 Juni 2008	48	11040	33120
Minggu, 15 Juni 2008	45	10350	31050



Gambar 3.9 Permintaan Harian Tabung LPG 3 Kg

3.4.5. Jumlah dan Interval Kedatangan Truk Agen

Interval kedatangan pertama setiap harinya dihitung dari dimulainya kegiatan operasional SPBE jam 06.00 WIB sampai truk pertama datang melapor ke penjaga gerbang. Sedangkan untuk kedatangan truk yang kedua dan seterusnya pada hari yang sama, interval kedatangan ditentukan dengan menghitung selisih waktu antara truk ke- (n) dan truk ke- ($n - 1$) melapor kepada penjaga gerbang.

Tabel 3.5 Interval Kedatangan Tiap Satu Unit Truk Agen

Senin, 9 Juni 2008			Selasa, 10 Juni 2008			Rabu, 11 Juni 2008		
No.	Jam Masuk	Interval (menit)	No.	Jam Masuk	Interval (menit)	No.	Jam Masuk	Interval (menit)
1	6:06	6	1	6:10	10	1	6:08	8
2	6:13	7	2	6:23	13	2	6:22	14
3	6:27	14	3	6:34	11	3	6:31	9
4	6:36	9	4	6:42	8	4	6:39	8
5	6:41	5	5	6:52	10	5	6:49	10
6	6:52	11	6	7:06	14	6	6:56	7
7	7:03	11	7	7:14	8	7	7:08	12
8	7:16	13	8	7:22	8	8	7:20	12
9	7:18	2	9	7:33	11	9	7:30	10
10	7:29	11	10	7:42	9	10	7:37	10
11	7:40	11	11	7:56	14	11	7:44	7
12	7:45	5	12	8:07	11	12	7:56	12
13	7:58	13	13	8:12	5	13	8:06	10
14	8:10	12	14	8:25	13	14	8:14	8
15	8:23	13	15	8:34	9	15	8:17	3
16	8:36	13	16	8:46	12	16	8:33	16
17	8:44	8	17	9:02	16	17	8:48	15
18	8:52	8	18	9:13	11	18	8:51	3
19	9:01	9	19	9:24	11	19	8:58	7
20	9:12	11	20	9:35	11	20	9:07	9
21	9:22	10	21	9:42	8	21	9:11	4
22	9:30	8	22	9:46	4	22	9:21	10
23	9:41	11	23	9:54	8	23	9:29	8
24	9:50	9	24	10:06	12	24	9:38	9
25	9:56	6	25	10:14	8	25	9:54	16
26	10:04	8	26	10:28	14	26	9:59	5
27	10:17	13	27	10:37	9	27	10:06	7
28	10:26	9	28	10:45	8	28	10:14	8
29	10:38	12	29	10:55	10	29	10:23	9
30	10:49	11	30	11:02	7	30	10:32	9
31	10:58	9	31	11:14	12	31	10:39	7
32	11:06	8	32	11:26	12	32	10:49	10
33	11:13	7	33	11:36	10	33	11:00	11
34	11:21	8	34	11:46	10	34	11:18	18
35	11:27	6	35	11:53	7	35	11:25	7
36	11:37	10	36	12:05	12	36	11:32	7
37	11:50	13	37	12:15	10	37	11:37	5
38	12:01	11	38	12:23	8	38	11:44	7
39	12:09	8	39	12:26	3	39	11:53	9
40	12:19	10	40	12:42	16	40	12:02	9
41	12:33	14	41	12:57	15	41	12:13	11
42	12:41	8	42	13:00	3	42	12:22	9
43	12:49	8	43	13:07	7	43	12:34	12
44	13:00	11	44	13:16	9	44	12:50	16
45	13:09	9	45	13:26	10	45	13:01	11
46	13:23	14				46	13:12	11
47	13:34	11				47	13:23	11
48	13:39	5				48	13:30	7
49	13:52	13						
50	14:01	9						

Tabel 3.5 Interval Kedatangan Tiap Satu Unit Truk Agen (sambungan)

Kamis, 12 Juni 2008			Jumat, 13 Juni 2008			Sabtu, 14 Juni 2008			Minggu, 15 Juni 2008		
No.	Jam Masuk	Interval (menit)	No.	Jam Masuk	Interval (menit)	No.	Jam Masuk	Interval (menit)	No.	Jam Masuk	Interval (menit)
1	6:08	8	1	6:11	11	1	6:02	2	1	6:09	9
2	6:16	8	2	6:22	11	2	6:13	11	2	6:14	5
3	6:27	11	3	6:33	11	3	6:24	11	3	6:25	11
4	6:36	9	4	6:41	8	4	6:29	5	4	6:36	11
5	6:50	14	5	6:45	4	5	6:42	13	5	6:49	13
6	7:01	11	6	6:53	8	6	6:54	12	6	6:51	2
7	7:06	5	7	7:05	12	7	7:07	13	7	7:02	11
8	7:19	13	8	7:13	8	8	7:20	13	8	7:13	11
9	7:28	9	9	7:27	14	9	7:28	8	9	7:18	5
10	7:40	12	10	7:36	9	10	7:36	8	10	7:31	13
11	7:56	16	11	7:44	8	11	7:45	9	11	7:42	12
12	8:07	11	12	7:54	10	12	7:56	11	12	7:55	13
13	8:18	11	13	8:01	7	13	8:06	10	13	8:08	13
14	8:29	11	14	8:13	12	14	8:14	8	14	8:16	8
15	8:37	8	15	8:25	12	15	8:25	11	15	8:24	8
16	8:41	4	16	8:35	10	16	8:34	9	16	8:33	9
17	8:49	8	17	8:45	10	17	8:40	6	17	8:44	11
18	8:51	12	18	8:52	7	18	8:48	8	18	8:54	10
19	8:59	8	19	9:04	12	19	9:01	13	19	9:04	8
20	9:13	14	20	9:14	10	20	9:10	9	20	9:15	11
21	9:22	9	21	9:22	8	21	9:22	12	21	9:24	9
22	9:30	8	22	9:25	3	22	9:33	11	22	9:30	6
23	9:40	10	23	9:41	16	23	9:42	9	23	9:38	8
24	9:47	7	24	9:56	15	24	9:50	8	24	9:51	13
25	9:59	12	25	9:59	3	25	9:57	7	25	10:00	9
26	10:11	12	26	10:06	7	26	10:05	8	26	10:12	12
27	10:21	10	27	10:15	9	27	10:11	6	27	10:23	11
28	10:31	10	28	10:19	4	28	10:21	10	28	10:32	9
29	10:38	7	29	10:29	10	29	10:34	13	29	10:40	8
30	10:50	12	30	10:37	8	30	10:45	11	30	10:47	7
31	11:00	10	31	10:46	9	31	10:53	8	31	10:55	8
32	11:08	8	32	11:02	16	32	11:03	10	32	11:01	6
33	11:11	3	33	11:07	5	33	11:17	14	33	11:11	10
34	11:27	16				34	11:25	8	34	11:24	13
35	11:42	15				35	11:33	8	35	11:35	11
36	11:45	3				36	11:44	11	36	11:43	8
37	11:52	7				37	11:53	9	37	11:53	10
38	12:01	9				38	12:07	14	38	12:07	14
39	12:05	4				39	12:18	11	39	12:15	8
40	12:15	10				40	12:23	5	40	12:23	8
41	12:23	8				41	12:36	13	41	12:34	11
42	12:32	9				42	12:45	9	42	12:43	9
						43	12:57	12	43	12:57	14
						44	13:13	16	44	13:08	11
						45	13:24	11	45	13:13	5
						46	13:35	11			
						47	13:46	11			
						48	13:54	8			

Waktu antar kedatangan truk agen terdistribusi *Poisson (9.5) MIN* yaitu distribusi Poisson dengan interval sebesar 9.5 menit.

3.4.6. Proses Bongkar Tabung

Proses *unloading* atau pembongkaran tabung kosong didefinisikan sebagai pemindahan tabung dari truk agen ke area pengisian. Yang dimaksud dengan area pengisian adalah *chain conveyor* titik masuknya tabung ke sistem *carrousel filling machines*. Dalam satu kali pembongkaran, tabung yang dipindahkan jumlahnya dua buah dan dilakukan oleh satu orang operator. Waktu pencatatan dimulai dari operator memegang *hand guard* (bagian atas) tabung hingga tabung dipindahkan ke area pengisian.

Tabel 3.6 *Time Study* Proses Bongkar Tabung

<i>n</i>	<i>t</i> (menit)
1	10
2	8
3	9
4	11
5	10
6	11
7	8
8	8
9	10
10	9
11	9
12	10
13	8
14	11
15	10
16	9
17	8
18	10
19	9
20	10
21	9
22	10
23	8
24	11
25	11
26	10
27	8
28	9
29	10
30	11
<i>N'</i>	5

Dari uji kecukupan data didapat jumlah observasi untuk proses pembongkaran tabung kosong yang harus dilakukan ialah $N' = 5$; sedangkan observasi dengan *time study* telah melebihi jumlah observasi ($N' = 5$) yaitu sebanyak 30. Kemudian dengan aplikasi Stat::Fit, didapat distribusi proses pembongkaran tabung kosong mengikuti distribusi *Triangular* (3.83, 5.65, 9.10) MIN yaitu distribusi Triangular

dengan waktu paling minimal 3.83 menit, waktu yang paling sering terjadi 5.65 menit, dan waktu paling maksimal 9.10 menit.

3.4.7. Proses Penimbangan Tabung

Tabung ditimbang ditandai dengan naiknya badan tabung beberapa cm lebih tinggi dibanding tabung-tabung lain di *conveyor*. Waktu pencatatan dengan metode *time study* untuk proses penimbangan tabung dimulai dari naiknya ketinggian tabung hingga tabung turun sebanding dengan tinggi *conveyor*.

Tabel 3.7 *Time Study* Proses Penimbangan Tabung

<i>n</i>	<i>t</i> (detik)
1	1,1
2	0,9
3	1
4	1,1
5	0,8
6	1
7	1,2
8	0,9
9	1,1
10	0,8
11	1
12	1,1
13	0,9
14	0,8
15	0,9
16	1
17	0,8
18	1,1
19	1,2
20	1
21	0,9
22	0,8
23	1
24	1,1
25	1
26	0,9
27	1
28	0,8
29	0,8
30	1
<i>N'</i>	6

Dari uji kecukupan data didapat jumlah observasi untuk proses penimbangan tabung yang harus dilakukan ialah $N' = 6$; sedangkan observasi dengan *time study* telah melebihi jumlah observasi ($N' = 6$) yaitu sebanyak 30. Kemudian dengan

aplikasi Stat::Fit, didapat distribusi proses penimbangan tabung mengikuti distribusi *Uniform (0.8, 1.2) SEC* yaitu distribusi Uniform dengan waktu paling minimal 0.8 detik dan waktu paling maksimal 1.2 detik.

3.4.8. Proses Pengisian Tabung

Yang dimaksud dengan proses pengisian tabung dengan mesin *carrousel* ini adalah ketika operator memasang waktu putar mesin *carrousel*. Seiring dengan berputarnya *carrousel*, gas elpiji mengalir masuk ke dalam tabung. Proses pengisian dihentikan saat selang secara otomatis terlepas dari mulut tabung yang menandakan bahwa tabung telah terisi penuh dengan berat total tabung dan isi. Pada saat observasi dilakukan, waktu putar mesin *carrousel* ialah 30 detik untuk satu kali putaran. Oleh sebab satu set mesin *carrousel* terdiri atas 12 unit mesin tunggal maka waktu pengisian satu tabung ialah 30 detik dibagi 12 tabung yaitu 2.5 detik per tabung. Karena waktu putar tersebut dipasang tetap, maka waktu putar cenderung konstan.

3.4.9. Proses Pemeriksaan Isi Tabung

Dalam perhitungan *time study*, waktu pemeriksaan ketepatan isi tabung adalah saat tabung ditimbang ditandai dengan naiknya badan tabung beberapa cm lebih tinggi dibanding tabung-tabung lain di *conveyor*.

Tabel 3.8 *Time Study* Proses Pemeriksaan Isi Tabung

<i>n</i>	<i>t</i> (detik)
1	0,8
2	0,9
3	0,9
4	0,9
5	1
6	1,1
7	1
8	0,9
9	1,1
10	1,1
11	1
12	0,8
13	0,8
14	0,8
15	0,9
16	1,1
17	1
18	1
19	1,1
20	1
21	1
22	0,9
23	1
24	1
25	0,9
26	1,1
27	1
28	1
29	0,8
30	1
<i>N'</i>	4

Dari uji kecukupan data didapat jumlah observasi untuk proses pemeriksaan isi tabung yang harus dilakukan ialah $N' = 4$; sedangkan observasi dengan *time study* telah melebihi jumlah observasi ($N' = 4$) yaitu sebanyak 30. Kemudian dengan aplikasi Stat::Fit, didapat distribusi proses pemeriksaan isi tabung mengikuti distribusi *Triangular* (0.732, 1., 1.16) SEC yaitu distribusi Triangular dengan waktu paling minimal 0.732 detik, waktu yang paling sering terjadi 1 detik, dan waktu paling maksimal 1.16 detik.

3.4.10. Proses Koreksi Isi Tabung

Pencatatan waktu pengkoreksian isi tabung ditentukan mulai dari operator memasang selang ke mulut tabung hingga selang secara otomatis lepas dari tabung. Waktu tunggu tabung hingga dikoreksi yaitu saat operator pengisian

datang ke stasiun koreksi dan memulai proses pengkoreksian isi tabung, tidak dimasukkan dalam pencatatan.

Tabel 3.9 *Time Study* Proses Koreksi Isi Tabung

<i>n</i>	<i>t</i> (detik)
1	1,2
2	1,3
3	1
4	1,2
5	1,3
6	1,4
7	1,5
8	1,5
9	1,2
10	1,3
11	1,2
12	1,3
13	1
14	1
15	1
16	1,2
17	1,4
18	1,3
19	1,2
20	1,4
21	1,3
22	1
23	1
24	1,2
25	1,3
26	1,1
27	1
28	1,1
29	1,1
30	1
<i>N'</i>	7

Dari uji kecukupan data didapat jumlah observasi untuk proses pengkoreksian isi tabung yang harus dilakukan ialah $N' = 7$; sedangkan observasi dengan *time study* telah melebihi jumlah observasi ($N' = 7$) yaitu sebanyak 30. Kemudian dengan aplikasi Stat::Fit, didapat distribusi proses pengkoreksian isi tabung mengikuti distribusi *Lognormal* (0.286, 0.171) *SEC* yaitu distribusi Lognormal dengan waktu paling minimal 0.286 detik dengan deviasi 0.171 detik.

3.4.11. Proses Uji Kebocoran Tabung

Waktu yang dicatat dalam pengambilan data *time study* untuk tabung 3 kg dimulai saat operator memegang tabung kemudian tabung dikeluarkan untuk dibariskan di area muat.

Tabel 3.10 *Time Study* Proses Uji Kebocoran Tabung

<i>n</i>	<i>t</i> (detik)
1	1,5
2	1,6
3	2
4	1,4
5	1,7
6	1,8
7	2
8	1,8
9	1,5
10	1,7
11	1,8
12	1,8
13	1,7
14	1,6
15	1,5
16	1,8
17	1,9
18	1,7
19	1,8
20	1,9
21	1,8
22	1,7
23	1,8
24	1,5
25	1,6
26	1,5
27	1,6
28	1,8
29	1,7
30	1,8
<i>N'</i>	3

Dari uji kecukupan data didapat jumlah observasi untuk proses uji kebocoran tabung yang harus dilakukan ialah $N' = 3$; sedangkan observasi dengan *time study* telah melebihi jumlah observasi ($N' = 3$) yaitu sebanyak 30. Kemudian dengan aplikasi Stat::Fit, didapat distribusi proses uji kebocoran tabung mengikuti distribusi *Normal* (1.71, 0.151) *SEC* yaitu distribusi Normal dengan waktu rata-rata 1.71 detik dengan deviasi 0.151 detik.

3.4.12. Proses Segel Tutup Tabung

Proses pemasangan pengaman tabung biasanya melibatkan dua orang operator, operator pertama akan memasang tutup plastik ke tabung kemudian operator kedua mengencangkan tutup dengan memukulnya menggunakan palu. Pencatatan waktu *time study* dimulai dari operator pertama mengambil tutup plastik hingga operator kedua selesai mengencangkannya.

Tabel 3.11 *Time Study* Proses Segel Tutup Tabung

<i>n</i>	<i>t</i> (detik)
1	0,5
2	0,5
3	0,4
4	0,5
5	0,4
6	0,3
7	0,5
8	0,3
9	0,3
10	0,3
11	0,5
12	0,3
13	0,5
14	0,4
15	0,4
16	0,3
17	0,5
18	0,4
19	0,3
20	0,5
21	0,4
22	0,5
23	0,3
24	0,5
25	0,4
26	0,3
27	0,6
28	0,5
29	0,4
30	0,4
<i>N'</i>	18

Dari uji kecukupan data didapat jumlah observasi untuk proses pemasangan pengaman tabung yang harus dilakukan ialah $N' = 18$; sedangkan observasi dengan *time study* telah melebihi jumlah observasi ($N' = 18$) yaitu sebanyak 30. Kemudian dengan aplikasi Stat::Fit, didapat distribusi proses uji kebocoran tabung

mengikuti distribusi *Normal* (0.413, 0.008) *SEC* yaitu distribusi Normal dengan waktu rata-rata 0.413 detik dengan deviasi 0.008 detik

3.4.13. Proses Muat Tabung

Proses *loading* atau pemuatan tabung isi elpiji pada dasarnya adalah kebalikan dari proses *unloading* yaitu pemindahan tabung dari area pengisian ke truk agen. Pengangkutan tabung dilakukan oleh satu orang operator. Satu kali pemuatan, operator membawa 2 buah tabung. Waktu proses yang dicatat dalam *time study* dimulai dari operator memegang *hand guard* salah satu tabung, membawa kemudian meletakkannya di dalam truk agen.

Tabel 3.12 *Time Study* Proses Muat Tabung

<i>n</i>	<i>t</i> (menit)
1	11
2	10
3	9
4	10
5	10
6	9
7	10
8	11
9	9
10	11
11	12
12	11
13	10
14	9
15	11
16	9
17	10
18	10
19	12
20	11
21	10
22	10
23	9
24	11
25	9
26	10
27	11
28	10
29	10
30	10
<i>N'</i>	3

Dari uji kecukupan data didapat jumlah observasi untuk proses pemuatan tabung isi yang harus dilakukan ialah $N' = 3$; sedangkan observasi dengan *time study* telah melebihi jumlah observasi ($N' = 3$) yaitu sebanyak 30. Kemudian dengan aplikasi Stat::Fit, didapat distribusi proses pemuatan tabung isi mengikuti distribusi *Triangular (7., 10.1, 11.7) MIN* yaitu distribusi Triangular dengan waktu paling minimal 7 menit, waktu yang paling sering terjadi 10.1 menit, dan waktu paling maksimal 11.7 menit.

3.4.14. Proses Evakuasi Isi Tabung

Yang dimaksud dengan proses evakuasi adalah pemindahan kembali gas elpiji yang ada dalam tabung ke tangki penyimpanan. Penyebab tabung harus dievakuasi adalah jika ditemukan kebocoran pada tabung baik yang diketahui saat dalam proses pengisian di SPBE atau pun oleh konsumen.

Tabel 3.13 *Time Study* Proses Evakuasi Isi Tabung

<i>n</i>	<i>t</i> (detik)
1	1
2	1,2
3	1,4
4	1
5	1
6	1,2
7	1,4
8	1,2
9	1,3
10	1,4
11	1,4
12	1,3
13	1,4
14	1,3
15	1,2
16	1,4
17	1,5
18	1,4
19	1,2
20	1,3
21	1,2
22	1,4
23	1,2
24	1,4
25	1,4
26	1,3
27	1,2
28	1,2
29	1,4
30	1,3
<i>N'</i>	4

Dari uji kecukupan data didapat jumlah observasi untuk proses evakuasi isi tabung yang harus dilakukan ialah $N' = 4$; sedangkan observasi dengan *time study* telah melebihi jumlah observasi ($N' = 4$) yaitu sebanyak 30. Kemudian dengan aplikasi Stat::Fit, didapat distribusi proses evakuasi isi tabung mengikuti distribusi *Triangular* (0.921, 1.4, 1.52) SEC yaitu distribusi Triangular dengan waktu paling minimal 0.921 detik, waktu yang paling sering terjadi 1.4 detik, dan waktu paling maksimal 1.52 detik.



4. PEMODELAN DAN EKSPERIMEN

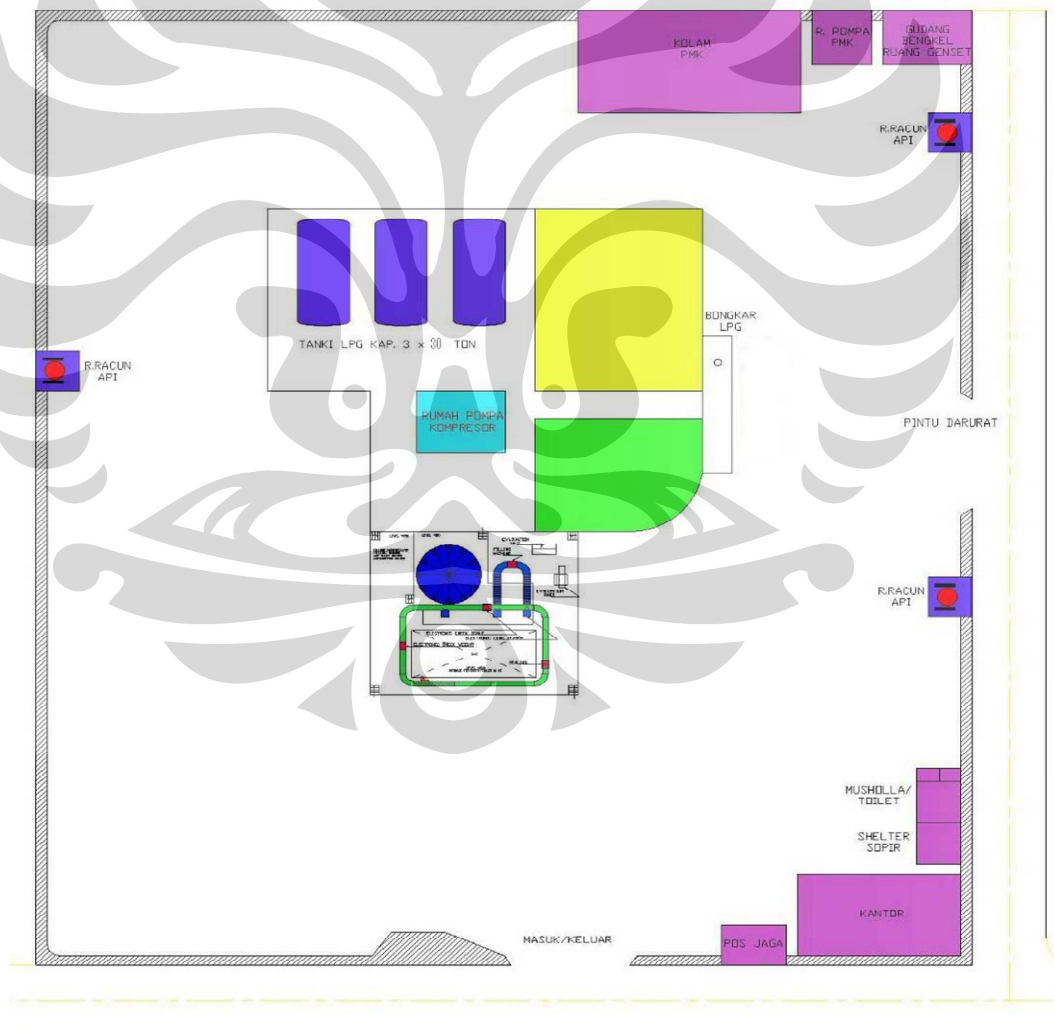
4.1. Formulasi Model

4.1.1. Elemen Struktural

Elemen struktural dalam sebuah model simulasi terdiri atas *Layout*, entitas, lokasi, sumberdaya, dan jalur.

a. *Layout*

Layout yang digunakan merupakan *Layout* SPBE dari Pertamina yang dibuat dengan *software* AutoCAD sehingga sangat memperhatikan ukuran yang sebenarnya dengan skala 1:500 dengan satuan mm (milimeter). *Layout* ini merupakan tempat berlangsungnya seluruh aktivitas operasional SPBE. Gambar 4.1 berikut merupakan *layout* yang digunakan dalam model.



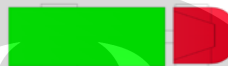
Gambar 4.1 *Layout* SPBE yang digunakan pada model simulasi

b. Entitas (*Entity*)

Entitas adalah sesuatu yang diproses dalam suatu sistem. Ada tiga jenis entitas yang terlibat dalam model simulasi SPBE ini yaitu *skid tank* (truk tangki), truk agen (truk pengangkut tabung 3 kg), dan tabung gas 3 kg. Ketiga entitas itu memiliki identitas berbeda sehingga secara grafis dibedakan sebagai berikut:



Gambar 4.2 *Skid tank* (Truk Tangki)



Gambar 4.3 Truk Agen (Pengangkut Tabung 3 kg Kosong)



Gambar 4.4 Truk Agen (Pengangkut Tabung 3 kg Isi)



Gambar 4.5 Tabung Gas 3 kg

Secara grafis, keempat entitas digambarkan jika dilihat dari atas. Khusus untuk *skid tank* dan truk agen, arah kepala kedua jenis entitas tersebut saat simulasi berjalan akan berubah mengikuti arah pergerakan entitas tersebut. Kecepatan untuk *skid tank* dan truk agen dalam simulasi ialah 50 km/jam sesuai standar kecepatan kendaraan yang aman. Sedangkan kecepatan tabung 3 kg sesuai kecepatan *conveyor* dan perpindahan operator.

c. Lokasi (*Location*)

Lokasi ialah tempat dimana entitas diproses atau berhenti. Secara total, model ini memiliki sekitar 71 lokasi. Karena model ini mensimulasikan kombinasi antara sistem diskrit dan kontinyu, maka 71 lokasi tersebut sudah mencakup keduanya. Setiap lokasi memiliki posisi, nama, kapasitas, jumlah, dan aturan spesifik masing-masing. Khusus untuk sistem kontinyu, lokasi yang digunakan berupa *tank* (tangki) dan *pipe* (pipa).

d. Sumberdaya (*Resources*)

Sumberdaya ialah pihak yang digunakan untuk memproses entitas. Secara total, terdapat lima jenis sumberdaya yang digunakan. Setiap sumberdaya

memiliki posisi, nama, jumlah dan aturan spesifik masing-masing. Operator dan mesin termasuk dalam kategori sumberdaya.

e. Jalur (*Paths*)

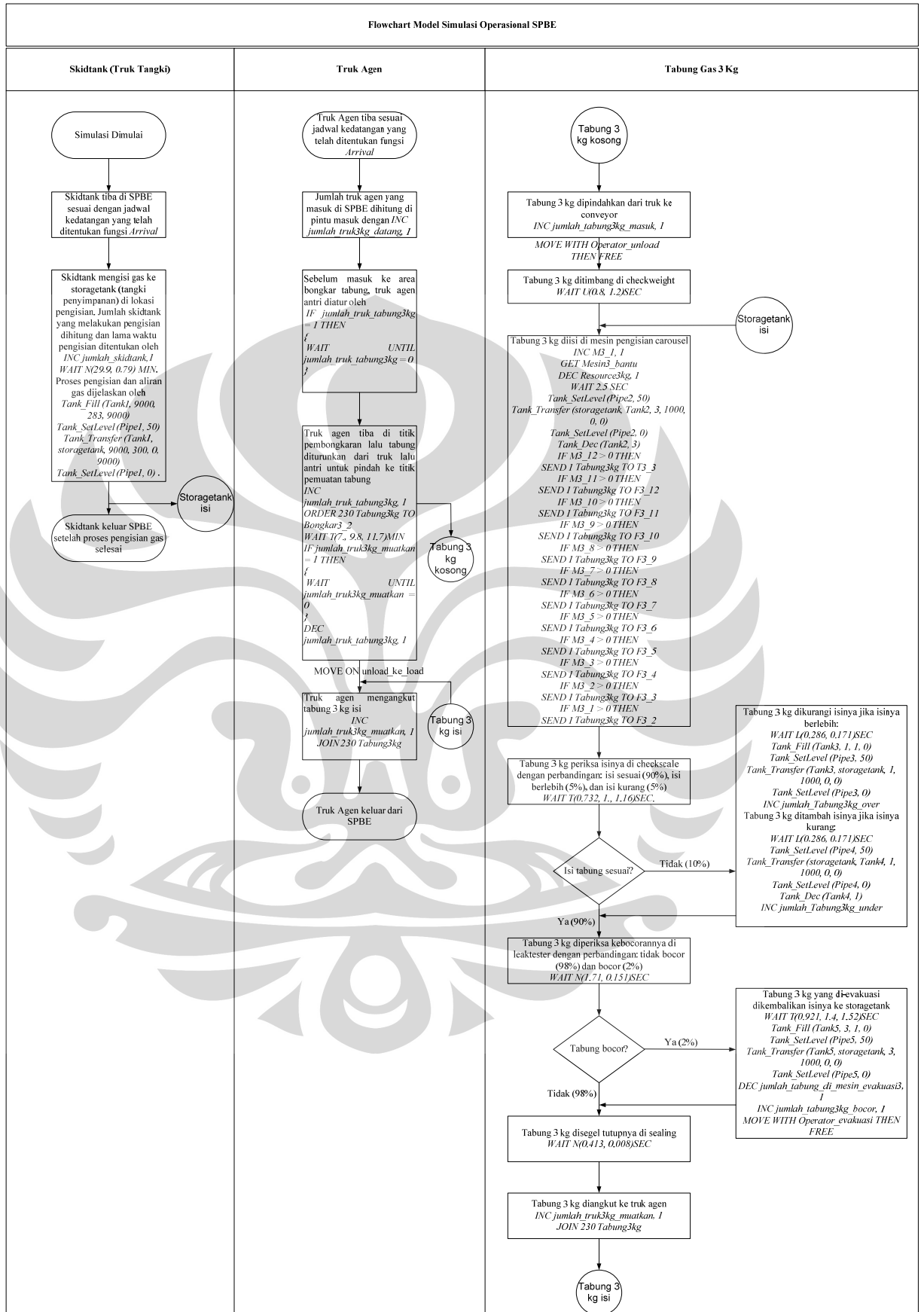
Jalur ialah jalan yang dilalui oleh entitas dan sumberdaya dalam sistem. Secara total, terdapat 28 jenis jalur dalam model simulasi ini. Setiap jalur memiliki posisi, nama, tipe, jarak, kecepatan entitas dan/atau sumberdaya yang melaluinya, dan aturan spesifik masing-masing. Karena *Layout* yang digunakan menggambarkan ukuran yang sebenarnya dengan skala, maka jarak dalam model ini mendekati jarak yang sesungguhnya.

4.1.2. Elemen Operasional

Elemen operasional menggambarkan aktivitas dan perilaku setiap elemen dalam sistem beserta interaksi yang terjadi antar elemen dalam sistem. Elemen operasional terdiri dari *routing*, *task selection rule*, operasi, kedatangan entitas dan sumberdaya, pergerakan entitas dan sumberdaya, *shift* lokasi, dan variabel lainnya.

a. Diagram alir model sistem SPBE

Beberapa elemen operasional seperti *routing*, *task selection rule*, operasi, kedatangan entitas dan sumberdaya, pergerakan entitas dan sumberdaya dapat sekaligus dijelaskan pada diagram alir model sistem SPBE. Diagram alir model berikut ini dapat menjelaskan secara ringkas aktivitas operasional dalam sistem SPBE.



Gambar 4.6 Diagram Alir Operasi SPBE

Dalam diagram alir operasi SPBE tersebut, terdapat tiga jenis entitas yang terlibat dalam sistem dan berinteraksi satu sama lain.

1) *Skid tank* (truk tangki)

Skid tank tiba di SPBE sesuai dengan fungsi *Arrival*. Fungsi ini mendefinisikan kedatangan entitas dalam sistem. Fungsi ini mengandung informasi jumlah entitas yang datang tiap satu kali kedatangan, frekuensi kedatangan, lokasi kedatangan, waktu kedatangan pertama kali, dan jumlah total entitas yang datang. *Skid tank* yang datang akan menuju lokasi pengisian gas. Di lokasi ini, dibuatlah kalimat *INC jumlah_skid tank, 1* dengan tujuan menghitung jumlah *skid tank* yang melakukan pengisian. Fungsi *INC* dibuat untuk menghitung suatu *variabel*, *array*, dan *attribute* dengan nilai kelipatan tertentu. Kemudian lamanya waktu pengisian digambarkan oleh kalimat *WAIT N(29.9, 0.79) MIN*. Fungsi *WAIT* dibuat untuk mensimulasikan lamanya waktu untuk memproses entitas di suatu lokasi. Sedangkan satuan waktu yang digunakan di sini ialah *N(29.9, 0.79) MIN* yang berarti waktu proses pengisian terdistribusi *Normal* dalam satuan menit (dengan: *min* = 29.9; dan *deviasi* = 0.79). Proses pengisian dan aliran gas dari *skid tank* ke *storage tank* digambarkan oleh kalimat

Tank_Fill (Tank1, 9000, 300, 9000)

Tank_SetLevel (Pipe1, 50)

Tank_Transfer (Tank1, storage tank, 9000, 300, 0, 9000)

Tank_SetLevel (Pipe1, 0)

Fungsi *Tank_Fill* digunakan untuk mentransfer fluida dari sumber yang bukan tangki atau bukan bagian dari sistem dengan mendefinisikan tangki tujuan, jumlah fluida yang ditransfer, kecepatan aliran fluida, dan level *resume*. Fungsi *Tank_SetLevel* digunakan untuk memasang jumlah fluida dalam tangki dengan mendefinisikan nama tangki dan jumlah fluida dalam tangki. Fungsi *Tank_Transfer* digunakan untuk mentransfer fluida antara sesama tangki dengan mendefinisikan tangki sumber fluida, tangki tujuan transfer fluida, jumlah fluida yang ditransfer, kecepatan aliran fluida dari tangki sumber fluida, kecepatan aliran fluida dari tangki tujuan transfer fluida, dan level *resume*. Setelah semua kalimat di atas dijalankan, maka *storage tank* akan

terisi dengan fluida sejumlah tertentu. Pada model simulasi ini, karena *skid tank* membawa gas sebanyak 9000 kg maka setiap kali transfer akan menambah gas sebanyak 9000 kg dalam *storage tank*. *Storage tank* yang telah terisi ini akan digunakan untuk mengisi tabung 3 kg. Setelah melakukan pengisian, *skid tank* akan keluar dari SPBE menurut jalur yang telah didefinisikan.

2) Truk Agen (pengangkut tabung 3 kg)

Truk agen datang ke SPBE menurut fungsi *Arrival* yang telah dijelaskan di bagian pembahasan tentang *skid tank*. Setiap kali memasuki SPBE, maka jumlah truk agen dihitung dengan fungsi *INC* (telah dijelaskan pada pembahasan tentang *skid tank*). Sebelum memasuki area bongkar tabung, truk agen akan antri dan diatur oleh fungsi

```
IF jumlah_truk_tabung3kg = 1 THEN
{
  WAIT UNTIL jumlah_truk_tabung3kg = 0
}
```

Fungsi ini artinya ialah jika ada truk agen yang sedang membongkar tabung (ditandai dengan nilai = 1) maka tunggu hingga truk agen pergi dari area bongkar tabung (ditandai dengan nilai = 0). Di area bongkar tabung, dikeluarkanlah sebanyak 230 buah tabung gas 3 kg dengan kalimat *ORDER 230 Tabung3kg TO Bongkar3_2*. Fungsi *ORDER* digunakan untuk memunculkan sejumlah entitas di lokasi tertentu. Lamanya waktu bongkar tabung digambarkan oleh kalimat *WAIT T(7., 9.8, 11.7)MIN* yang berarti waktu proses pengisian terdistribusi *Triangular* dalam satuan menit (dengan: *min* = 7.00; *mode* = 9.8; dan *max* = 11.7). Di area bongkar tabung ini, dihasilkanlah tabung 3 kg kosong yang nantinya akan melalui serangkaian proses pengisian hingga dihasilkan tabung 3 kg isi. Truk agen yang berada di area bongkar tabung harus antri hingga bisa pindah ke area angkut tabung. Proses perpindahan truk agen dari area bongkar tabung ke area angkut tabung disimulasikan dengan kalimat *MOVE ON unload_ke_load* yang artinya truk agen berpindah dengan melalui jalur *unload_ke_load*. Fungsi *MOVE ON* menjelaskan berpindahnya entitas atau sumberdaya dengan melalui jalur

tertentu yang telah didefinisikan. Fungsi *MOVE* mensimulasikan perpindahan entitas atau sumberdaya. Di area muat tabung ini, truk agen memuatkan sebanyak 230 buah tabung 3 kg isi yang disimulasikan dengan kalimat *JOIN 230 Tabung3kg* dimana fungsi *JOIN* dibuat untuk menggabungkan beberapa jenis entitas yang berbeda. Setelah proses pemuatan tabung 3 kg isi selesai, truk agen pergi meninggalkan SPBE menurut jalur tertentu yang telah didefinisikan.

3) Tabung gas 3 kg

Tabung 3 kg kosong dipindahkan dari area bongkar tabung ke *conveyor*. Jumlah tabung 3 kg kosong yang masuk dihitung dengan fungsi *INC jumlah_tabung3kg_masuk, 1*. Sedangkan proses pemindahan tabung 3 kg kosong dari area bongkar tabung ke *conveyor* dengan bantuan operator disimulasikan dengan fungsi *MOVE WITH Operator_unload THEN FREE* yang berarti dipindahkan oleh operator kemudian tabung ditaruh di *conveyor*. Fungsi *MOVE WITH* merupakan variasi dari fungsi *MOVE* yang berarti perpindahan entitas oleh sumberdaya tertentu. *Conveyor* membawa tabung 3 kg kosong ke *checkweight* untuk mengetahui berat awal tabung 3 kg kosong tersebut. Lamanya waktu *checkweight* disimulasikan oleh fungsi *WAIT U(0.8, 1.2)SEC* yang berarti waktu proses pengisian terdistribusi *Uniform* dalam satuan detik (dengan: *min* = 0.8; dan *max* = 1.2). Setelah melewati *checkweight*, tabung 3 kg kosong diisi di mesin *carousel* berputar yang terdiri dari 12 unit. Pengisian tersebut disimulasikan dengan kalimat berikut:

```
INC M3_1, 1
```

```
GET Mesin3_bantu
```

```
DEC Resource3kg, 1
```

```
WAIT 2.5 SEC
```

```
Tank_SetLevel (Pipe2, 50)
```

```
Tank_Transfer (storage tank, Tank2, 3, 1000, 0, 0)
```

```
Tank_SetLevel (Pipe2, 0)
```

```
Tank_Dec (Tank2, 3)
```

```
IF M3_12 > 0 THEN
```

```
SEND 1 Tabung3kg TO T3_3
```

```
IF M3_11 > 0 THEN
SEND 1 Tabung3kg TO F3_12
IF M3_10 > 0 THEN
SEND 1 Tabung3kg TO F3_11
IF M3_9 > 0 THEN
SEND 1 Tabung3kg TO F3_10
IF M3_8 > 0 THEN
SEND 1 Tabung3kg TO F3_9
IF M3_7 > 0 THEN
SEND 1 Tabung3kg TO F3_8
IF M3_6 > 0 THEN
SEND 1 Tabung3kg TO F3_7
IF M3_5 > 0 THEN
SEND 1 Tabung3kg TO F3_6
IF M3_4 > 0 THEN
SEND 1 Tabung3kg TO F3_5
IF M3_3 > 0 THEN
SEND 1 Tabung3kg TO F3_4
IF M3_2 > 0 THEN
SEND 1 Tabung3kg TO F3_3
IF M3_1 > 0 THEN
SEND 1 Tabung3kg TO F3_2
```

Fungsi *GET* digunakan untuk menugasi sumberdaya untuk memproses entitas. Proses pengisian tabung berlangsung selama waktu yang disimulasikan dengan fungsi *WAIT 2.5 SEC* yang artinya waktu proses pengisian berlangsung selama 2.5 detik. Angka 2.5 detik diperoleh dari hasil pembagian antara 30 detik dengan 12 mesin di *carousel*. Kenapa demikian? Karena waktu putar *carousel* bisa dipasang, maka jika dipasang waktu putar 30 detik untuk sekali putar maka lama pengisian untuk satu tabung ialah waktu putar dibagi dengan jumlah mesin di *carousel*. Gabungan fungsi *Tank_SetLevel*, *Tank_Transfer*, dan *Tank_Dec* mensimulasikan proses perpindahan gas dari *storage tank* ke mesin *carousel* serta dari mesin *carousel* ke tabung 3 kg

kosong. $M3_1$ hingga $M3_12$ merupakan variabel yang menunjukkan status mesin yang berjumlah 12 tersebut (status = 1 berarti ada tabung yang sedang diproses dan status = 0). Digunakannya fungsi *SEND* untuk menggeser tabung 3 kg dari satu mesin ke mesin berikutnya hingga masuk ke *conveyor*. Dari mesin *carousel*, tabung 3 kg isi diperiksa beratnya di *checkscale* apakah sesuai standar (90%), kelebihan isi (5%), atau kekurangan isi (5%). Jika berat tabung sesuai standar, maka tabung akan menuju *leaktester* untuk diperiksa apakah tabung bocor atau tidak. Tetapi jika berat tabung tidak sesuai standar, maka akan bergerak menuju *correctionscale* untuk penyesuaian isi tabung hingga memenuhi standar. Di *checkscale*, proses penyesuaian berlangsung selama waktu yang disimulasikan dengan fungsi $WAIT T(0.732, 1., 1.16)SEC$ yang artinya waktu proses pengisian terdistribusi *Triangular* dalam satuan detik (dengan: $min = 0.732$; $mode = 1.0$ dan $max = 1.16$). Di *correctionscale*, proses penyesuaian berlangsung selama waktu yang disimulasikan dengan fungsi $WAIT L(0.286, 0.171)SEC$ yang artinya waktu proses pengisian terdistribusi *Lognormal* dalam satuan detik (dengan: $mean = 0.286$; dan $standar deviation = 0.171$). Jika kelebihan isi maka akan dikembalikan ke *storage tank*; dengan disimulasikan oleh fungsi berikut:

$Tank_Fill (Tank3, 1, 1000, 0)$

$Tank_SetLevel (Pipe3, 50)$

$Tank_Transfer (Tank3, storage\ tank, 1, 1000, 0, 0)$

$Tank_SetLevel (Pipe3, 0)$

Sedangkan jika isinya kurang, maka akan diisi langsung dari *storage tank*; dengan disimulasikan oleh fungsi berikut:

$Tank_SetLevel (Pipe4, 50)$

$Tank_Transfer (storage\ tank, Tank4, 1, 1000, 0, 0)$

$Tank_SetLevel (Pipe4, 0)$

$Tank_Dec (Tank4, 1)$

Tabung yang telah disesuaikan isinya di *correctionscale* akan bergerak menuju *leaktester*. Di *leaktester* ini, akan dilakukan proses pemeriksaan kebocoran tabung yang lamanya disimulasikan dengan $WAIT N(1.71, 0.151)SEC$ yang artinya waktu proses pemeriksaan terdistribusi *Normal* dalam

satuan detik (dengan: $min = 1.71$; dan $deviasi = 0.151$). Jika tidak bocor (98%), tabung akan bergerak menuju *sealing*. Tetapi jika bocor (2%), tabung akan bergerak menuju evakuasi. Di mesin evakuasi, isi tabung bocor akan dikembalikan ke *storage tank*. Lamanya proses evakuasi disimulasikan dengan $WAIT T(0.921, 1.4, 1.52)SEC$ yang berarti waktu proses pengisian terdistribusi *Triangular* dalam satuan detik (dengan: $min = 0.921$; $mode = 1.44$; dan $max = 1.52$). Sedangkan proses pengembalian isi gas dari tabung yang bocor ke *storage tank* disimulasikan dengan fungsi

$Tank_Fill(Tank5, 3, 1000, 0)$

$Tank_SetLevel(Pipe5, 50)$

$Tank_Transfer(Tank5, storage\ tank, 3, 1000, 0, 0)$

$Tank_SetLevel(Pipe5, 0)$

Setelah di-evakuasi maka tabung yang bocor ditumpuk di area tabung bocor. Sedangkan tabung yang tidak bocor akan disegel tutupnya di *sealing* dimana waktu proses *sealing* disimulasikan dengan $WAIT N(0.413, 0.008)SEC$ yang artinya waktu proses pengisian terdistribusi *Normal* dalam satuan detik (dengan: $mean = 0.413$; dan $deviasi = 0.008$). Tabung 3 kg isi yang telah disegel tutupnya akan dikeluarkan dari *conveyor* untuk dimuatkan ke atas truk agen.

b. *Shift* lokasi SPBE

Waktu kerja (operasional) di SPBE ialah jam 06.00-18.00 dengan dua kali waktu istirahat yaitu jam 12.00-13.00 dan jam 14.45-15.15. Untuk mensimulasikannya digunakanlah fungsi *shift* pada ProModel.

c. Variabel pada model

Secara total, terdapat 25 variabel yang dibuat dalam model simulasi ini. Akan tetapi sebenarnya terdapat beberapa jenis variabel sebagai berikut:

1. Variabel $jumlah_truk3kg_datang$. Variabel ini menghitung jumlah truk agen yang datang ke SPBE sehingga variabel ini ditempatkan di dekat pintu masuk SPBE. Nilai variabel ini bersifat kumulatif sehingga akan terus bertambah. Penentuan sifat kumulatif pada variabel ini dilakukan dengan bantuan fungsi *INC* untuk menaikkan nilai variabel.

2. Variabel *jumlah_truk_tabung3kg*. Variabel ini menunjukkan status apakah ada truk agen yang sedang melakukan pembongkaran tabung 3 kg kosong. Oleh karenanya, variabel ini ditempatkan di area bongkar tabung. Status variabel ini ditunjukkan dengan angka 1 atau 0 dimana jika status = 1 maka berarti ada truk agen di area bongkar tabung; sedangkan status = 0 maka berarti tidak ada truk agen di area bongkar tabung. Penentuan status variabel ini dibantu dengan fungsi *INC* untuk menaikkan nilai variabel dan dengan fungsi *DEC* untuk menurunkan nilai variabel. Maksudnya ialah saat truk agen berada di area bongkar tabung maka fungsi *INC jumlah_truk_tabung3kg, 1* membuat status variabel menjadi 1. Sedangkan jika truk agen sudah meninggalkan area bongkar tabung, maka fungsi *DEC jumlah_truk_tabung3kg, 1* membuat status variabel menjadi 0. Penentuan status variabel ini berguna untuk membuat antrian pembongkaran tabung. Untuk membuat antrian dengan bantuan status variabel ini, maka di titik lokasi sebelum area bongkar tabung yaitu di lokasi *Masuk_3kg* dibuatlah fungsi logika *IF..THEN..* dengan memanfaatkan status variabel ini. Simulasi antrian ini dibuat dengan kalimat

```
IF jumlah_truk_tabung3kg = 1 THEN  
{  
  WAIT UNTIL jumlah_truk_tabung3kg = 0  
}
```

Kalimat di atas berarti jika ada truk agen di area bongkar tabung maka tunggu hingga tidak ada truk agen di area bongkar tabung atau dengan kata lain truk agen baru boleh pergi ke area bongkar tabung jika tidak ada lagi truk agen di sana.

3. Variabel *jumlah_truk3kg_muatkan*. Variabel ini menunjukkan status apakah ada truk agen yang sedang melakukan pemuatan tabung 3 kg isi. Oleh karenanya, variabel ini ditempatkan di area muatkan tabung. Status variabel ini ditunjukkan dengan angka 1 atau 0 dimana jika status = 1 maka berarti ada truk agen di area muatkan tabung; sedangkan status = 0 maka berarti tidak ada truk agen di area muatkan

tabung. Penentuan status variabel ini dibantu dengan fungsi *INC* untuk menaikkan nilai variabel dan dengan fungsi *DEC* untuk menurunkan nilai variabel. Maksudnya ialah saat truk agen berada di area muatkan tabung maka fungsi *INC jumlah_truk3kg_muatkan, 1* membuat status variabel menjadi 1. Sedangkan jika truk agen sudah meninggalkan area muatkan tabung, maka fungsi *DEC jumlah_truk3kg_muatkan, 1* membuat status variabel menjadi 0. Penentuan status variabel ini berguna untuk membuat antrian pemuatan tabung. Untuk membuat antrian dengan bantuan status variabel ini, maka di titik lokasi sebelum area muatkan tabung yaitu di lokasi *Bongkar3_2* dibuatlah fungsi logika *IF..THEN..* dengan memanfaatkan status variabel ini. Simulasi antrian ini dibuat dengan kalimat

```
IF jumlah_truk3kg_muatkan = 1 THEN
{
  WAIT UNTIL jumlah_truk3kg_muatkan = 0
}
```

Kalimat di atas berarti jika ada truk agen di area muatkan tabung maka tunggu hingga tidak ada truk agen di area muatkan tabung atau dengan kata lain truk agen baru boleh pergi ke area muatkan tabung jika tidak ada lagi truk agen di sana.

4. Variabel *Jumlah_truk3kg_keluar*. Variabel ini menghitung jumlah truk agen yang keluar dari SPBE sehingga variabel ini ditempatkan setelah area muatkan tabung dekat pintu keluar SPBE. Nilai variabel ini bersifat kumulatif sehingga akan terus bertambah. Penentuan sifat kumulatif pada variabel ini dilakukan dengan bantuan fungsi *INC* untuk menaikkan nilai variabel.
5. Variabel *jumlah_skid tank*. Variabel ini menghitung jumlah *skid tank* (truk tangki) yang melakukan pengisian gas ke *storage tank* milik SPBE sehingga variabel ini ditempatkan dekat area pengisian gas dekat rumah pompa kompresor dan *storage tank* milik SPBE. Nilai variabel ini bersifat kumulatif sehingga akan terus bertambah. Penentuan sifat

kumulatif pada variabel ini dilakukan dengan bantuan fungsi *INC* untuk menaikkan nilai variabel.

6. Variabel *jumlah_tabung3kg_masuk*. Variabel ini ditempatkan di lokasi *Persediaan_tabung3kg_kosong* untuk memperlihatkan jumlah tabung 3 kg kosong sebelum tabung 3 kg kosong dimasukkan ke *conveyor*. Variabel memiliki nilai yang akan berkurang sejumlah tabung 3 kg kosong yang dimasukkan ke *conveyor*. Variabel ini juga memiliki nilai yang akan bertambah sejumlah tabung 3 kg kosong yang dibongkar dari truk agen. Sebelum simulasi dimulai, nilai awal variabel ini ialah 500. Ketika simulasi berjalan, nilai variabel ini mengalami fluktuasi karena nilai variabel ini akan berkurang sejumlah tabung 3 kg kosong yang dimasukkan ke dalam *conveyor* dan juga akan bertambah sejumlah tabung 3 kg kosong yang dibongkar dari truk agen. Simulasi penambahan nilai variabel ini dibantu dengan fungsi *INC* yang diletakkan di lokasi sebelum lokasi *Persediaan_tabung3kg_kosong* yaitu di lokasi *Bongkar3_2*. Sedangkan simulasi pengurangan nilai variabel ini dibantu dengan fungsi *DEC* yang diletakkan pada *move logic* lokasi *Persediaan_tabung3kg_kosong* yaitu saat akan memasuki *conveyor*.
7. Variabel *Resource3kg*. Variabel ini menunjukkan status berapa jumlah sisa mesin *carousel* yang *idle*. Mesin *carousel* berjumlah 12 unit. Jika semua mesin *carousel* bekerja (sedang mengisi tabung 3 kg kosong), maka status variabel ini = 0. Akan tetapi jika ada satu atau dua unit mesin *carousel* yang *idle* (tidak sedang mengisi tabung 3 kg kosong) maka status variabel ini = 1 atau 2. Pengurangan nilai variabel ini dilakukan dengan bantuan fungsi *DEC* yang diletakkan pada mesin *carousel* dengan nomor urut satu.
8. Variabel *M3_1* hingga *M3_12*. Variabel ini berjumlah 12 unit yang mewakili jumlah mesin *carousel* dengan nomor urut 1 hingga 12. Variabel ini menunjukkan status apakah mesin di nomor urut tertentu sedang bekerja atau tidak. Status nilai variabel = 1 menandakan mesin sedang bekerja (mengisi tabung 3 kg). Sedangkan status nilai variabel

= 0 menandakan mesin *idle* (tidak mengisi tabung 3 kg). Fungsi *INC* digunakan untuk menaikkan nilai variabel sehingga fungsi ini diletakkan di *operation* lokasi. Sedangkan fungsi *DEC* digunakan untuk menurunkan nilai variabel sehingga fungsi ini diletakkan di *move logic* lokasi. Penentuan status pada nilai variabel ini berhubungan dengan fungsi yang digunakan untuk menggeser tabung 3 kg dari satu mesin ke mesin selanjutnya hingga keluar dari sistem pengisian mesin *carousel* dengan tujuan menggambarkan perputaran kedua belas mesin *carousel* seperti yang terjadi pada sistem yang sebenarnya. Untuk mensimulasikan perputaran mesin *carousel* maka digunakan gabungan fungsi *IF..THEN..* dengan fungsi *SEND* sebagai berikut

```
IF M3_12 > 0 THEN
SEND 1 Tabung3kg TO T3_3
IF M3_11 > 0 THEN
SEND 1 Tabung3kg TO F3_12
IF M3_10 > 0 THEN
SEND 1 Tabung3kg TO F3_11
IF M3_9 > 0 THEN
SEND 1 Tabung3kg TO F3_10
IF M3_8 > 0 THEN
SEND 1 Tabung3kg TO F3_9
IF M3_7 > 0 THEN
SEND 1 Tabung3kg TO F3_8
IF M3_6 > 0 THEN
SEND 1 Tabung3kg TO F3_7
IF M3_5 > 0 THEN
SEND 1 Tabung3kg TO F3_6
IF M3_4 > 0 THEN
SEND 1 Tabung3kg TO F3_5
IF M3_3 > 0 THEN
SEND 1 Tabung3kg TO F3_4
```

IF M3_2 > 0 THEN

SEND 1 Tabung3kg TO F3_3

IF M3_1 > 0 THEN

SEND 1 Tabung3kg TO F3_2

Arti dari kalimat di atas ialah jika ada tabung gas 3 kg isi di mesin *carousel* dengan nomor urut 12 maka kirim tabung tersebut ke *conveyor* (T3_3). Jika ada tabung gas 3 kg isi di mesin *carousel* dengan nomor urut 11 maka kirim tabung tersebut ke mesin *carousel* dengan nomor urut 12. Jika ada tabung gas 3 kg isi di mesin *carousel* dengan nomor urut 10 maka kirim tabung tersebut ke mesin *carousel* dengan nomor urut 11. Jika ada tabung gas 3 kg isi di mesin *carousel* dengan nomor urut 9 maka kirim tabung tersebut ke mesin *carousel* dengan nomor urut 10. Jika ada tabung gas 3 kg isi di mesin *carousel* dengan nomor urut 8 maka kirim tabung tersebut ke mesin *carousel* dengan nomor urut 9. Jika ada tabung gas 3 kg isi di mesin *carousel* dengan nomor urut 7 maka kirim tabung tersebut ke mesin *carousel* dengan nomor urut 8. Jika ada tabung gas 3 kg isi di mesin *carousel* dengan nomor urut 6 maka kirim tabung tersebut ke mesin *carousel* dengan nomor urut 7. Jika ada tabung gas 3 kg isi di mesin *carousel* dengan nomor urut 5 maka kirim tabung tersebut ke mesin *carousel* dengan nomor urut 6. Jika ada tabung gas 3 kg isi di mesin *carousel* dengan nomor urut 4 maka kirim tabung tersebut ke mesin *carousel* dengan nomor urut 5. Jika ada tabung gas 3 kg isi di mesin *carousel* dengan nomor urut 3 maka kirim tabung tersebut ke mesin *carousel* dengan nomor urut 4. Jika ada tabung gas 2 kg isi di mesin *carousel* dengan nomor urut 5 maka kirim tabung tersebut ke mesin *carousel* dengan nomor urut 3. Jika ada tabung gas 3 kg isi di mesin *carousel* dengan nomor urut 1 maka kirim tabung tersebut ke mesin *carousel* dengan nomor urut 2. Maksud kenapa logika pemrograman dibuat mundur dari mesin *carousel* bernomor urut 12 ke mesin *carousel* bernomor urut 1 ialah untuk menghindari terjadinya “tabrakan” jika

mesin bernomor selanjutnya ternyata masih mengisi tabung padahal tabung harus segera digeser.

9. Variabel *jumlah_Tabung3kg_over*. Variabel ini menghitung jumlah tabung 3 kg yang jumlah isinya melebihi standar sehingga variabel ini ditempatkan di *correctionscale*. Nilai variabel ini bersifat kumulatif sehingga akan terus bertambah. Penentuan sifat kumulatif pada variabel ini dilakukan dengan bantuan fungsi *INC* untuk menaikkan nilai variabel.
10. Variabel *jumlah_Tabung3kg_under*. Variabel ini menghitung jumlah tabung 3 kg yang jumlah isinya kurang dari standar sehingga variabel ini ditempatkan di *correctionscale*. Nilai variabel ini bersifat kumulatif sehingga akan terus bertambah. Penentuan sifat kumulatif pada variabel ini dilakukan dengan bantuan fungsi *INC* untuk menaikkan nilai variabel.
11. Variabel *jumlah_tabung_di_mesin_evakuasi3*. Variabel ini menunjukkan status apakah ada tabung 3 kg bocor yang sedang diproses oleh mesin evakuasi. Oleh karenanya, variabel ini ditempatkan di area mesin evakuasi. Status variabel ini ditunjukkan dengan angka 1 atau 0 dimana jika status = 1 maka berarti ada tabung 3 kg bocor yang sedang diproses oleh mesin evakuasi; sedangkan status = 0 maka berarti tidak ada tabung 3 kg bocor yang sedang diproses oleh mesin evakuasi. Penentuan status variabel ini dibantu dengan fungsi *INC* untuk menaikkan nilai variabel dan dengan fungsi *DEC* untuk menurunkan nilai variabel.
12. Variabel *jumlah_tabung3kg_bocor*. Variabel ini menghitung jumlah tabung 3 kg yang bocor sehingga variabel ini ditempatkan di *Area_tumpuk_tabung3kg_bocor*. Nilai variabel ini bersifat kumulatif sehingga akan terus bertambah. Penentuan sifat kumulatif pada variabel ini dilakukan dengan bantuan fungsi *INC* untuk menaikkan nilai variabel.
13. Variabel *Jumlah_tabung3kg_keluar*. Variabel ini menghitung jumlah akumulasi tabung 3 kg isi yang dihasilkan sehingga variabel ini

ditempatkan di lokasi *Persediaan_tabung3kg_isi*. Nilai variabel ini bersifat kumulatif sehingga akan terus bertambah. Penentuan sifat kumulatif pada variabel ini dilakukan dengan bantuan fungsi *INC* untuk menaikkan nilai variabel.

d. *Counter* pada model. Pada dasarnya, *counter* pada model simulasi digunakan untuk mencatat jumlah suatu benda yang melalui suatu lokasi. Perbedaan antara *counter* dengan variabel secara umum ialah *counter* hanya mencatat jumlah benda yang melalui suatu lokasi (sehingga jumlah benda tersebut bersifat fluktuatif) tanpa bantuan fungsi apapun; sedangkan penggunaan variabel lebih luas daripada *counter*; misalnya digunakan dengan bantuan fungsi tertentu untuk menghitung jumlah suatu benda (sehingga jumlah benda tersebut bersifat statis). Terdapat dua buah *counter* yang digunakan pada model simulasi ini. Satu buah *counter* digunakan untuk suatu benda yang bersifat diskrit. Dan satu buah *counter* digunakan untuk suatu benda yang bersifat kontinyu.

1. *Counter* di lokasi *bayangan_3*. *Counter* ini mencatat jumlah tabung 3 kg isi yang “singgah” di lokasi ini sehingga kita bisa mengetahui berapa jumlah tabung 3 kg isi yang dimuatkan ke atas truk agen dan berapa jumlah tabung 3 kg isi yang diproduksi.
2. *Counter* di lokasi jenis *tank* yaitu *storage tank*. *Counter* ini mencatat fluktuasi jumlah gas yang berada di *storage tank*. Karena pada sistem yang nyata terjadi bahwa jumlah gas dalam tangki bisa berkurang dan bisa bertambah sehingga dalam model ini kita dapat mengamati fluktuasi jumlah gas dalam *storage tank* selama simulasi berlangsung dengan bantuan *counter* ini.

e. Lama waktu simulasi

Simulasi ini berlangsung selama tujuh hari (satu pekan) dengan tiap hari kerja dimulai dari jam 06.00 hingga 18.00 dengan dua kali waktu istirahat (jam 12.00-13.00 dan jam 14.45-15.15).

4.2. Verifikasi Dan Validasi Model

4.2.1. Verifikasi Model

Verifikasi model merupakan metode untuk memastikan apakah model simulasi telah dibuat dengan benar sesuai dengan yang diinginkan. Ada beberapa cara untuk melakukan verifikasi pada model simulasi:

a. Memeriksa kode pada model simulasi

Pemeriksaan kode tersebut dilakukan oleh peneliti dan pihak yang memiliki pengetahuan luas dalam pemodelan serta pihak yang paham terhadap sistem yang dimodelkan. Selain itu, ProModel tidak mengeluarkan *debugger* ketika model simulasi berlangsung. Hal ini menandakan kode pada model simulasi berjalan dengan benar.

b. Memeriksa *output* yang logis

Dalam model simulasi, terdapat hubungan antara operasional dengan nilai kuantitatif yang diprediksi selama simulasi berjalan. Salah satu contoh hubungan antara operasional dengan nilai kuantitatif yang diprediksi selama simulasi berjalan pada model ini adalah ketika operasional yaitu pembongkaran tabung 3 kg kosong sebanyak 230 buah dari truk agen ke lokasi *Persediaan_tabung3kg_kosong* terjadi, maka nilai kuantitatif yang diprediksi selama simulasi berjalan yaitu berupa variabel *jumlah_tabung3kg_masuk* akan bertambah sebanyak 230. Nilai kuantitatif yaitu 230 merupakan nilai yang diinginkan jika operasional pembongkaran tabung terjadi. Selain itu, ada beberapa *counter* dan variabel yang juga menunjukkan nilai kuantitatif yang diprediksi dengan operasional yang terjadi seperti:

1. *Counter* pada *storage tank* selalu menunjukkan angka yang menggambarkan jumlah gas pada *storage tank*. Dimana angka ini selalu berubah seperti yang diinginkan sesuai operasional yang terjadi, misalnya ketika *skid tank* melakukan pengisian pada *storage tank* maka jumlah gas pada *storage tank* akan bertambah 9000, kemudian ketika mesin *carousel* mengisi tabung maka jumlah gas pada *storage tank* akan berkurang 3 per tabung, sedangkan ketika *correctionscale* melakukan penyesuaian pada isi tabung maka jumlah gas pada *storage*

tank akan bertambah dan berkurang 1 per tabung, dan ketika evakuasi mengembalikan isi tabung gas yang bocor maka jumlah gas pada *storage tank* akan bertambah 3 per tabung.

2. Variabel *jumlah_tabung3kg_masuk* pada lokasi *Persediaan_tabung3kg_kosong* akan berkurang 1 tiap 1 buah tabung masuk ke *conveyor* dan akan bertambah 1 hingga 230 saat truk agen melakukan operasional pembongkaran tabung

3. Variabel *Jumlah_tabung3kg_keluar* pada lokasi *Persediaan_tabung3kg_isi* akan bertambah 1 tiap 1 tabung keluar dari *conveyor* dan akan berkurang 1 hingga 230 saat truk agen melakukan operasional pemuatan tabung

c. Mengamati perilaku sistem yang benar dalam animasi

Animasi saat model menjalankan simulasi akan menunjukkan apakah entitas dan sumberdaya berjalan pada jalur yang benar yang telah didefinisikan pada kode model serta apakah tangki berubah jumlahnya sesuai dengan penambahan dan pengurangan gas yang terjadi. Saat simulasi berjalan, animasi pada model memperlihatkan bahwa semua entitas dan sumberdaya berjalan pada jalur yang telah didefinisikan sebelumnya serta jumlah gas pada tangki berikut aliran gas pada pipa telah memperlihatkan perubahan (baik penambahan maupun pengurangan) yang benar dan sesuai.

d. Menggunakan fungsi *debug* dan *trace* untuk mengamati perilaku sistem yang tidak terlihat secara animasi.

Dengan menggunakan fungsi *debug* dan *trace* untuk mengamati perilaku sistem yang tidak terlihat secara animasi maka terlihat bahwa *logic* yang digunakan berjalan sesuai dengan keinginan peneliti.

4.2.2. Validasi Model

Validasi adalah proses penentuan apakah model berguna dan secara akurat menggambarkan sistem yang sebenarnya (Hoover dan Perry 1990). Sebenarnya sistem SPBE untuk tabung 3 kg telah berjalan di beberapa lokasi. Akan tetapi baru sekitar di empat lokasi didirikan SPBE untuk tabung 3 kg dengan *Layout* baru dari Pertamina yang digunakan pada model yang sedang diteliti ini. Oleh karena

proses operasionalnya sama maka pengambilan data dilakukan dari SPBE untuk tabung 3 kg yang telah ada. Oleh karena itu, validasi model dilakukan dengan dua cara:

1. Menjalankan model dalam kondisi ekstrim. Misalnya dengan meniadakan kedatangan *skid tank* ($occurrences\ skid\ tank = 0$) sehingga sistem pengisian tabung 3 kg masih berjalan hingga jumlah gas pada *storage tank* habis dan ketika jumlah gas pada *storage tank* habis maka sistem pengisian tabung 3 kg terhenti karena tidak ada *skid tank* yang melakukan penambahan gas pada *storage tank*.
2. Membandingkan *input-output* dengan sistem serupa yang telah ada. Dengan memasukkan semua data *input* dari sistem yang telah ada maka seharusnya *output* yang dihasilkan juga identik dengan sistem nyata yang telah berjalan. Dengan memasukkan *input* dari sistem nyata ke model, maka dihasilkanlah *output* yang dihasilkan model sama dengan *output* yang dihasilkan sistem nyata yaitu sebanyak 1200 unit tabung gas 3 kg setiap jam. Ini menandakan bahwa model ini telah lulus uji validasi dengan hasil yang baik.

4.3. Eksperimen

4.3.1. Kondisi Awal Model

Model simulasi ini memiliki kondisi awal yang merupakan tiruan dari kondisi pada sistem nyata dengan tujuan menguji verifikasi dan validasi model. Setelah model kondisi awal ini lulus uji verifikasi dan validasi model, maka model ini dianggap telah menyerupai sistem nyata dan dapat dilakukan eksperimen lebih lanjut. Kondisi awal pada model ini memiliki karakteristik atau variabel *input* yang ditampilkan dengan tabel berikut ini.

Tabel 4.1 Kondisi Awal Model

No.	Nama Proses	Distribusi Probabilitas yang sesuai
1	Tingkat kedatangan <i>skid tank</i>	Poisson (1.1) HR
2	Pengisian <i>storage tank</i>	Normal (25., 5.) MIN
3	Tingkat kedatangan agen	Poisson (9.5) MIN
4	Pembongkaran tabung	Triangular (3.83, 5.65, 9.10) MIN
5	<i>Check Weight</i>	Uniform (0.8, 1.2) SEC
6	Pengisian tabung dari <i>carousel</i>	2.5 SEC
7	<i>Check Scale</i>	Triangular (0.732, 1., 1.16) SEC
8	<i>Correction Scale</i>	Lognormal (0.286, 0.171) SEC
9	<i>Leak Tester</i>	Normal (1.71, 0.151) SEC
10	Penyegelan tutup tabung	Normal (0.413, 0.008) SEC
11	Pemuatan tabung	Triangular (7., 10.1, 11.7) MIN
12	Evakuasi	Triangular (0.921, 1.4, 1.52) SEC

Dengan memasukkan variabel *input* seperti pada tabel di atas ke dalam model simulasi, maka dihasilkanlah *output* sebesar 1200 unit tabung gas 3 kg setiap jam. Hasil ini sangat sesuai dengan kondisi pada sistem nyata. Dalam sistem nyata, kedatangan truk agen dan kedatangan *skid tank* tidak menimbulkan masalah dalam sistem pengisian SPBE.

Selain itu, SPBE pada kondisi awal menggunakan 12 unit mesin dalam satu *carousel*. Sebagai tambahan, sebelum simulasi berjalan telah ditentukan bahwa SPBE memiliki *buffer* tabung gas 3 kg kosong sebanyak 500 unit, *buffer* tabung gas 3 kg isi sebanyak 100 unit, dan *buffer* gas di *storage tank* sebanyak 9000 kg. Telah ditetapkan pula bahwa pada hari senin terdapat 50 unit truk agen dan 4 unit *skid tank* yang datang, pada hari selasa terdapat 45 unit truk agen dan 3 unit *skid tank* yang datang, pada hari rabu terdapat 48 unit truk agen dan 3 unit *skid tank* yang datang, pada hari kamis terdapat 42 unit truk agen dan 4 unit *skid tank* yang datang, pada hari jumat terdapat 33 unit truk agen dan 2 unit *skid tank* yang datang, pada hari sabtu terdapat 48 unit truk agen dan 4 unit *skid tank* yang datang, dan pada hari minggu terdapat 45 unit truk agen dan 3 unit *skid tank* yang datang.

Simulasi dilakukan dengan tiga kali replikasi dan tingkat presisi waktu 0.001 detik. Simulasi ini berlangsung selama tujuh hari (satu pekan) yaitu dari hari senin hingga hari minggu. Setiap hari kerja berlangsung dari jam 06.00 hingga jam 18.00. Setiap hari kerja memiliki dua kali waktu istirahat, yaitu waktu istirahat yang pertama berlangsung dari jam 12.00 hingga jam 13.00; sedangkan waktu istirahat yang kedua berlangsung dari jam 14.45 hingga jam 15.15.

4.3.2. Rancangan Skenario

Eksperimen ini mencoba untuk menguji model SPBE melalui simulasi menggunakan ProModel untuk mengetahui variabel-variabel apa saja yang lebih mempengaruhi sistem SPBE dengan beberapa kondisi tertentu. Ada dua kondisi eksternal yang mempengaruhi operasional SPBE yaitu tingkat kedatangan truk agen dan tingkat kedatangan *skid tank*. Sedangkan dalam internal sistem SPBE, waktu proses pada mesin *carousel* merupakan *cycle time* dari keseluruhan waktu proses dalam sistem SPBE sehingga menjadi waktu penentu produksi. Waktu

proses pada mesin *carousel* dapat diubah dengan dua cara yaitu dengan memasang waktu putar *carousel* dan dengan mengatur jumlah mesin pada satu *carousel*. Namun dalam eksperimen ini, waktu putar *carousel* dipasang tetap untuk semua skenario dan/atau kondisi sehingga eksperimen ini nantinya akan membandingkan jumlah mesin pada satu *carousel* yang manakah yang paling optimal dengan utilisasi yang terbaik yang dapat menghasilkan *output* terbanyak sesuai dengan kondisi tertentu yang akan di-skenario-kan.

Pada eksperimen ini, akan diteliti perilaku sistem SPBE dengan kombinasi tiga macam kondisi terhadap lima *output*. Ketiga macam kondisi tersebut ialah

1. Tingkat kedatangan truk agen. Akan digunakan dua jenis tingkat kedatangan truk agen pada eksperimen model simulasi SPBE ini yaitu normal dan ekstrim. Tingkat kedatangan truk agen jenis normal merupakan tingkat kedatangan truk agen yang normal sesuai dengan kondisi awal model dimana tingkat kedatangan truk agen ini terdistribusi $P(9.5)$ MIN yaitu terdistribusi sesuai distribusi Poisson dengan interval kedatangan 9.5 menit. Sedangkan tingkat kedatangan truk agen jenis ekstrim merupakan tingkat kedatangan truk agen yang terdistribusi $ER(6, 2)$ MIN yaitu terdistribusi sesuai distribusi Erlang dengan interval kedatangan rata-rata 6 menit dan parameter = 2. Pada tingkat kedatangan truk agen jenis ekstrim ini, truk agen akan datang ke SPBE dengan intensitas yang tinggi dalam interval antar-kedatangan yang relatif singkat sehingga akan terjadi antrian truk agen yang padat di halaman SPBE untuk menunggu membongkar dan memuat tabung gas 3 kg.
2. Tingkat kedatangan *skid tank*. Akan digunakan dua jenis tingkat kedatangan *skid tank* pada eksperimen model simulasi SPBE ini yaitu normal dan ekstrim. Tingkat kedatangan *skid tank* jenis normal merupakan tingkat kedatangan *skid tank* yang normal sesuai dengan kondisi awal model dimana tingkat kedatangan truk agen ini terdistribusi $P(1.1)$ HR yaitu terdistribusi sesuai distribusi Poisson dengan interval kedatangan rata-rata 1.1 jam. Sedangkan tingkat kedatangan *skid tank* jenis ekstrim merupakan tingkat kedatangan *skid tank* yang terdistribusi $ER(4, 2)$ HR yaitu terdistribusi sesuai distribusi Erlang dengan interval

kedatangan rata-rata 4 jam dan parameter = 2. Pada tingkat kedatangan *skid tank* jenis ekstrim ini, *skid tank* akan datang ke SPBE dengan intensitas yang rendah dalam interval antar-kedatangan yang relatif lama sehingga akan terjadi keterlambatan kedatangan *skid tank* yang berdampak pada terhentinya proses pengisian tabung gas 3 kg karena *storage tank* yang kosong belum terisi kembali.

3. Jumlah mesin pada satu *carousel*. Satu *carousel* minimal terdiri atas 12 unit mesin dan maksimal terdiri atas 18 unit mesin. Dengan kata lain, satu mesin *carousel* dapat terdiri atas 12 atau 13 atau 14 atau 15 atau 16 atau 17, atau 18 unit mesin. Waktu putar *carousel* dipasang tetap (konstan) yaitu 30 detik untuk satu kali putaran, sedangkan jumlah mesin pada *carousel* divariasikan antara 12 hingga 18 mesin. Logikanya, semakin banyak jumlah mesin pada *carousel* menyebabkan semakin cepatnya proses pengisian sehingga semakin banyak tabung gas 3 kg yang dihasilkan serta semakin cepat pula berkurangnya jumlah gas dalam *storage tank*.

Hasil kombinasi antara variasi tingkat kedatangan truk agen, variasi tingkat kedatangan *skid tank*, dan variasi jumlah mesin pada *carousel* akan menghasilkan lima jenis *output* yang berbeda. Kelima jenis *output* tersebut ialah

1. Unit tabung. *Output* ini merupakan akumulasi jumlah tabung gas 3 kg yang dihasilkan selama satu pekan simulasi berjalan.
2. Unit truk. *Output* ini merupakan akumulasi jumlah truk agen yang keluar dari SPBE selama satu pekan simulasi berjalan.
3. Unit *skid tank*. *Output* ini merupakan akumulasi jumlah *skid tank* yang melakukan pengisian *storage tank* di SPBE selama satu pekan simulasi berjalan.
4. Utilisasi mesin *carousel*. *Output* ini merupakan rata-rata utilisasi mesin *carousel* selama satu pekan simulasi berjalan.
5. *Output* gas harian. *Output* ini merupakan rata-rata jumlah gas yang diisi ke dalam tabung gas 3 kg oleh SPBE tiap hari selama satu pekan simulasi berjalan.

Secara keseluruhan, eksperimen ini akan terbagi menjadi sebanyak empat skenario sebagai berikut:

1. Skenario 1: Truk Normal dan *Skid Tank* Normal. Maksudnya ialah skenario dimana tingkat kedatangan truk berlangsung secara normal dan tingkat kedatangan *skid tank* berlangsung secara normal.
2. Skenario 2: Truk Normal dan *Skid Tank* Ekstrim. Maksudnya ialah skenario dimana tingkat kedatangan truk berlangsung secara normal dan tingkat kedatangan *skid tank* berlangsung secara ekstrim.
3. Skenario 3: Truk Ekstrim dan *Skid Tank* Normal. Maksudnya ialah skenario dimana tingkat kedatangan truk berlangsung secara ekstrim dan tingkat kedatangan *skid tank* berlangsung secara normal.
4. Skenario 4: Truk Ekstrim dan *Skid Tank* Ekstrim. Maksudnya ialah skenario dimana tingkat kedatangan truk berlangsung secara ekstrim dan tingkat kedatangan *skid tank* berlangsung secara ekstrim.

Sebagai tambahan, sebelum simulasi berjalan untuk semua skenario telah ditentukan bahwa SPBE memiliki *buffer* tabung gas 3 kg kosong sebanyak 500 unit, *buffer* tabung gas 3 kg isi sebanyak 100 unit, dan *buffer* gas di *storage tank* sebanyak 9000 kg. Telah ditetapkan pula bahwa pada hari senin akan terdapat 50 unit truk agen dan 4 unit *skid tank* yang datang, pada hari selasa akan terdapat 45 unit truk agen dan 3 unit *skid tank* yang datang, pada hari rabu akan terdapat 48 unit truk agen dan 3 unit *skid tank* yang datang, pada hari kamis akan terdapat 42 unit truk agen dan 4 unit *skid tank* yang datang, pada hari jumat akan terdapat 33 unit truk agen dan 2 unit *skid tank* yang datang, pada hari sabtu akan terdapat 48 unit truk agen dan 4 unit *skid tank* yang datang, dan pada hari minggu akan terdapat 45 unit truk agen dan 3 unit *skid tank* yang datang.

Sedangkan untuk setiap skenario yang akan disimulasikan, eksperimen ini menguji pula jumlah mesin pada *carousel* yaitu dengan mensimulasikan di tiap skenario mesin pada *carousel* yang dapat berjumlah 12 atau 13 atau 14 atau 15 atau 16 atau 17, atau 18 unit mesin. Simulasi untuk semua skenario dilakukan dengan tiga kali replikasi dan dengan tingkat presisi waktu 0.001 detik. Simulasi ini berlangsung selama tujuh hari (satu pekan) yaitu dari hari senin hingga hari

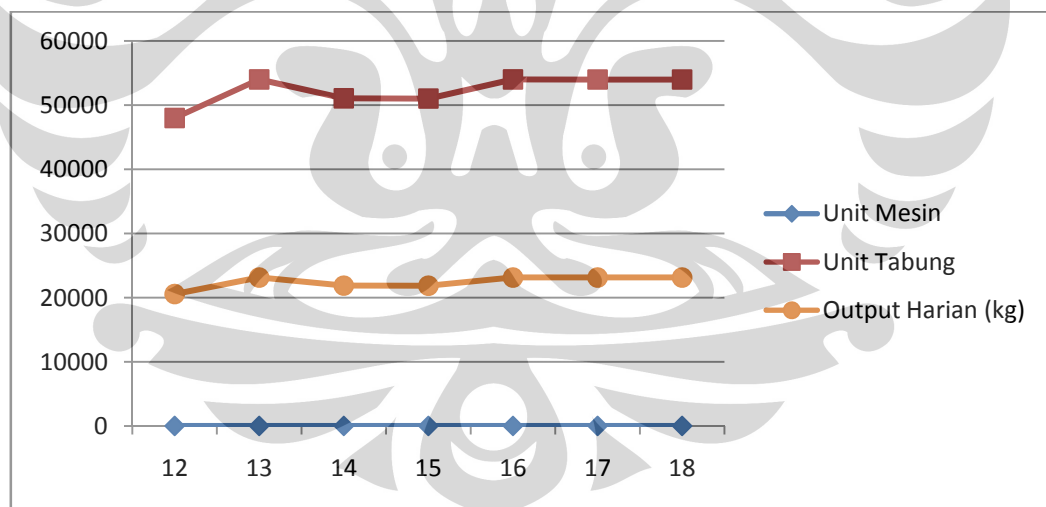
minggu. Setiap hari kerja berlangsung dari jam 06.00 hingga jam 18.00. Setiap hari kerja memiliki dua kali waktu istirahat, yaitu waktu istirahat yang pertama berlangsung dari jam 12.00 hingga jam 13.00; sedangkan waktu istirahat yang kedua berlangsung dari jam 14.45 hingga jam 15.15.

4.3.3. Analisa Hasil Skenario 1

Skenario 1 ialah skenario dimana tingkat kedatangan truk berlangsung secara normal dan tingkat kedatangan *skid tank* berlangsung secara normal. Tabel 4.2 dan gambar 4.7 berikut ini menampilkan hasil skenario 1.

Tabel 4.2 Hasil Skenario 1

1. Skenario kedatangan truk normal dan kedatangan <i>skid tank</i> normal					
Unit Mesin	Unit Tabung	Unit Truk	Unit <i>Skid Tank</i>	Utilisasi Mesin	Output Harian (kg)
12	47983	209	15	89,37%	20564
13	54004	236	17	89,07%	23145
14	51050	223	16	89,46%	21879
15	51016	223	16	89,19%	21864
16	54000	236	17	90,32%	23143
17	53991	236	17	89,91%	23139
18	53991	236	17	89,91%	23139



Gambar 4.7 Hasil Skenario 1

Dari tabel dan gambar di atas terlihat bahwa dalam skenario dimana tingkat kedatangan truk agen berlangsung secara normal dan tingkat kedatangan *skid tank* juga berlangsung secara normal maka dihasilkanlah tingkat *output* tertinggi yaitu sebanyak rata-rata 54000 unit tabung gas 3 kg dalam satu pekan, sebanyak rata-rata 236 unit truk agen yang terlayani di SPBE selama satu pekan, sebanyak rata-rata 17 unit *skid tank* yang melakukan pengisian ke *storage tank* di SPBE selama

satu pekan, sebanyak rata-rata 23000 kg dalam satu hari, dan memiliki tingkat utilitas terendah yaitu sebanyak rata-rata 90% utilisasi mesin *carousel* selama satu pekan; hal yang menarik ialah ternyata semua *output* tertinggi tersebut dihasilkan oleh mesin *carousel* yang terdiri atas 13, 16, 17, dan 18 unit.

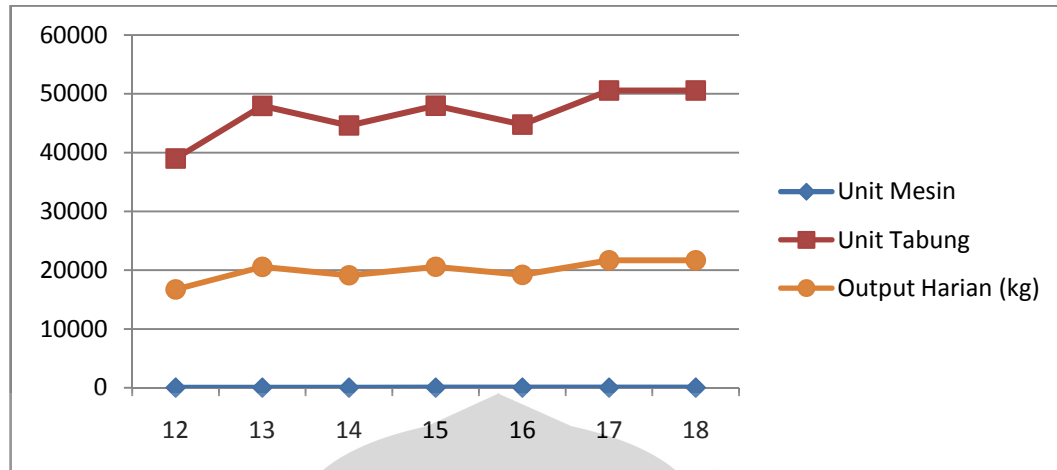
Ke-empat variasi unit mesin *carousel* tersebut memiliki tingkat *output* yang hampir sama tingginya. Akan tetapi, mesin *carousel* yang terdiri atas 13 unit ternyata lebih unggul daripada ketiga variasi yang lain karena memiliki utilisasi mesin yang paling rendah yaitu sebesar 89,07%; juga menghasilkan jumlah tabung selama satu pekan terbanyak yaitu sebesar 54004 unit sehingga menghabiskan rata-rata 23145 kg gas per hari. Ini menandakan bahwa mesin *carousel* yang terdiri dari 13 unit mesin merupakan komposisi yang paling sesuai dengan skenario ini. Hal ini disebabkan mesin *carousel* yang terdiri dari 13 unit mesin dapat melayani pengisian untuk truk agen yang datang secara normal ke SPBE sekaligus dapat menyesuaikan diri dengan tingkat kedatangan *skid tank*.

4.3.4. Analisa Hasil Skenario 2

Skenario 2 ialah skenario dimana tingkat kedatangan truk berlangsung secara normal dan tingkat kedatangan *skid tank* yang berlangsung secara ekstrim. Pada tingkat kedatangan *skid tank* jenis ekstrim ini, *skid tank* akan datang ke SPBE dengan intensitas yang rendah dalam interval antar-kedatangan yang relatif lama sehingga akan terjadi keterlambatan kedatangan *skid tank* yang berdampak pada terhentinya proses pengisian tabung gas 3 kg karena *storage tank* yang kosong belum terisi kembali. Tabel 4.3 dan gambar 4.8 berikut ini menampilkan hasil skenario 2.

Tabel 4.3 Hasil Skenario 2

2. Skenario kedatangan truk normal dan kedatangan <i>skid tank</i> ekstrim					
Unit Mesin	Unit Tabung	Unit Truk	Unit <i>Skid Tank</i>	Utilisasi Mesin	Output Harian (kg)
12	39000	168	12	91,05%	16714
13	47954	209	15	89,61%	20552
14	44622	185	14	89,69%	19124
15	47986	211	15	88,40%	20565
16	44767	188	14	91,87%	19186
17	50572	220	16	89,22%	21674
18	50572	220	16	89,22%	21674



Gambar 4.8 Hasil Skenario 2

Dari tabel dan gambar di atas terlihat bahwa dalam skenario dimana tingkat kedatangan truk agen berlangsung secara normal dan tingkat kedatangan *skid tank* yang berlangsung secara ekstrim maka dihasilkanlah tingkat *output* tertinggi yaitu sebanyak rata-rata 50572 unit tabung gas 3 kg dalam satu pekan, sebanyak rata-rata 220 unit truk agen yang terlayani di SPBE selama satu pekan, sebanyak rata-rata 16 unit *skid tank* yang melakukan pengisian ke *storage tank* di SPBE selama satu pekan, sebanyak rata-rata 21674 kg dalam satu hari, dan memiliki tingkat utilitas terendah yaitu sebanyak rata-rata 89,22% utilisasi mesin *carousel* selama satu pekan; hal yang menarik ialah ternyata semua *output* tertinggi tersebut dihasilkan oleh mesin *carousel* yang terdiri atas 17 dan 18 unit.

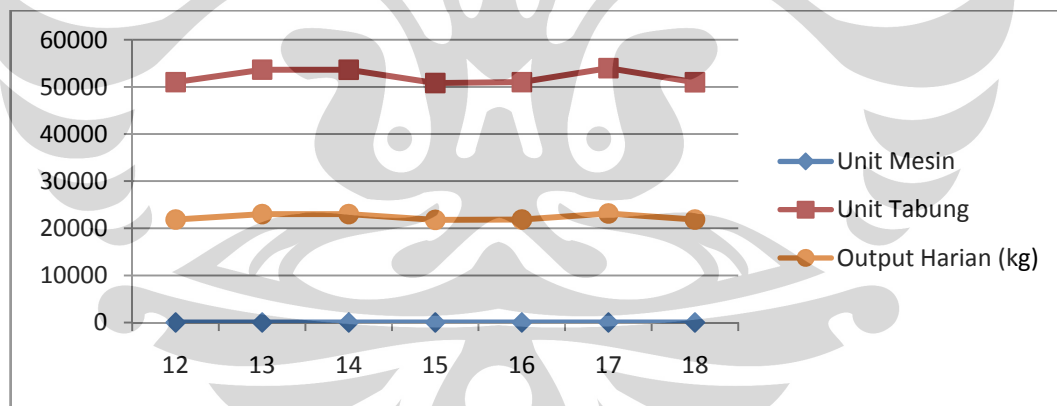
Ini menandakan bahwa mesin *carousel* yang terdiri dari 17 dan 18 unit mesin merupakan komposisi yang paling sesuai dengan skenario ini. Meskipun kedatangan *skid tank* sering terlambat dan bahkan tidak menentu, akan tetapi mesin *carousel* yang terdiri dari 17 dan 18 unit mesin dapat lebih banyak melayani truk agen yang datang secara normal di SPBE daripada variasi komposisi mesin *carousel* lainnya. Hal ini disebabkan mesin *carousel* yang terdiri dari 17 dan 18 unit lebih cepat melakukan proses pengisian tabung gas 3 kg (meskipun ini juga berarti lebih cepat mengurangi jumlah gas di *storage tank*); sehingga jumlah truk agen yang dapat dilayani lebih banyak sekaligus mengurangi antrian truk agen. Semakin cepat pengisian tabung gas maka semakin banyak truk agen yang dilayani sehingga semakin berkurang jumlah truk agen yang gagal dilayani walaupun kedatangan *skid tank* telat dan tidak menentu.

4.3.5. Analisa Hasil Skenario 3

Skenario 3 ialah skenario dimana tingkat kedatangan truk berlangsung secara ekstrim dan tingkat kedatangan *skid tank* berlangsung secara normal. Pada tingkat kedatangan truk agen jenis ekstrim ini, truk agen akan datang ke SPBE dengan intensitas yang tinggi dalam interval antar-kedatangan yang relatif singkat sehingga akan terjadi antrian truk agen yang padat di halaman SPBE untuk menunggu membongkar dan memuat tabung gas 3 kg. Tabel 4.4 dan gambar 4.9 berikut ini menampilkan hasil skenario 3.

Tabel 4.4 Hasil Skenario 3

3. Skenario kedatangan truk ekstrim dan kedatangan <i>skid tank</i> normal					
Unit Mesin	Unit Tabung	Unit Truk	Unit <i>Skid Tank</i>	Utilisasi Mesin	Output Harian (kg)
12	51005	224	16	87,42%	21859
13	53662	236	17	85,46%	22998
14	53637	236	17	85,38%	22987
15	50812	223	16	86,36%	21777
16	51008	224	16	86,15%	21861
17	53987	236	17	89,14%	23137
18	50994	223	16	90,87%	21855



Gambar 4.9 Hasil Skenario 3

Dari tabel dan gambar di atas terlihat bahwa dalam skenario dimana tingkat kedatangan truk agen berlangsung secara ekstrim dan tingkat kedatangan *skid tank* yang berlangsung secara normal maka dihasilkanlah tingkat *output* tertinggi yaitu sebanyak rata-rata 53987 unit tabung gas 3 kg dalam satu pekan, sebanyak rata-rata 236 unit truk agen yang terlayani di SPBE selama satu pekan, sebanyak rata-rata 17 unit *skid tank* yang melakukan pengisian ke *storage tank* di SPBE selama satu pekan, sebanyak rata-rata 23137 kg dalam satu hari, dan memiliki tingkat

utilitas yaitu sebanyak rata-rata 89,14% utilisasi mesin *carousel* selama satu pekan; hal yang menarik ialah ternyata semua *output* tertinggi tersebut dihasilkan oleh mesin *carousel* yang terdiri atas 17 unit mesin.

Ini menandakan bahwa mesin *carousel* yang terdiri dari 17 unit mesin merupakan komposisi yang paling sesuai dengan skenario ini. Hal yang cukup aneh juga terjadi yaitu mesin *carousel* yang terdiri dari 18 unit mesin ternyata memiliki tingkat *output* yang lebih rendah dari mesin *carousel* yang terdiri dari 17 unit mesin dan bahkan sebanding dengan mesin *carousel* yang terdiri dari 12, 15, dan 16 unit mesin. Sedangkan mesin *carousel* yang terdiri dari 17 unit mesin memiliki tingkat *output* yang hampir sama dengan mesin *carousel* yang terdiri dari 13, dan 14 unit mesin.

Keanehan seperti di atas dapat terjadi karena beberapa kemungkinan. Pertama, penyebab keanehan tersebut disebabkan oleh karena mesin *carousel* yang memiliki jumlah mesin yang lebih banyak berarti memiliki waktu proses pengisian tabung gas 3 kg yang lebih cepat sehingga lebih banyak truk agen yang terlayani sekaligus memiliki peluang mengurangi panjang/waktu antrian dan mengurangi jumlah truk agen yang gagal dilayani di SPBE. Akan tetapi, saat semakin cepat proses pengisian tabung gas 3 kg berarti semakin cepat pula berkurangnya jumlah gas di *storage tank* hingga suatu ketika jumlah gas di *storage tank* habis tetapi *skid tank* belum datang yang mengakibatkan proses pengisian tabung gas 3 kg terhenti selama beberapa jam sehingga berakibat pada banyaknya jumlah truk agen yang gagal datang ke SPBE dan gagal dilayani di SPBE.

Kemungkinan penyebab kedua ialah mesin *carousel* yang memiliki jumlah mesin yang lebih sedikit berarti memiliki waktu proses pengisian tabung gas 3 kg yang lebih lambat sehingga lebih sedikit truk agen yang terlayani sekaligus memiliki peluang menambah panjang/waktu antrian dan menambah jumlah truk agen yang gagal dilayani di SPBE. Sedangkan kemungkinan terakhir ialah mesin *carousel* yang memiliki jumlah mesin yang sesuai memiliki waktu proses pengisian tabung gas 3 kg yang tepat dengan kedatangan *skid tank* sehingga saat jumlah gas di *storage tank* habis, tidak butuh waktu yang lama menunggu terisinya *storage tank*

kembali karena kedatangan *skid tank* tepat. Hal inilah yang membuat lebih banyak truk agen yang terlayani sekaligus memiliki peluang mengurangi panjang/waktu antrian dan mengurangi jumlah truk agen yang gagal dilayani di SPBE.

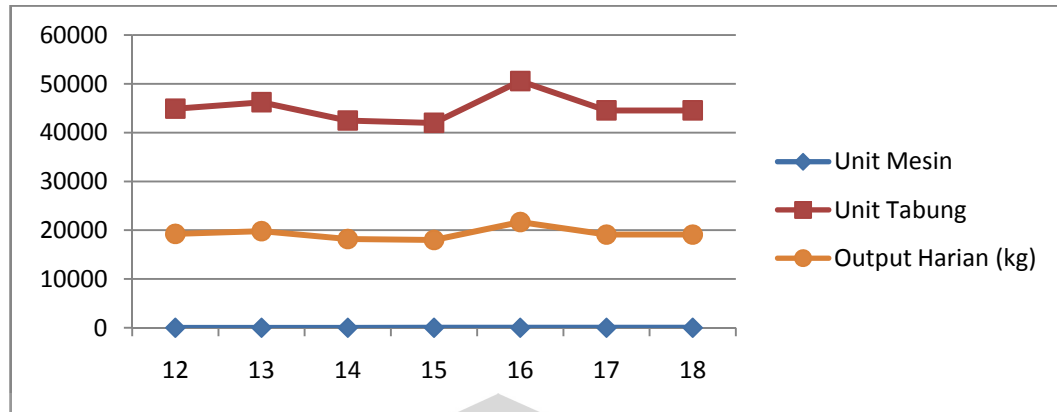
4.3.6. Analisa Hasil Skenario 4

Skenario 4 ialah skenario dimana tingkat kedatangan truk berlangsung secara ekstrim dan tingkat kedatangan *skid tank* berlangsung secara ekstrim. Pada tingkat kedatangan truk agen jenis ekstrim ini, truk agen akan datang ke SPBE dengan intensitas yang tinggi dalam interval antar-kedatangan yang relatif singkat sehingga akan terjadi antrian truk agen yang padat di halaman SPBE untuk menunggu membongkar dan memuat tabung gas 3 kg.

Pada tingkat kedatangan *skid tank* jenis ekstrim ini, *skid tank* akan datang ke SPBE dengan intensitas yang rendah dalam interval antar-kedatangan yang relatif lama sehingga akan terjadi keterlambatan kedatangan *skid tank* yang berdampak pada terhentinya proses pengisian tabung gas 3 kg karena *storage tank* yang kosong belum terisi kembali. Tabel 4.5 dan gambar 4.10 berikut ini menampilkan hasil skenario 4.

Tabel 4.5 Hasil Skenario 4

4. Skenario kedatangan truk ekstrim dan kedatangan <i>skid tank</i> ekstrim					
Unit Mesin	Unit Tabung	Unit Truk	Unit <i>Skid Tank</i>	Utilisasi Mesin	Output Harian (kg)
12	44906	195	14	87,19%	19245
13	46221	203	15	86,61%	19809
14	42473	182	14	88,76%	18203
15	41993	185	13	89,23%	17997
16	50567	220	16	89%	21672
17	44559	191	14	90,07%	19097
18	44559	191	14	90,07%	19097



Gambar 4.10 Hasil Skenario 4

Dari tabel dan gambar di atas terlihat bahwa dalam skenario dimana tingkat kedatangan truk agen berlangsung secara ekstrim dan tingkat kedatangan *skid tank* yang juga berlangsung secara ekstrim maka dihasilkanlah tingkat *output* tertinggi yaitu sebanyak rata-rata 50567 unit tabung gas 3 kg dalam satu pekan, sebanyak rata-rata 220 unit truk agen yang terlayani di SPBE selama satu pekan, sebanyak rata-rata 16 unit *skid tank* yang melakukan pengisian ke *storage tank* di SPBE selama satu pekan, sebanyak rata-rata 21672 kg dalam satu hari, dan memiliki tingkat utilitas yaitu sebanyak rata-rata 89% utilisasi mesin *carousel* selama satu pekan; hal yang menarik ialah ternyata semua *output* tertinggi tersebut dihasilkan oleh mesin *carousel* yang terdiri atas 16 unit.

Ini menandakan bahwa mesin *carousel* yang terdiri dari 16 unit mesin merupakan komposisi yang paling sesuai dengan skenario ini. Hal yang cukup aneh juga terjadi yaitu mesin *carousel* yang terdiri dari 18 unit mesin ternyata memiliki tingkat *output* yang lebih rendah dari mesin *carousel* yang terdiri dari 16 unit mesin dan bahkan sebanding dengan mesin *carousel* yang terdiri dari 12, 14, dan 17 unit mesin. Sedangkan mesin *carousel* yang terdiri dari 16 unit mesin memiliki tingkat *output* yang hampir sama dengan mesin *carousel* yang terdiri dari 13 unit mesin. Bahkan mesin *carousel* yang terdiri dari 15 unit mesin memiliki tingkat *output* yang paling rendah.

Keanehan seperti di atas dapat terjadi karena beberapa kemungkinan. Pertama, penyebab keanehan tersebut disebabkan oleh karena mesin *carousel* yang memiliki jumlah mesin yang lebih banyak berarti memiliki waktu proses

pengisian tabung gas 3 kg yang lebih cepat sehingga lebih banyak truk agen yang terlayani sekaligus memiliki peluang mengurangi panjang/waktu antrian dan mengurangi jumlah truk agen yang gagal dilayani di SPBE. Akan tetapi, saat semakin cepat proses pengisian tabung gas 3 kg berarti semakin cepat pula berkurangnya jumlah gas di *storage tank* hingga suatu ketika jumlah gas di *storage tank* habis tetapi *skid tank* belum datang yang mengakibatkan proses pengisian tabung gas 3 kg terhenti selama beberapa jam sehingga berakibat pada banyaknya jumlah truk agen yang gagal datang ke SPBE dan gagal dilayani di SPBE.

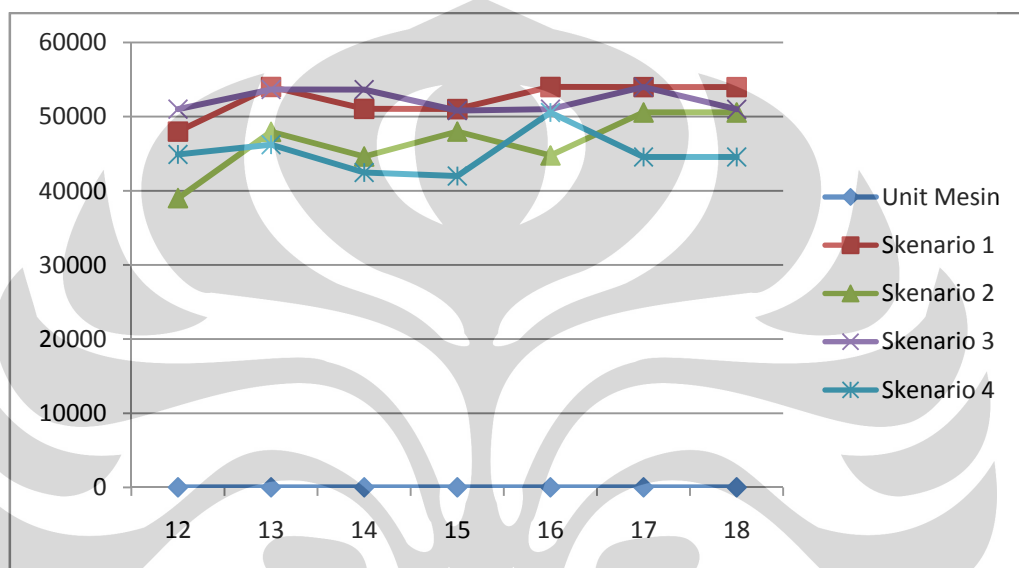
Kemungkinan penyebab kedua ialah mesin *carousel* yang memiliki jumlah mesin yang lebih sedikit berarti memiliki waktu proses pengisian tabung gas 3 kg yang lebih lambat sehingga lebih sedikit truk agen yang terlayani sekaligus memiliki peluang menambah panjang/waktu antrian dan menambah jumlah truk agen yang gagal dilayani di SPBE. Sedangkan kemungkinan terakhir ialah mesin *carousel* yang memiliki jumlah mesin yang sesuai memiliki waktu proses pengisian tabung gas 3 kg yang tepat dengan kedatangan *skid tank* sehingga saat jumlah gas di *storage tank* habis, tidak butuh waktu yang lama menunggu terisinya *storage tank* kembali karena kedatangan *skid tank* tepat. Hal inilah yang membuat lebih banyak truk agen yang terlayani sekaligus memiliki peluang mengurangi panjang/waktu antrian dan mengurangi jumlah truk agen yang gagal dilayani di SPBE.

4.3.7 Analisa Perilaku Jumlah Mesin pada *Carousel* terhadap Semua Skenario

Karena jumlah mesin pada *carousel* bervariasi antara 12 hingga 18 unit dan semua variasi tersebut diuji di semua skenario, maka perlu diketahui perilaku tiap variasi jumlah mesin pada *carousel* di tiap skenario. Diharapkan dari pengamatan perilaku jumlah mesin pada *carousel* terhadap semua skenario ini, dapat diketahui variasi jumlah mesin pada *carousel* yang manakah yang dapat menghasilkan tingkat *output* yang relatif tinggi sekaligus stabil pada semua jenis skenario. Berikut ini disajikan tabel dan gambar perilaku jumlah mesin pada *carousel* terhadap semua skenario.

Tabel 4.6 Perilaku Jumlah Mesin pada *Carousel* terhadap Semua Skenario

Jumlah Tabung Tiap Pekan Tiap Variasi Jumlah Mesin di <i>Carousel</i>					
Unit Mesin	Skenario 1	Skenario 2	Skenario 3	Skenario 4	Rata-rata
12	47983	39000	51005	44906	45724
13	54004	47954	53662	46221	50460
14	51050	44622	53637	42473	47946
15	51016	47986	50812	41993	47952
16	54000	44767	51008	50567	50086
17	53991	50572	53987	44559	50777
18	53991	50572	50994	44559	50029

**Gambar 4.11** Perilaku Jumlah Mesin pada *Carousel* terhadap Semua Skenario

Dari tabel dan gambar di atas terlihat bahwa secara rata-rata tingkat *output* tertinggi untuk semua jenis skenario terdapat pada mesin *carousel* yang terdiri dari 17 unit mesin yaitu sebanyak rata-rata 50777 unit tabung gas 3 kg dalam satu pekan. Peringkat kedua terdapat pada mesin *carousel* yang terdiri dari 13 unit mesin yaitu sebanyak rata-rata 50460 unit tabung gas 3 kg dalam satu pekan. Peringkat ketiga terdapat pada mesin *carousel* yang terdiri dari 16 unit mesin yaitu sebanyak rata-rata 50086 unit tabung gas 3 kg dalam satu pekan. Dan terpaut tipis dengan peringkat tiga, yaitu peringkat empat terdapat pada mesin *carousel* yang terdiri dari 18 unit mesin yaitu sebanyak rata-rata 50029 unit tabung gas 3 kg dalam satu pekan.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Tujuan dari penelitian ini adalah membuat model simulasi sistem SPBE standar Pertamina guna mengetahui penggunaan sumberdaya yang menghasilkan tingkat *output* tertinggi dalam menghadapi fluktuasi permintaan tabung dan pasokan gas.

1. Telah berhasil merancang model simulasi SPBE standar Pertamina dengan menggunakan ProModel yang telah lulus uji verifikasi dan validasi dengan asumsi-asumsi yaitu waktu kerja (operasional) di SPBE ialah jam 06.00-18.00 dengan dua kali waktu istirahat yaitu jam 12.00-13.00 dan jam 14.45-15.15; menggunakan 12 unit mesin dalam satu *carousel*, tiap unit truk agen membawa 230 unit tabung gas 3 kg, dan tiap unit truk tangki (*skid tank*) membawa 9000 kg *bulk* elpiji. Sebagai tambahan, sebelum simulasi berjalan telah ditentukan bahwa SPBE memiliki *buffer* tabung gas 3 kg kosong sebanyak 500 unit, *buffer* tabung gas 3 kg isi sebanyak 100 unit, dan *buffer* gas di *storage tank* sebanyak 9000 kg; menghasilkan *output* 1200 unit tabung gas 3 kg setiap jam dengan distribusi probabilitas tiap proses sebagai berikut.

Tabel 5.1 Kondisi Awal Model Simulasi SPBE Standar Pertamina

No.	Nama Proses	Distribusi Probabilitas yang sesuai
1	Tingkat kedatangan <i>skid tank</i>	Poisson (1.1) HR
2	Pengisian <i>storage tank</i>	Normal (25., 5.) MIN
3	Tingkat kedatangan agen	Poisson (9.5) MIN
4	Pembongkaran tabung	Triangular (3.83, 5.65, 9.10) MIN
5	<i>Check Weight</i>	Uniform (0.8, 1.2) SEC
6	Pengisian tabung dari <i>carousel</i>	2.5 SEC
7	<i>Check Scale</i>	Triangular (0.732, 1., 1.16) SEC
8	<i>Correction Scale</i>	Lognormal (0.286, 0.171) SEC
9	<i>Leak Tester</i>	Normal (1.71, 0.151) SEC
10	Penyegelan tutup tabung	Normal (0.413, 0.008) SEC
11	Pemuatan tabung	Triangular (7., 10.1, 11.7) MIN
12	Evakuasi	Triangular (0.921, 1.4, 1.52) SEC

2. Skenario dimana tingkat kedatangan truk agen berlangsung secara normal dan tingkat kedatangan *skid tank* juga berlangsung secara normal maka mesin *carousel* yang terdiri atas 13 unit mesin menghasilkan tingkat *output* tertinggi yaitu sebanyak rata-rata 54000 unit tabung gas 3 kg dalam satu pekan, sebanyak rata-rata 236 unit truk agen yang terlayani di SPBE selama satu pekan, sebanyak rata-rata 17 unit *skid tank* yang melakukan pengisian ke *storage tank* di SPBE selama satu pekan, sebanyak rata-rata

23000 kg dalam satu hari, dan memiliki tingkat utilitas terendah yaitu sebanyak rata-rata 90% utilisasi mesin *carousel* selama satu pekan.

3. Skenario dimana tingkat kedatangan truk agen berlangsung secara normal dan tingkat kedatangan *skid tank* berlangsung secara ekstrim maka mesin *carousel* yang terdiri atas 17 dan 18 unit mesin menghasilkan tingkat *output* tertinggi yaitu sebanyak rata-rata 50572 unit tabung gas 3 kg dalam satu pekan, sebanyak rata-rata 220 unit truk agen yang terlayani di SPBE selama satu pekan, sebanyak rata-rata 16 unit *skid tank* yang melakukan pengisian ke *storage tank* di SPBE selama satu pekan, sebanyak rata-rata 21674 kg dalam satu hari, dan memiliki tingkat utilitas terendah yaitu sebanyak rata-rata 89,22% utilisasi mesin *carousel* selama satu pekan.
4. Skenario dimana tingkat kedatangan truk agen berlangsung secara ekstrim dan tingkat kedatangan *skid tank* berlangsung secara normal maka mesin *carousel* yang terdiri atas 17 unit mesin menghasilkan tingkat *output* tertinggi yaitu sebanyak rata-rata 53987 unit tabung gas 3 kg dalam satu pekan, sebanyak rata-rata 236 unit truk agen yang terlayani di SPBE selama satu pekan, sebanyak rata-rata 17 unit *skid tank* yang melakukan pengisian ke *storage tank* di SPBE selama satu pekan, sebanyak rata-rata 23137 kg dalam satu hari, dan memiliki tingkat utilitas yaitu sebanyak rata-rata 89,14% utilisasi mesin *carousel* selama satu pekan.
5. Skenario dimana tingkat kedatangan truk agen berlangsung secara ekstrim dan tingkat kedatangan *skid tank* juga berlangsung secara ekstrim maka mesin *carousel* yang terdiri atas 16 unit mesin menghasilkan tingkat *output* tertinggi yaitu sebanyak rata-rata 50567 unit tabung gas 3 kg dalam satu pekan, sebanyak rata-rata 220 unit truk agen yang terlayani di SPBE selama satu pekan, sebanyak rata-rata 16 unit *skid tank* yang melakukan pengisian ke *storage tank* di SPBE selama satu pekan, sebanyak rata-rata 21672 kg dalam satu hari, dan memiliki tingkat utilitas yaitu sebanyak rata-rata 89% utilisasi mesin *carousel* selama satu pekan.
6. Secara rata-rata tingkat *output* tertinggi untuk semua jenis skenario terdapat pada mesin *carousel* yang terdiri dari 17 unit mesin yaitu sebanyak rata-rata 50777 unit tabung gas 3 kg dalam satu pekan. Peringkat

kedua terdapat pada mesin *carousel* yang terdiri dari 13 unit mesin yaitu sebanyak rata-rata 50460 unit tabung gas 3 kg dalam satu pekan. Peringkat ketiga terdapat pada mesin *carousel* yang terdiri dari 16 unit mesin yaitu sebanyak rata-rata 50086 unit tabung gas 3 kg dalam satu pekan. Dan terpaut tipis dengan peringkat tiga, yaitu peringkat empat terdapat pada mesin *carousel* yang terdiri dari 18 unit mesin yaitu sebanyak rata-rata 50029 unit tabung gas 3 kg dalam satu pekan.

7. Setelah merancang model simulasi SPBE standar Pertamina maka *output* yang dihasilkan dapat melebihi 30 ton per hari. Akan tetapi, *output* tersebut tidak dapat terpenuhi pada saat tingkat kedatangan truk agen memiliki intensitas tinggi dalam waktu antar-kedatangan yang berdekatan karena keterbatasan halaman antrian truk agen yang hanya dapat menampung 8 unit truk agen dalam waktu bersamaan. *Output* tersebut juga tidak dapat terpenuhi pada saat tingkat kedatangan truk tangki (*skid tank*) memiliki intensitas rendah dalam waktu antar-kedatangan yang berjauhan.

5.2 Saran

1. Penambahan kapasitas halaman antrian truk agen perlu dilakukan jika jumlah truk agen yang datang melebihi kapasitas pada waktu yang bersamaan karena kapasitas halaman antrian truk agen (model berdasarkan ukuran *layout* sebenarnya) yang berukuran 16 meter x 40 meter hanya dapat menampung 8 unit truk agen dalam waktu yang bersamaan. Sedangkan jika ada 16 unit truk agen yang mengantri dalam waktu yang bersamaan maka dibutuhkan halaman antrian truk agen berukuran 32 meter x 40 meter. Investasi penambahan lahan parkir truk agen akan mempengaruhi perhitungan ROR (*Rate Of Return*) atau ROI (*Return On Investment*) atau IRR (*Internal Rate of Return*) atau PI (*Profitability Index*).
2. Perlu standarisasi tingkat utilisasi mesin *carousel* yang aman sehingga ada waktu untuk perawatan agar dapat memperpanjang usia produktif mesin *carousel* yang akan mengurangi biaya perbaikan dan penggantian.

DAFTAR PUSTAKA

Anderson, T.W. dan D.A. Darling. *A Test of Goodness of Fit*. J. Am.Stat.Assoc. 1954: h 765.

A.Taha, Hamdy. *Operation Research: An Introduction*. Pearson Education, Inc. 2003

Boediono dan Wayan Koster. *Teori dan Aplikasi Statistika dan Probabilitas*. Bandung: Remaja Rosdakarya. 2002

Bronson, Richard. *Theory and Problems of Operation Research*. McGraw-Hill, Inc. 1993: h 308

Brooks, R. J. dan A. M. Tobias. *Choosing the best model: Level of detail, complexity and model performance*. Mathematical and Computer Modelling 24 (4): 1996: h 1-14.

Brunk, H.D.. *An Introduction to Mathematical Statistics*, Ginn&Co. 1960: h 261.

Chwif, L.. *Discrete event simulation model reduction: A causal approach*. Simulation Modelling Practice and Theory 14: 2006: h 930–944.

Feldmann, Clarence G.. *The Practical Guide to Business Process Reengineering Using IDEF0*, Dorset House Publishing Co., Inc. 1998. h 63.

Fishwick, P.A.. *Computer simulation: growth through extension*, Transactions of the Society for Computer Simulation International 14. 1997: h 13–23.

Guo, Y.. *SimOpt: A new simulation optimization system based virtual simulation for manufacturing system*. Simulation Modelling Practice and Theory 14: 2006: h 577–585.

Harrell, Charles, B. K. Ghosh, dan R. Bowden. 2000. *Simulation Using ProModel*, ed. ke-3. Boston: McGraw-Hill.

Kosturiak, J. dan M. Gregor. *FMS simulation: some experience and recommendations*. *Simulation Practice and Theory* 6. 1998: h 423–442.

Niebel, Benjamin dan Andris Freivalds. *Methods, Standards, and Work Design*. ed. ke-11. McGraw Hill. h 393-340.

Pujo, P. *Formal DEVS modelling and simulation of a flow-shop relocation method without interrupting the production*. *Simulation Modelling Practice and Theory* 14. 2006: h 817–842.

Robinson, S.. *Distributed Simulation and Simulation Practice*. *SIMULATION*, Vol. 81, Issue 1, January 2005 h 5-13 (2005) The Society for Modeling and Simulation International.

Stuart, Alan dan J. Keith Ord. *Kendall's Advanced Theory of Statistics*. Volume 2, Oxford University Press. 1991: h 1159.

Willemain, T. R .. *Insights on modeling from a dozen experts*. *Operations Research* 42 (2): 1994: h 213-22.

LAMPIRAN

Lampiran 3.4 Uji Penentuan Kesesuaian Distribusi

1. Waktu Antar Kedatangan *Skid Tank*

Auto::Fit of Distributions

distribution	rank	acceptance
Poisson(1.1)	100	do not reject

2. Proses Pengisian *Storage Tank*

Auto::Fit of Distributions

distribution	rank	acceptance
Normal(29.9, 0.79)	94.3	do not reject
Exponential(29., 0.9)	0.75	reject
Triangular(29., 31.7, 29.)	1.31e-002	reject
Lognormal(29., 0.292, 0.342)	1.42e-006	reject
Uniform(29., 31.)	0.	reject

3. Waktu Antar Kedatangan Truk Agen

Auto::Fit of Distributions

distribution	rank	acceptance
Binomial(263, 3.61e-002)	100	do not reject
Poisson(9.5)	98.5	do not reject

4. Proses Pembongkaran Tabung Kosong

Auto::Fit of Distributions

distribution	rank	acceptance
Triangular(7., 11.7, 9.8)	100	do not reject
Uniform(8., 11.)	10.6	do not reject
Lognormal(8., 0.588, 0.422)	6.23e-002	reject

5. Proses Penimbangan Tabung

Auto::Fit of Distributions

distribution	rank	acceptance
Exponential(0.8, 0.167)	36.6	reject
Uniform(0.8, 1.2)	23.	do not reject
Lognormal(0.8, -1.62, 0.458)	0.475	reject
Triangular(-0.2, 1.25, 1.2)	0.	reject

6. Proses Pemeriksaan Isi Tabung

Auto::Fit of Distributions

distribution	rank	acceptance
Lognormal(-291, 5.68, 3.37e-004)	69.8	reject
Triangular(0.732, 1.16, 1.)	57.2	do not reject
Normal(0.963, 9.83e-002)	31.9	reject
Uniform(0.8, 1.1)	5.51	reject

7. Proses Koreksi Isi Tabung

Auto::Fit of Distributions

distribution	rank	acceptance
Lognormal(0.286, -0.105, 0.171)	78.1	do not reject
Normal(1.2, 0.155)	48.2	do not reject
Exponential(1., 0.2)	1.99	reject
Triangular(1., 1.59, 1.)	0.488	reject
Uniform(1., 1.5)	0.193	reject

8. Proses Uji Kebocoran Tabung

Auto::Fit of Distributions

distribution	rank	acceptance
Lognormal(-290, 5.68, 5.18e-004)	99.9	do not reject
Normal(1.71, 0.151)	62.	do not reject
Triangular(1.33, 2.04, 1.8)	33.8	do not reject
Uniform(1.4, 2.)	25.2	do not reject

9. Proses Segel Tutup Tabung

Auto::Fit of Distributions

distribution	rank	acceptance
Lognormal[-2.19, 0.954, 3.4e-002]	99.5	do not reject
Normal(0.413, 8.84e-002)	32.5	do not reject
Exponential(0.3, 0.113)	3.87	reject
Triangular(0.3, 0.633, 0.3)	0.415	reject
Uniform(0.3, 0.6)	0.127	reject

10. Proses Pemuatan Tabung Isi

Auto::Fit of Distributions

distribution	rank	acceptance
Triangular(7., 11.7, 10.1)	100	do not reject
Uniform(8., 11.)	4.21	do not reject
Lognormal(8., 0.609, 0.425)	0.154	reject

11. Proses Evakuasi Isi Tabung

Auto::Fit of Distributions

distribution	rank	acceptance
Triangular(0.921, 1.52, 1.4)	62.2	do not reject
Lognormal[-467, 6.15, 2.78e-004]	61.	do not reject
Normal(1.28, 0.13)	26.3	do not reject
Uniform(1., 1.5)	0.842	reject

12. Waktu Antar Kedatangan *Skid Tank* Ekstrim

Auto::Fit of Distributions

distribution	rank	acceptance
Weibull[6.37e-002, 1.57, 4.4]	99.6	do not reject
Johnson SB[-4.72e-002, 11.7, 0.679, 0.855]	99.1	do not reject
Chi Squared[-1.91e-002, 4.09]	91.2	do not reject
Gamma[-0.166, 2.41, 1.74]	79.	do not reject
Erlang[-0.166, 2., 2.09]	78.7	do not reject
Triangular[-0.1, 11.2, 1.19]	76.2	do not reject
Inverse Gaussian[-1.96, 29.7, 5.98]	68.	do not reject
Lognormal[-1.91, 1.69, 0.435]	66.4	do not reject
Pearson 5[-4.09, 11., 81.6]	63.9	do not reject
Inverse Weibull[-161, 82.3, 6.09e-003]	62.6	do not reject
Beta[0.178, 12., 1.36, 2.77]	61.3	do not reject
LogLogistic[-1.18, 3.3, 4.66]	52.9	do not reject
Extreme Value IA[2.84, 2.]	48.8	do not reject
Rayleigh[-0.434, 3.62]	25.8	do not reject
Logistic[3.81, 1.47]	6.92	reject
Normal[4.02, 2.52]	1.54	do not reject
Exponential[0.178, 3.84]	0.265	reject
Pearson 6[0.178, 2.e+004, 0.916, 4.86e+003]	5.5e-002	reject
Extreme Value IB[5.34, 2.69]	3.32e-002	reject
Power Function[0.177, 10.9, 0.728]	6.86e-003	reject
Pareto[0.178, 0.349]	0.	reject
Uniform[0.178, 10.7]	0.	reject

13. Waktu Antar Kedatangan Truk Agen Ekstrim

Auto::Fit of Distributions

distribution	rank	acceptance
Weibull(9.55e-002, 1.57, 6.59)	99.6	do not reject
Johnson SB(-7.08e-002, 17.5, 0.679, 0.855)	99.1	do not reject
Gamma(-0.248, 2.41, 2.61)	79.	do not reject
Erlang(-0.248, 2., 3.14)	78.7	do not reject
Pearson 6(0.266, 5.04e+003, 1.96, 1.7e+003)	78.3	do not reject
Triangular(-0.15, 16.8, 1.78)	76.2	do not reject
Inverse Gaussian(-2.94, 44.6, 8.97)	68.	do not reject
Lognormal(-2.87, 2.09, 0.435)	66.4	do not reject
Pearson 5(-6.07, 10.9, 120)	64.5	do not reject
Chi Squared(-1.03, 6.99)	64.	do not reject
Beta(0.266, 16.1, 1.3, 2.25)	57.6	do not reject
LogLogistic(-1.76, 3.3, 6.98)	52.9	do not reject
Extreme Value IA(4.26, 3.)	48.1	do not reject
Rayleigh(-0.651, 5.42)	25.8	do not reject
Logistic(5.77, 2.18)	5.1	do not reject
Normal(6.03, 3.77)	1.54	do not reject
Exponential(0.266, 5.76)	0.267	reject
Extreme Value IB(8.01, 4.04)	3.52e-002	reject
Power Function(0.265, 16.3, 0.728)	1.11e-003	reject
Pareto(0.266, 0.349)	0.	reject
Uniform(0.266, 16.1)	0.	reject
Inverse Weibull	no fit	reject

Lampiran 4.1 Formulasi Model

```
*****
*
*           Formatted Listing of Model:           *
*   C:\Program Files\ProModel\Models\SPBE scenarios 1 week.mod   *
*
*****
```

```
Time Units:           Seconds
Distance Units:       Meters
Initialization Logic: Tank_SetLevel (storagetank, 9000)
                    activate Set_View
Termination Logic:   // Updates Tank1 statistics to account for final state and contents
                    int Tank_StatIndex = 1
                    while Tank_StatIndex <= 100 do
                    {
                        Tank_UpdateStats(Tank_StatIndex)
                        Tank_SetState(Tank_StatIndex, Tank_Idle)
                        Inc Tank_StatIndex
                    }
```

```
*****
*           Locations           *
*****
```

Name	Cap	Units	Stats	Rules	Cost
Loc1	1	1	None	Oldest, ,	
Loc2	1	1	None	Oldest, ,	
Loc3	1	1	None	Oldest, ,	
Loc4	1	1	None	Oldest, ,	
Loc5	1	1	None	Oldest, ,	
Loc6	1	1	Time Series	Oldest, ,	
Loc7	1	1	None	Oldest, ,	
Loc8	1	1	None	Oldest, ,	
Loc9	1	1	None	Oldest, ,	
Loc10	1	1	None	Oldest, ,	
Loc11	1	1	None	Oldest, ,	
Loc12	1	1	None	Oldest, ,	
Loc13	1	1	None	Oldest, ,	
Loc14	1	1	None	Oldest, ,	
Loc15	1	1	None	Oldest, ,	
Loc16	1	1	None	Oldest, ,	
Masuk_3kg	1	1	Time Series	Oldest, ,	
Bongkar3_1	1	1	Time Series	Oldest, ,	
Bongkar3_2		INFINITE	1	Time Series	Oldest, ,
bayangan3_masuk		INFINITE	1	Time Series	Oldest, ,
Persediaan_tabung3kg_kosong		INFINITE	1	Time Series	Oldest, ,
T3_1	67	1	Time Series	Oldest, FIFO,	
Checkweight	1	1	Time Series	Oldest, ,	
T3_2	57	1	Time Series	Oldest, FIFO,	
F3_1	1	1	Time Series	Oldest, ,	
F3_2	1	1	Time Series	Oldest, ,	
F3_3	1	1	Time Series	Oldest, ,	
F3_4	1	1	Time Series	Oldest, ,	
F3_5	1	1	Time Series	Oldest, ,	
F3_6	1	1	Time Series	Oldest, ,	
F3_7	1	1	Time Series	Oldest, ,	
F3_8	1	1	Time Series	Oldest, ,	
F3_9	1	1	Time Series	Oldest, ,	
F3_10	1	1	Time Series	Oldest, ,	
F3_11	1	1	Time Series	Oldest, ,	
F3_12	1	1	Time Series	Oldest, , First	
T3_3	30	1	Time Series	Oldest, FIFO,	
Checkscale	1	1	Time Series	Oldest, ,	
T3_4	24	1	Time Series	Oldest, FIFO,	
T3_5	41	1	Time Series	Oldest, FIFO,	
Correctionscale	1	1	Time Series	Oldest, ,	
T3_6	38	1	Time Series	Oldest, FIFO,	
Leaktester	1	1	Time Series	Oldest, ,	
T3_7	45	1	Time Series	Oldest, FIFO,	
T3_8	49	1	Time Series	Oldest, FIFO,	
Sealing	1	1	Time Series	Oldest, ,	
T3_9	50	1	Time Series	Oldest, FIFO,	
Persediaan_tabung3kg_isi		INFINITE	1	Time Series	Oldest, ,
dari_3		INFINITE	1	Time Series	Oldest, ,
bayangan_3		INFINITE	1	Time Series	Oldest, ,
Join_3		INFINITE	1	Time Series	Oldest, ,
parkir3_keluar	1	1	Time Series	Oldest, ,	
Loc17	1	1	Time Series	Oldest, ,	
Loc18	1	1	Time Series	Oldest, ,	
Tabung3_siap_evakuasi		INFINITE	1	Time Series	Oldest, ,
Evakuasi3kg	1	1	Time Series	Oldest, ,	
Area_tumpuk_tabung3kg_bocor		INFINITE	1	None	Oldest, ,
storagetank	9000	1	Time Series	Oldest, , First	
Loc19	1	1	None	Oldest, ,	
Loc20	1	1	None	Oldest, ,	
Loc21	1	1	None	Oldest, ,	
Pipe1	50	1	Time Series	Oldest, ,	
Tank1	9000	1	Time Series	Oldest, ,	
Tank2	100	1	Time Series	Oldest, ,	
Pipe2	50	1	Time Series	Oldest, ,	
Tank3	100	1	Time Series	Oldest, ,	


```

Pipe3      50  1  Time Series Oldest, ,
Tank5     100 1  Time Series Oldest, ,
Pipe5     50  1  Time Series Oldest, ,
Tank4     100 1  Time Series Oldest, ,
Pipe4     50  1  Time Series Oldest, ,
Loc22     1   1  None      Oldest, ,

```

```

*****
*                               *
*                               *
*****

```

Entities

```

Name      Speed (mpm)  Stats      Cost
-----
skidtank_kiri  50      Time Series
Tabung3kg     25      Time Series
TrukAgen_kanan  50      Time Series
Tabung3kg_over  25      Time Series
Tabung3kg_under 25      Time Series
TrukAgen_isi  50      Time Series

```

```

*****
*                               *
*                               *
*****

```

Path Networks

```

Name      Type      T/S      From  To  BI  Dist/Time  Speed Factor
-----
Net1      Passing  Speed & Distance N1  N2  Bi  80.14  1
Net2      Passing  Speed & Distance N1  N2  Bi  32.42  1
Net3      Passing  Speed & Distance N1  N2  Bi  36.42  1
Net4      Passing  Speed & Distance N1  N2  Bi  69.85  1
Net5      Passing  Speed & Distance N1  N2  Bi  32.42  1
Net6      Passing  Speed & Distance N1  N2  Bi  34.87  1
Net7      Passing  Speed & Distance N1  N2  Bi  35.57  1
Net8      Passing  Speed & Distance N1  N2  Bi  67.87  1
Net9      Passing  Speed & Distance N1  N2  Bi  45.19  1
Net10     Passing  Speed & Distance N1  N2  Bi  59.71  1
Net11     Passing  Speed & Distance N1  N2  Bi  51.28  1
Net12     Passing  Speed & Distance N1  N2  Bi  71.14  1
Net13     Passing  Speed & Distance N1  N2  Bi  97.71  1
Net14     Passing  Speed & Distance N1  N2  Bi  29.00  1
Net15     Passing  Speed & Distance N1  N2  Bi  32.14  1
Net21     Passing  Speed & Distance N1  N2  Bi  28.57  1
jalur3    Passing  Speed & Distance N1  N2  Bi  17.03  1
jalur_mesin_3kg  Passing  Speed & Distance N1  N2  Bi  6.46  1
          N1  N3  Bi  5.16  1
          N1  N4  Bi  5.20  1
          N1  N5  Bi  5.15  1
          N1  N6  Bi  4.66  1
          N1  N7  Bi  4.46  1
          N1  N8  Bi  4.37  1
          N1  N9  Bi  3.97  1
          N1  N10 Bi  4.06  1
          N1  N11 Bi  4.75  1
          N1  N12 Bi  5.30  1
          N1  N13 Bi  5.91  1
          N2  N3  Bi  2.30  1
          N3  N4  Bi  2.14  1
          N4  N5  Bi  2.74  1
          N5  N6  Bi  1.35  1
          N6  N7  Bi  2.16  1
          N7  N8  Bi  2.23  1
          N8  N9  Bi  2.62  1
          N9  N10 Bi  2.58  1
          N10 N11 Bi  2.57  1
          N11 N12 Bi  3.39  1
          N12 N13 Bi  3.50  1
          N13 N2  Bi  1.98  1
keluar3   Passing  Speed & Distance N1  N2  Bi  13.28  1
Net16     Passing  Speed & Distance N1  N2  Bi  39.42  1
Net17     Passing  Speed & Distance N1  N2  Bi  102.14  1
jalur_evakuasi3kg_keluar  Passing  Speed & Distance N1  N2  Bi  6.46  1
Net18     Passing  Speed & Distance N1  N2  Bi  27.14  1
Net19     Passing  Speed & Distance N1  N2  Bi  25.14  1
Net20     Passing  Speed & Distance N1  N2  Bi  26.71  1
unload_ke_load  Passing  Speed & Distance N1  N2  Bi  36.00  1
dari_3_ke_bayangan_3  Passing  Speed & Distance N1  N2  Bi  7.00  1
bayangan_3_ke_join_3  Passing  Speed & Distance N1  N2  Bi  20.42  1
jalur_truk_3  Passing  Speed & Distance N1  N2  Bi  23.85  1

```

```

*****
*                               *
*                               *
*****

```

Interfaces

```

Net      Node      Location
-----
Net1     N1      Loc1
          N2      Loc2
Net2     N1      Loc2
          N2      Loc3
Net3     N1      Loc3
          N2      Loc4
Net4     N1      Loc4
          N2      Loc5
Net5     N1      Loc5

```

```

Net6      N2   Loc6
          N1   Loc6
          N2   Loc7
Net7      N1   Loc7
          N2   Loc8
Net8      N1   Loc8
          N2   Loc9
Net9      N1   Loc9
          N2   Loc10
Net10     N1   Loc10
          N2   Loc11
Net11     N1   Loc11
          N2   Loc12
Net12     N1   Loc12
          N2   Loc13
Net13     N1   Loc14
          N2   Loc15
Net14     N1   Loc15
          N2   Loc16
Net15     N1   Loc16
          N2   Loc19
Net21     N1   Masuk_3kg
          N2   Loc22
jalur3    N1   Bongkar3_2
          N2   bayangan3_masuk
jalur_mesin_3kg  N2   F3_1
          N3   F3_2
          N4   F3_3
          N5   F3_4
          N6   F3_5
          N7   F3_6
          N8   F3_7
          N9   F3_8
          N10  F3_9
          N11  F3_10
          N12  F3_11
          N13  F3_12
keluar3   N1   Join_3
          N2   parkir3_keluar
Net16     N1   parkir3_keluar
          N2   Loc17
Net17     N1   Loc17
          N2   Loc18
jalur_evakuasi3kg_keluar N1   Evakuasi3kg
          N2   Area_tumpuk_tabung3kg_bocor
Net18     N1   Loc19
          N2   Loc21
Net19     N1   Loc20
          N2   Masuk_3kg
Net20     N1   Loc21
          N2   Loc20
unload_ke_load  N1   Bongkar3_1
          N2   Join_3
dari_3_ke_bayangan_3  N1   dari_3
          N2   bayangan_3
bayangan_3_ke_join_3  N1   bayangan_3
          N2   Join_3
jalur_truk_3  N1   Loc22
          N2   Bongkar3_1

```

```

*****
*                               *
*                               *
*****

```

```

Net      From  To  Dest
-----
jalur_mesin_3kg N2   N1
          N3   N1
          N4   N1
          N5   N1
          N6   N1
          N7   N1
          N8   N1
          N9   N1
          N10  N1
          N11  N1
          N12  N1
          N13  N1
          N1   N2
          N3   N2
          N4   N3
          N5   N4
          N6   N5
          N7   N6
          N11  N12
          N12  N13
          N13  N2
          N1   N3
          N2   N3
          N1   N4
          N3   N4
          N8   N7
          N1   N5
          N4   N5
          N9   N8
          N1   N6

```

```

N5  N6
N1  N7
N6  N7
N10 N9
N1  N8
N7  N8
N11 N10
N1  N9
N8  N9
N12 N11
N1  N10
N9  N10
N13 N12
N1  N11
N2  N13
N10 N11
N1  N12
N1  N13

```

```

*****
*           Resources           *
*****

```

Name	Units	Res Stats	Ent Search	Path	Motion	Cost
Mesin3_bantu	jumlah_mesin	By Unit	Closest	Oldest jalur_mesin_3kg	Home: N1 Full: 500 mpm	Empty: 500 mpm
Operator_load_2	2	By Unit	Closest	Oldest bayangan_3_ke_join_3	Home: N1 Full: 800 mpm (Return)	Empty: 800 mpm
Operator_load_1	1	By Unit	Closest	Oldest dari_3_ke_bayangan_3	Home: N1 Full: 800 mpm (Return)	Empty: 800 mpm
Operator_unload	2	By Unit	Closest	Oldest jalur3	Home: N1 Full: 500 mpm (Return)	Empty: 500 mpm
Operator_evakuasi	1	By Unit	Closest	Oldest jalur_evakuasi3kg_keluar	Home: N1 Full: 50 mpm (Return)	Empty: 50 mpm

```

*****
*           Processing           *
*****

```

Entity	Location	Process Operation	Blk	Output	Routing Destination	Rule	Move Logic
skidtank_kiri	Loc1		1	skidtank_kiri	Loc2	FIRST 1	MOVE ON Net1
skidtank_kiri	Loc2	graphic 7 wait 2 sec	1	skidtank_kiri	Loc3	FIRST 1	graphic 3 MOVE ON Net2
skidtank_kiri	Loc3	graphic 5 wait 2 sec	1	skidtank_kiri	Loc4	FIRST 1	graphic 2 MOVE ON Net3
skidtank_kiri	Loc4	graphic 5 wait 2 sec	1	skidtank_kiri	Loc5	FIRST 1	graphic 3 MOVE ON Net4
skidtank_kiri	Loc5	graphic 3	1	skidtank_kiri	Loc6	FIRST 1	MOVE ON Net5
skidtank_kiri	Loc6	graphic 3 INC jumlah_skidtank, 1 Tank_Fill (Tank1, 9000, 300, 9000) Tank_SetLevel (Pipe1, 50) Tank_Transfer (Tank1, storagetank, 9000, 5, 0, 9000) Tank_SetLevel (Pipe1, 0)					
			1	skidtank_kiri	Loc7	FIRST 1	MOVE ON Net6
skidtank_kiri	Loc7	graphic 5 wait 2 sec	1	skidtank_kiri	Loc8	FIRST 1	graphic 2 MOVE ON Net7
skidtank_kiri	Loc8	graphic 6 wait 2 sec	1	skidtank_kiri	Loc9	FIRST 1	graphic 4 MOVE ON Net8
skidtank_kiri	Loc9	graphic 4	1	skidtank_kiri	Loc10	FIRST 1	MOVE ON Net9

```

skidtank_kiri Loc10      graphic 8
wait 2 sec 1 skidtank_kiri Loc11      FIRST 1 graphic 1
MOVE ON Net10

skidtank_kiri Loc11      graphic 8
wait 2 sec 1 skidtank_kiri Loc12      FIRST 1 graphic 4
MOVE ON Net11

skidtank_kiri Loc12      graphic 6
wait 2 sec 1 skidtank_kiri Loc13      FIRST 1 graphic 2
MOVE ON Net12

skidtank_kiri Loc13      graphic 2      1 skidtank_kiri EXIT      FIRST 1
TrukAgen_kanan Loc14      1 TrukAgen_kanan Loc15      FIRST 1 MOVE ON Net13

TrukAgen_kanan Loc15      graphic 5
wait 2 sec 1 TrukAgen_kanan Loc16      FIRST 1 graphic 3
MOVE ON Net14

TrukAgen_kanan Loc16      INC jumlah_truk3kg_datang, 1
graphic 7
wait 2 sec 1 TrukAgen_kanan Loc19      FIRST 1 graphic 2
MOVE ON Net15

TrukAgen_kanan Loc19      graphic 2      1 TrukAgen_kanan Loc21      FIRST 1 MOVE ON Net18
TrukAgen_kanan Loc21      graphic 2      1 TrukAgen_kanan Loc20      FIRST 1 MOVE ON Net20
TrukAgen_kanan Loc20      graphic 7
wait 2 sec 1 TrukAgen_kanan Masuk_3kg      FIRST 1 graphic 3
MOVE ON Net19

TrukAgen_kanan Masuk_3kg      graphic 1
1 TrukAgen_kanan Loc22      FIRST 1 MOVE ON Net21

TrukAgen_kanan Loc22      graphic 1
IF jumlah_truk_tabung3kg = 1 THEN
{
WAIT UNTIL jumlah_truk_tabung3kg = 0
1 TrukAgen_kanan Bongkar3_1      FIRST 1 MOVE ON jalur_truk_3
}

TrukAgen_kanan Bongkar3_1      INC jumlah_truk_tabung3kg, 1
ORDER 230 Tabung3kg TO Bongkar3_2
graphic 1      1 TrukAgen_isi Join_3      FIRST 1 WAIT T(3.83, 5.65, 9.10)MIN
IF jumlah_truk3kg_muatkan = 1 THEN
{
WAIT UNTIL jumlah_truk3kg_muatkan = 0
}
DEC jumlah_truk_tabung3kg, 1
MOVE ON unload_ke_load

Tabung3kg Bongkar3_2      1 Tabung3kg bayangan3_masuk      FIRST 1 MOVE WITH Operator_unload THEN FREE
INC jumlah_tabung3kg_masuk, 1

Tabung3kg bayangan3_masuk      1 Tabung3kg Persediaan_tabung3kg_kosong FIRST 1
Tabung3kg Persediaan_tabung3kg_kosong      1 Tabung3kg T3_1      FIRST 1 DEC jumlah_tabung3kg_masuk, 1
Tabung3kg T3_1      1 Tabung3kg Checkweight      FIRST 1
Tabung3kg Checkweight      WAIT U(0.8, 1.2) SEC
1 Tabung3kg T3_2      FIRST 1
Tabung3kg T3_2      1 Tabung3kg F3_1      FIRST 1
Tabung3kg F3_1      INC M3_1, 1
GET Mesin3_bantu
DEC Resource3kg, 1
WAIT waktu_CFM
Tank_SetLevel (Pipe2, 50)
Tank_Transfer (storagetank, Tank2, 3, 1000, 0, 0)
Tank_SetLevel (Pipe2, 0)
Tank_Dec (Tank2, 3)
IF M3_12 > 0 THEN
SEND 1 Tabung3kg TO T3_3
IF M3_11 > 0 THEN
SEND 1 Tabung3kg TO F3_12
IF M3_10 > 0 THEN
SEND 1 Tabung3kg TO F3_11
IF M3_9 > 0 THEN
SEND 1 Tabung3kg TO F3_10
IF M3_8 > 0 THEN
SEND 1 Tabung3kg TO F3_9
IF M3_7 > 0 THEN
SEND 1 Tabung3kg TO F3_8
IF M3_6 > 0 THEN
SEND 1 Tabung3kg TO F3_7
IF M3_5 > 0 THEN
SEND 1 Tabung3kg TO F3_6
IF M3_4 > 0 THEN
SEND 1 Tabung3kg TO F3_5
IF M3_3 > 0 THEN
SEND 1 Tabung3kg TO F3_4
IF M3_2 > 0 THEN
SEND 1 Tabung3kg TO F3_3
IF M3_1 > 0 THEN
SEND 1 Tabung3kg TO F3_2
1 Tabung3kg F3_2      SEND 1 MOVE WITH Mesin3_bantu
DEC M3_1, 1

```

Tabung3kg	F3_2	INC M3_2, 1			
		1 Tabung3kg	F3_3	SEND 1	MOVE WITH Mesin3_bantu
Tabung3kg	F3_3	INC M3_3, 1		DEC M3_2, 1	
		1 Tabung3kg	F3_4	SEND 1	MOVE WITH Mesin3_bantu
Tabung3kg	F3_4	INC M3_4, 1		DEC M3_3, 1	
		1 Tabung3kg	F3_5	SEND 1	MOVE WITH Mesin3_bantu
Tabung3kg	F3_5	INC M3_5, 1		DEC M3_4, 1	
		1 Tabung3kg	F3_6	SEND 1	MOVE WITH Mesin3_bantu
Tabung3kg	F3_6	INC M3_6, 1		DEC M3_5, 1	
		1 Tabung3kg	F3_7	SEND 1	MOVE WITH Mesin3_bantu
Tabung3kg	F3_7	INC M3_7, 1		DEC M3_6, 1	
		1 Tabung3kg	F3_8	SEND 1	MOVE WITH Mesin3_bantu
Tabung3kg	F3_8	INC M3_8, 1		DEC M3_7, 1	
		1 Tabung3kg	F3_9	SEND 1	MOVE WITH Mesin3_bantu
Tabung3kg	F3_9	INC M3_9, 1		DEC M3_8, 1	
		1 Tabung3kg	F3_10	SEND 1	MOVE WITH Mesin3_bantu
Tabung3kg	F3_10	INC M3_10, 1		DEC M3_9, 1	
		1 Tabung3kg	F3_11	SEND 1	MOVE WITH Mesin3_bantu
Tabung3kg	F3_11	INC M3_11, 1		DEC M3_10, 1	
		1 Tabung3kg	F3_12	SEND 1	MOVE WITH Mesin3_bantu
Tabung3kg	F3_12	INC M3_12, 1		DEC M3_11, 1	
		1 Tabung3kg	T3_3	FIRST 1	FREE 1 Mesin3_bantu
				INC Resource3kg, 1	
				DEC M3_12, 1	
Tabung3kg	T3_3	1 Tabung3kg	Checksacle	FIRST 1	
Tabung3kg	Checksacle	WAIT T(0.732, 1, 1.16) SEC			
		1 Tabung3kg	T3_4	0.900000	1
		Tabung3kg_over	T3_5	0.050000	
		Tabung3kg_under	T3_5	0.050000	
Tabung3kg_over	T3_5	1 Tabung3kg_over	Correctionscale	FIRST 1	
Tabung3kg_under	T3_5	1 Tabung3kg_under	Correctionscale	FIRST 1	
Tabung3kg_over	Correctionscale	WAIT L(0.286, 0.171) SEC			
		Tank_Fill (Tank3, 1, 1000, 0)			
		Tank_SetLevel (Pipe3, 50)			
		Tank_Transfer (Tank3, storagetank, 1, 1000, 0, 0)			
		Tank_SetLevel (Pipe3, 0)			
		INC jumlah_Tabung3kg_over			
		1 Tabung3kg	T3_6	FIRST 1	
Tabung3kg_under	Correctionscale	WAIT L(0.286, 0.171) SEC			
		Tank_SetLevel (Pipe4, 50)			
		Tank_Transfer (storagetank, Tank4, 1, 1000, 0, 0)			
		Tank_SetLevel (Pipe4, 0)			
		Tank_Dec (Tank4, 1)			
		INC jumlah_Tabung3kg_under			
		1 Tabung3kg	T3_6	FIRST 1	
Tabung3kg	T3_6	1 Tabung3kg	Leaktester	FIRST 1	
Tabung3kg	T3_4	1 Tabung3kg	Leaktester	FIRST 1	
Tabung3kg	Leaktester	WAIT N(1.71, 0.151) SEC			
		1 Tabung3kg	T3_8	0.980000	1
		Tabung3kg	T3_7	0.020000	
Tabung3kg	T3_8	1 Tabung3kg	Sealing	FIRST 1	
Tabung3kg	Sealing	WAIT N(0.413, 0.008) SEC			
		1 Tabung3kg	T3_9	FIRST 1	
Tabung3kg	T3_9	1 Tabung3kg	Persediaan_tabung3kg_isi	FIRST 1	
Tabung3kg	Persediaan_tabung3kg_isi	INC jumlah_tabung3kg_keluar, 1			
		1 Tabung3kg	dari_3	FIRST 1	
Tabung3kg	dari_3	1 Tabung3kg	bayangan_3	FIRST 1	MOVE WITH Operator_load_1 THEN FREE
Tabung3kg	bayangan_3	1 Tabung3kg	Join_3	JOIN 1	MOVE WITH Operator_load_2 THEN FREE
TrukAgen_isi	Join_3	INC jumlah_truk3kg_muatkan, 1			
		JOIN 230 Tabung3kg			
		graphic 6			
		wait 2 sec			
		1 TrukAgen_isi	parkir3_keluar	FIRST 1	DEC jumlah_truk3kg_muatkan, 1
				graphic 4	
				MOVE ON keluar3	
TrukAgen_isi	parkir3_keluar	graphic 4			
		INC Jumlah_truk3kg_keluar, 1			

```

1 TrukAgen_isi Loc17 FIRST 1 MOVE ON Net16

TrukAgen_isi Loc17 graphic 8
wait 2 sec 1 TrukAgen_isi Loc18 FIRST 1 graphic 2
MOVE ON Net17

TrukAgen_isi Loc18 graphic 2 1 TrukAgen_isi EXIT FIRST 1
Tabung3kg T3_7 1 Tabung3kg Tabung3_siap_evakuasi FIRST 1

Tabung3kg Tabung3_siap_evakuasi IF jumlah_tabung_di_mesin_evakuasi3 = 1 THEN
{
WAIT UNTIL jumlah_tabung_di_mesin_evakuasi3 = 0
}

INC jumlah_tabung_di_mesin_evakuasi3, 1
1 Tabung3kg Evakuasi3kg FIRST 1
Tabung3kg Evakuasi3kg WAIT T(0.921, 1.4, 1.52) SEC
Tank_Fill (Tank5, 3, 500, 0)
Tank_SetLevel (Pipe5, 50)
Tank_Transfer (Tank5, storagetank, 3, 500, 0, 0)
Tank_SetLevel (Pipe5, 0)
DEC jumlah_tabung_di_mesin_evakuasi3, 1
1 Tabung3kg Area_tumpuk_tabung3kg_bocor FIRST 1 INC jumlah_tabung3kg_bocor, 1
MOVE WITH Operator_evakuasi THEN FREE

Tabung3kg Area_tumpuk_tabung3kg_bocor 1 Tabung3kg EXIT FIRST 1

```

```

*****
* Arrivals *
*****

```

Entity	Location	Qty Each	First Time	Occurrences	Frequency	Logic
skidtank_kiri	Loc1	1	Wk 1, Mon @ 06:00 AM	4		tangki_datang
skidtank_kiri	Loc1	1	Wk 1, Tue @ 06:00 AM	3		tangki_datang
skidtank_kiri	Loc1	1	Wk 1, Wed @ 06:00 AM	3		tangki_datang
skidtank_kiri	Loc1	1	Wk 1, Thu @ 06:00 AM	4		tangki_datang
skidtank_kiri	Loc1	1	Wk 1, Fri @ 06:00 AM	2		tangki_datang
skidtank_kiri	Loc1	1	Wk 1, Sat @ 06:00 AM	4		tangki_datang
skidtank_kiri	Loc1	1	Wk 1, Sun @ 06:00 AM	3		tangki_datang
TrukAgen_kanan	Loc14	1	Wk 1, Mon @ 06:00 AM	50		truk_datang
TrukAgen_kanan	Loc14	1	Wk 1, Tue @ 06:00 AM	45		truk_datang
TrukAgen_kanan	Loc14	1	Wk 1, Wed @ 06:00 AM	48		truk_datang
TrukAgen_kanan	Loc14	1	Wk 1, Thu @ 06:00 AM	42		truk_datang
TrukAgen_kanan	Loc14	1	Wk 1, Fri @ 06:00 AM	33		truk_datang
TrukAgen_kanan	Loc14	1	Wk 1, Sat @ 06:00 AM	48		truk_datang
TrukAgen_kanan	Loc14	1	Wk 1, Sun @ 06:00 AM	45		truk_datang
Tabung3kg	Persediaan_tabung3kg_kosong	500	Wk 1, Mon @ 06:00 AM	1		
Tabung3kg	Persediaan_tabung3kg_kosong	500	Wk 1, Tue @ 06:00 AM	1		
Tabung3kg	Persediaan_tabung3kg_kosong	500	Wk 1, Wed @ 06:00 AM	1		
Tabung3kg	Persediaan_tabung3kg_kosong	500	Wk 1, Thu @ 06:00 AM	1		
Tabung3kg	Persediaan_tabung3kg_kosong	500	Wk 1, Fri @ 06:00 AM	1		
Tabung3kg	Persediaan_tabung3kg_kosong	500	Wk 1, Sat @ 06:00 AM	1		
Tabung3kg	Persediaan_tabung3kg_kosong	500	Wk 1, Sun @ 06:00 AM	1		
Tabung3kg	bayangan_3	100	Wk 1, Mon @ 06:00 AM	1		
Tabung3kg	bayangan_3	100	Wk 1, Tue @ 06:00 AM	1		
Tabung3kg	bayangan_3	100	Wk 1, Wed @ 06:00 AM	1		
Tabung3kg	bayangan_3	100	Wk 1, Thu @ 06:00 AM	1		
Tabung3kg	bayangan_3	100	Wk 1, Fri @ 06:00 AM	1		
Tabung3kg	bayangan_3	100	Wk 1, Sat @ 06:00 AM	1		
Tabung3kg	bayangan_3	100	Wk 1, Sun @ 06:00 AM	1		

```

*****
* Shift Assignments *
*****

```

Locations...	Resources...	Shift Files...	Priorities...	Disable Logic...
Area_tumpuk_tabung3kg_bocor	Mesin3_bantu	C:\Program Files\ProModel\Shif	99,99,99,99	No
bayangan_3	Operator_evakuasi			
bayangan3_masuk	Operator_load_1			
Bongkar3_1	Operator_load_2			
Bongkar3_2	Operator_unload			
Checkscale				
Checkweight				
Correctionscale				
dari_3				
Evakuasi3kg				
F3_1				
F3_10				
F3_11				
F3_12				
F3_2				
F3_3				
F3_4				
F3_5				
F3_6				
F3_7				
F3_8				
F3_9				
Join_3				
Leaktester				
Loc1				

ID	Dimensions	Type
Tank_Level	100	Real
Tank_Statistics	100,14	Real
Tank_State	100	Integer
Tank_Product	100	Integer
Tank_Fills	100	Integer

 * Macros *

ID	Text
Tank_Idle	0 // state for an idle Tank1
Tank_Operation	1
Tank_Setup	2
Tank_Filling	3
Tank_Emptying	4
Tank_Blocked	5
Tank_Down	6
Tank_Loop	While 1 do
Tank_TimeStep	.2
Tank_Stop	-1.0
Tank_LongestIdle	1
Tank_LongestBlocked	2
Tank_InOrder	3
jumlah_mesin	12
mesin_nambah	1
waktu_CFM	2.5 SEC
tangki_datang	P(1.1) HR
truk_datang	P(9.5) MIN

 * Subroutines *

ID	Type	Parameter	Type	Logic
Set_View	None			VIEW "Awal"
				ANIMATE 70
				WAIT 1 MIN
				VIEW "skidtank_masuk"
				ANIMATE 70
				WAIT 2 MIN
				VIEW "skidtank mengisi"
				ANIMATE 70
				WAIT 4 MIN
				VIEW "isi_storagetank"
				ANIMATE 70
				WAIT 2 MIN
				VIEW "bongkar_tabung"
				ANIMATE 70
WAIT 3 MIN				
Tank_Fill	None	Tank_ID	Integer	// Fills a tank_ID by a quantity and rate.
				Tank_FillQty Real
				Tank_FillRate Real Real Tank_TimeInc
				Tank_ResumeLevel Real Real Tank_FillInc
				Real Tank_FillCapacity = Tank_Cap(Tank_ID)
				Int Tank_VarRate=0
				Real Tank_FillRequest = Tank_FillQty
				If Tank_FillRate <= 0.0
				Then Tank_VarRate = 1
				Else Tank_FillRate = Tank_FillRate * Tank_TimeStep // convert units per minute to units per timestep
				Wait Until Tank_State[Tank_ID] <> Tank_Down // Don't fill if Tank_ID is down


```

While Tank_FillQty > 0.0000001 do
{
  Tank_TimeInc = Tank_TimeStep
  If Tank_Level [Tank_ID] = Tank_FillCapacity then // Is Tank_ID already full?
  {
    If Tank_ResumeLevel = Tank_Stop then
    {
      Tank_QtyLeft = Tank_FillQty
      Return
    }
    If Tank_ResumeLevel = 0.0
    Then Stop "Attempted to fill " $Loc(Tank_ID) $ " beyond its capacity of " $Tank_Cap(Tank_ID)
    Else If Tank_ResumeLevel >= Tank_FillCapacity
    Then Stop "Resume Level in Tank_Fill subroutine for " $Loc(Tank_ID) $ " must be less than its capacity"
    Else Wait Until Tank_Level[Tank_ID] <= Tank_ResumeLevel // wait for drop to resume filling
  }
  if Tank_VarRate then
  {
    Tank_FillRate = Tank_Rate(0, Tank_ID) * Tank_TimeStep
    if Tank_FillRate = 0.0 then
    {
      Tank_QtyLeft = Tank_FillQty
      Return
    }
  }
  // Adjust rate for last remaining quantity
  If Tank_FillQty < Tank_FillRate then
  {
    Tank_TimeInc = Tank_TimeInc * Tank_FillQty / Tank_FillRate // proportionalize
    Tank_FillInc = Tank_FillQty
  }
  Else Tank_FillInc = Tank_FillRate
  If Tank_FillCapacity - Tank_Level[Tank_ID] < Tank_FillInc then // Test for insufficient cap BEFORE Wait
  {
    Tank_TimeInc = Tank_TimeInc * (Tank_FillCapacity - Tank_Level[Tank_ID]) / Tank_FillInc //
    Tank_FillInc = Tank_FillCapacity - Tank_Level[Tank_ID]
  }
  if Tank_State[Tank_ID] <> Tank_Filling
  then Tank_SetState(Tank_ID, Tank_Filling)
  Wait Tank_TimeInc
  Wait Until Tank_State[Tank_ID] <> Tank_Down // Don't fill if Tank_ID is down
  If Tank_FillCapacity - Tank_Level[Tank_ID] < Tank_FillInc // Test for insufficient cap AFTER Wait
  then Tank_FillInc = Tank_FillCapacity - Tank_Level[Tank_ID]
  Dec Tank_FillQty, Tank_FillInc
  Inc Tank_Level [Tank_ID], Tank_FillInc
  Tank_UpdateStats (Tank_ID)
}
// Adjust for floating point error
if Tank_Level[Tank_ID] < Tank_FillRequest
then if Tank_Level[Tank_ID] + .0000001 > Tank_FillRequest then
{
  Tank_Level[Tank_ID] = Tank_FillRequest
  Tank_UpdateStats (Tank_ID)
}
Tank_QtyLeft = 0.0
Inc Tank_Fills[Tank_ID]
}

Tank_Empty None Tank_ID Integer // Empties a Tank_ID by a quantity and rate.
Tank_EmptyQty Real
Tank_EmptyRate Real Real Tank_TimeInc
Tank_ResumeLevel Real Real Tank_EmptyInc
Int Tank_VarRate

If Tank_EmptyRate <= 0.0
Then Tank_VarRate = 1
Else Tank_EmptyRate = Tank_EmptyRate * Tank_TimeStep // convert units per minute to units per timestep
Wait Until Tank_State[Tank_ID] <> Tank_Down // Don't empty if Tank_ID is down
While Tank_EmptyQty > 0.0 do
{
  Tank_TimeInc = Tank_TimeStep
  If Tank_Level [Tank_ID] = 0.0 then // Is Tank_ID empty?
  {
    If Tank_ResumeLevel = Tank_Stop then
    {
      Tank_QtyLeft = Tank_EmptyQty
      Return
    }
    If Tank_ResumeLevel = 0.0
    Then Stop "Attempted to empty " $Loc(Tank_ID) $ " by more than its contents"
    Else If Tank_ResumeLevel > Tank_Cap(Tank_ID)
    Then Stop "Resume Level in Tank_Empty subroutine for " $Loc(Tank_ID) $ " must be less than or equal
    Else Wait Until Tank_Level[Tank_ID] >= Tank_ResumeLevel // wait for rise to resume filling
  }
  If Tank_VarRate then
  {
    Tank_EmptyRate = Tank_Rate(Tank_ID, 0) * Tank_TimeStep
    if Tank_EmptyRate = 0.0 then
    {
      Tank_QtyLeft = Tank_EmptyQty
      Return
    }
  }
  If Tank_EmptyQty < Tank_EmptyRate then // test for final partial amount
}

```

```

        {
            Tank_TimeInc = Tank_TimeInc * Tank_EmptyQty / Tank_EmptyRate //proportionalize
            Tank_EmptyInc = Tank_EmptyQty
        }
    Else Tank_EmptyInc = Tank_EmptyRate
    If Tank_Level [Tank_ID] < Tank_EmptyInc then // Test for insufficient qty BEFORE Wait
    {
        Tank_TimeInc = Tank_TimeInc * Tank_Level [Tank_ID] / Tank_EmptyInc // proportionalize
        Tank_EmptyInc = Tank_Level [Tank_ID]
    }
    if Tank_State[Tank_ID] <> Tank_Emptying
    then Tank_SetState(Tank_ID,Tank_Emptying)
    Wait Tank_TimeInc
    Wait Until Tank_State[Tank_ID] <> Tank_Down // Don't empty if Tank_ID is down
    If Tank_Level [Tank_ID] < Tank_EmptyInc then // Test for insufficient qty AFTER Wait
        Tank_EmptyInc = Tank_Level [Tank_ID]
    Dec Tank_EmptyQty, Tank_EmptyInc
    Dec Tank_Level [Tank_ID], Tank_EmptyInc
    Tank_UpdateStats (Tank_ID)
    }
    Tank_QtyLeft = 0.0
}

Tank_Transfer  None    Tank_FromID  Integer // Transfers a quantity from one Tank_ID to another.
               Tank_ToID  Integer
               Tank_TransferQty  Real    Real Tank_ToQty
               Tank_FromRate  Real    Real Tank_FromQty
               Tank_ToRate  Real    Real Tank_TimeInc
               Tank_ResumeLevel  Real    Real Tank_ToCapacity = Tank_Cap(Tank_ToID)

Int Tank_VarRate = 0
Wait Until Tank_State[Tank_FromID] <> Tank_Down // Don't transfer if FROM Tank_ID is down
If Tank_FromRate <= 0.0 // test for variable rate transfer
Then Tank_VarRate = 1 // used as a flag
Else
{
    Tank_FromRate = Tank_FromRate * Tank_TimeStep // convert units per clock unit to per timestep
    If Tank_ToRate = 0.0
    Then Tank_ToRate = Tank_FromRate
    Else Tank_ToRate = Tank_ToRate * Tank_TimeStep // convert units per clock unit to per timestep
}
While Tank_TransferQty > 0.0000001 do
{
    Tank_TimeInc = Tank_TimeStep
    // Make sure there is available TO capacity
    If Tank_Level [Tank_ToID] >= Tank_ToCapacity then
    {
        If Tank_ResumeLevel = Tank_Stop then
        {
            Tank_QtyLeft = Tank_TransferQty
            Return
        }
        If Tank_ResumeLevel = 0.0
        Then Stop "Attempted to fill "$Loc(Tank_ToID)$ " beyond its capacity of " $Tank_ToCapacity
        Else If Tank_ResumeLevel >= Tank_ToCapacity
        Then Stop "Resume Level in Tank_Transfer subroutine for "$Loc(Tank_FromID)$ " must be less than
its capacity"
    else Wait Until Tank_Level[Tank_ToID] <= Tank_ResumeLevel
    }
    If Tank_Level [Tank_FromID] <= 0.0 then Wait Until Tank_Level [Tank_FromID] > 0.0 // test for empty FROM
tank_ID

// For variable flow rate, call Tank_Rate subroutine to find rate
If Tank_VarRate then
{
    Tank_FromRate = Tank_Rate (Tank_FromID, Tank_ToID) * Tank_TimeStep // convert var units per
clock unit to per timestep

    if Tank_FromRate = 0.0 then
    {
        Tank_QtyLeft = Tank_TransferQty
        Return
    }
    Tank_ToRate = Tank_FromRate
}

// Test for final remaining quantity
If Tank_TransferQty < Tank_FromRate then
{
    Tank_FromQty = Tank_TransferQty // reduce to remaining quantity
    Tank_TimeInc = Tank_TimeInc * Tank_FromQty / Tank_FromRate // proportionalize
    Tank_ToQty = Tank_ToRate * Tank_FromQty / Tank_FromRate // proportionalize
}
Else
{
    Tank_FromQty = Tank_FromRate
    Tank_ToQty = Tank_ToRate
}

// Adjust flow rates based on FROM level BEFORE time step
If Tank_Level[Tank_FromID] < Tank_FromQty then
{
    Tank_TimeInc = Tank_TimeInc * Tank_Level[Tank_FromID] / Tank_FromQty // proportionalize
    Tank_ToQty = Tank_ToQty * Tank_Level[Tank_FromID] / Tank_FromQty // proportionalize
    Tank_FromQty = Tank_Level[Tank_FromID] // reduce to remaining quantity
}

// Further adjust flow rates based on TO level BEFORE time step
If Tank_ToCapacity - Tank_Level[Tank_ToID] < Tank_ToQty then

```

```

        {
            Tank_TimeInc = Tank_TimeInc * (Tank_ToCapacity - Tank_Level[Tank_ToID]) / Tank_ToQty
            Tank_FromQty = Tank_FromQty * (Tank_ToCapacity - Tank_Level[Tank_ToID]) / Tank_ToQty
            Tank_ToQty = Tank_ToCapacity - Tank_Level[Tank_ToID]
        }
        if Tank_State[Tank_FromID] <> Tank_Emptying
        then Tank_SetState(Tank_FromID, Tank_Emptying)
        if Tank_State[Tank_ToID] <> Tank_Filling
        then Tank_SetState(Tank_ToID, Tank_Filling)
        Wait Tank_TimeInc
        Wait Until Tank_State[Tank_FromID] <> Tank_Down // Don't transfer if FROM Tank_ID is down
        // Adjust rate based on FROM level after time step
        If Tank_Level[Tank_FromID] < Tank_FromQty then
        {
            Tank_ToQty = Tank_ToQty * Tank_Level[Tank_FromID] / Tank_FromQty // proportionalize
            Tank_FromQty = Tank_Level[Tank_FromID] // reduce to remaining qty
        }
        // Further adjust flow rates based on available TO capacity
        If Tank_ToCapacity - Tank_Level[Tank_ToID] < Tank_ToQty then
        {
            Tank_FromQty = Tank_FromQty * (Tank_ToCapacity - Tank_Level[Tank_ToID]) / Tank_ToQty //
            Tank_ToQty = Tank_ToCapacity - Tank_Level[Tank_ToID]
        }
        Dec Tank_Level [Tank_FromID], Tank_FromQty
        Tank_UpdateStats (Tank_FromID)
        Inc Tank_Level [Tank_ToID], Tank_ToQty
        Tank_UpdateStats (Tank_ToID)
        Dec Tank_TransferQty, Tank_FromQty
    }
    Tank_QtyLeft = 0.0
    Inc Tank_Fills[Tank_ToID]

Tank_TransferDownTo None Tank_FromID Integer // Transfers until FROM tank_ID drops to a specified level
Tank_ToID Integer
Tank_FromLevel Real if (Tank_FromLevel < 0.0) or (Tank_FromLevel > Tank_Cap(Tank_FromID))
Tank_FromRate Real then Stop ""$Tank_FromLevel $ " is not a valid level for "$Loc(Tank_FromID)$" in Tank_TransferDownTo
subroutine call"
Tank_ToRate Real if Tank_Level[Tank_FromID] <= Tank_FromLevel
then Stop "Level specified in Tank_TransferDownTo is greater than or equal to current level for "$loc(Tank_FromID)

Real Tank_ToQty
Real Tank_FromQty
Real Tank_TimeInc
Real Tank_ToCapacity = Tank_Cap(Tank_ToID)

Int Tank_VarRate = 0
Wait Until Tank_Level [Tank_FromID] > 0.0 // Don't start until there is at least something in From Tank_ID

Wait Until Tank_State[Tank_FromID] <> Tank_Down // Don't transfer if FROM Tank_ID is down
If Tank_FromRate <= 0.0 // test for variable rate transfer
Then Tank_VarRate = 1 // used as a flag
Else
{
    Tank_FromRate = Tank_FromRate * Tank_TimeStep // convert units per clock unit to per timestep
    If Tank_ToRate = 0.0
    Then Tank_ToRate = Tank_FromRate
    Else Tank_ToRate = Tank_ToRate * Tank_TimeStep //convert units per clock unit to per timestep
}
While Tank_Level[Tank_FromID] > Tank_FromLevel Do
{
    Tank_TimeInc = Tank_TimeStep
    // Make sure there is capacity in TO tank_ID before time step or an infinite loop will occur
    If Tank_Level[Tank_ToID] >= Tank_ToCapacity
    then Wait Until Tank_Level [Tank_ToID] < Tank_ToCapacity

    // For variable flow rate, call Tank_Rate subroutine to find rate
    If Tank_VarRate then
    {
        Tank_FromRate = Tank_Rate(Tank_FromID, Tank_ToID) * Tank_TimeStep // convert var units per
        if Tank_FromRate <= 0.0 then Stop "Tank_Rate must return a value greater than 0 in
        Tank_ToRate = Tank_FromRate
    }

    // Adjust flow rates based on FROM contents
    If Tank_Level[Tank_FromID] - Tank_FromLevel < Tank_FromRate then
    {
        Tank_FromQty = Tank_Level[Tank_FromID] - Tank_FromLevel // reduce to remaining qty
        Tank_TimeInc = Tank_TimeInc * Tank_FromQty / Tank_FromRate // proportionalize
        Tank_ToQty = Tank_ToRate * Tank_FromQty / Tank_FromRate // proportionalize
    }
    Else
    {
        Tank_FromQty = Tank_FromRate
        Tank_ToQty = Tank_ToRate
    }
    // Further adjust flow rates based on available TO capacity
    If Tank_ToCapacity - Tank_Level[Tank_ToID] < Tank_ToQty then
    {
        Tank_FromQty = Tank_FromQty * (Tank_ToCapacity - Tank_Level[Tank_ToID]) / Tank_ToQty
        Tank_TimeInc = Tank_TimeInc * (Tank_ToCapacity - Tank_Level[Tank_ToID]) / Tank_ToQty
        Tank_ToQty = Tank_ToCapacity - Tank_Level[Tank_ToID]
    }
}
if Tank_State[Tank_FromID] <> Tank_Emptying

```

```

then Tank_SetState(Tank_FromID, Tank_Emptying)
if Tank_State[Tank_ToID] <> Tank_Filling
then Tank_SetState(Tank_ToID, Tank_Filling)
Wait Tank_TimeInc
Wait Until Tank_State[Tank_FromID] <> Tank_Down // Don't transfer if FROM Tank_ID is down
// Adjust transfer qty based on FROM level AFTER time step
If Tank_Level[Tank_FromID] - Tank_FromLevel < Tank_FromQty then
{
    Tank_ToQty = Tank_ToQty * (Tank_Level[Tank_FromID] - Tank_FromLevel) / Tank_FromQty //
    Tank_FromQty = Tank_Level[Tank_FromID] - Tank_FromLevel // reduce to remaining qty
}
// Further adjust flow rates based on available TO capacity
If Tank_ToCapacity - Tank_Level[Tank_ToID] < Tank_ToQty then
{
    Tank_FromQty = Tank_FromQty * (Tank_ToCapacity - Tank_Level[Tank_ToID]) / Tank_ToQty //
    Tank_ToQty = Tank_ToCapacity - Tank_Level[Tank_ToID]
}
Dec Tank_Level [Tank_FromID], Tank_FromQty
Tank_UpdateStats (Tank_FromID)
Inc Tank_Level [Tank_ToID], Tank_ToQty
Tank_UpdateStats (Tank_ToID)
}

Tank_TransferUpTo None Tank_FromID Integer // Transfers until TO tank_ID rises to a specified level
Tank_ToID Integer
Tank_ToLevel Real if (Tank_ToLevel < 0.0) or (Tank_ToLevel > Tank_Cap(Tank_ToID))
Tank_FromRate Real then Stop ""$Tank_ToLevel $* is not a valid level for " $Loc(Tank_ToID)$" in Tank_TransferUpTo subroutine

call"
Tank_ToRate Real if Tank_Level[Tank_ToID] >= Tank_ToLevel
then Stop "Level specified in Tank_TransferUpTo is less than or equal to current level for "$Loc(Tank_ToID)

Real Tank_ToQty
Real Tank_FromQty
Real Tank_TimeInc

Int Tank_VarRate = 0
Wait Until Tank_State[Tank_FromID] <> Tank_Down // Don't transfer if FROM Tank_ID is down
If Tank_FromRate <= 0.0 // test for variable rate transfer
Then Tank_VarRate = 1 // used as a flag
Else
{
    Tank_FromRate = Tank_FromRate * Tank_TimeStep // convert units per clock unit to per timestep
    If Tank_ToRate = 0.0
    Then Tank_ToRate = Tank_FromRate
    Else Tank_ToRate = Tank_ToRate * Tank_TimeStep //convert units per clock unit to per timestep
}
While Tank_Level[Tank_ToID] < Tank_ToLevel Do
{
    Tank_TimeInc = Tank_TimeStep
    // Make sure there is enough From qty BEFORE time step or an infinite loop will occur
    If Tank_Level [Tank_FromID] <= 0.0
    then Wait Until Tank_Level [Tank_FromID] > 0.0

    // For variable flow rate, call Tank_Rate subroutine to find rate
    If Tank_VarRate then
    {
        Tank_FromRate = Tank_Rate(Tank_FromID, Tank_ToID) * Tank_TimeStep // convert var units per
        if Tank_FromRate <= 0.0 then Stop "Tank_Rate must return a value greater than 0 in
        Tank_ToRate = Tank_FromRate
    }

    // Adjust flow rates based on TO level BEFORE time step.
    If Tank_ToLevel - Tank_Level[Tank_ToID] < Tank_ToRate then
    {
        Tank_ToQty = Tank_ToLevel - Tank_Level[Tank_ToID] // reduce to remaining qty
        Tank_TimeInc = Tank_TimeInc * Tank_ToQty / Tank_ToRate // proportionalize
        Tank_FromRate = Tank_FromRate * Tank_ToQty / Tank_ToRate // proportionalize
    }
    Else
    {
        Tank_FromQty = Tank_FromRate
        Tank_ToQty = Tank_ToRate
    }

    // Further adjust flow rates based on available From qty and To level BEFORE time step
    If Tank_Level [Tank_FromID] < Tank_FromRate then // test for less than From qty
    {
        Tank_FromQty = Tank_Level [Tank_FromID]
        Tank_ToQty = Tank_ToRate * Tank_FromQty / Tank_FromRate
        Tank_TimeInc = Tank_TimeInc * Tank_FromQty / Tank_FromRate
    }
    if Tank_State[Tank_FromID] <> Tank_Emptying
    then Tank_SetState(Tank_FromID, Tank_Emptying)
    if Tank_State[Tank_ToID] <> Tank_Filling
    then Tank_SetState(Tank_ToID, Tank_Filling)
    Wait Tank_TimeInc
    Wait Until Tank_State[Tank_FromID] <> Tank_Down // Don't transfer if FROM Tank_ID is down

    // Adjust transfer qty based on levels AFTER time step
    If Tank_ToLevel - Tank_Level[Tank_ToID] < Tank_ToQty then
    {
        if Tank_Level[Tank_ToID] >= Tank_ToLevel then Return // terminate if already filled to level

```

```

proportionalize
    Tank_FromQty = Tank_FromQty * (Tank_ToLevel - Tank_Level[Tank_ToID]) / Tank_ToQty //
    Tank_ToQty = Tank_ToLevel - Tank_Level[Tank_ToID] // reduce to remaining qty
}
If Tank_Level [Tank_FromID] < Tank_FromQty then
{
    Tank_ToQty = Tank_ToQty * Tank_Level[Tank_FromID] / Tank_FromQty
    Tank_FromQty = Tank_Level [Tank_FromID]
}
Dec Tank_Level [Tank_FromID], Tank_FromQty
Tank_UpdateStats (Tank_FromID)
Inc Tank_Level [Tank_ToID], Tank_ToQty
Tank_UpdateStats (Tank_ToID)
}

Tank_Inc      None    Tank_ID      Integer // Instantly increases the level of a Tank_ID by a specified quantity.
              Tank_FillQty Real      // If the Tank_ID has insufficient capacity, it is filled as capacity becomes available.

              Real Tank_FillAmt

              While Tank_FillQty > 0 do
              {
                  Tank_FillAmt = Tank_FillQty
                  Wait Until Tank_Level[Tank_ID] < Cap (Loc(Tank_ID)) // wait until Tank_ID is not full
                  If Tank_FreeCap(Tank_ID) < Tank_FillAmt then Tank_FillAmt = Tank_FreeCap(Tank_ID)
                  Dec Tank_FillQty, Tank_FillAmt
                  Inc Tank_Level[Tank_ID], Tank_FillAmt
                  Tank_UpdateStats(Tank_ID)
              }

Tank_Dec      None    Tank_ID      Integer // Instantly decreases the level of a Tank_ID by a specified quantity.
              Tank_EmptyQty Real      // If the Tank_ID has insufficient quantity, it is emptied as material becomes available.

              Real Tank_EmptyAmt

              While Tank_EmptyQty > 0 do
              {
                  Tank_EmptyAmt = Tank_EmptyQty
                  Wait Until Tank_Level[Tank_ID] > 0.0 // wait until Tank_ID is not empty
                  If Tank_Level [Tank_ID] < Tank_EmptyAmt then Tank_EmptyAmt = Tank_Level [Tank_ID]
                  Dec Tank_EmptyQty, Tank_EmptyAmt
                  Dec Tank_Level[Tank_ID], Tank_EmptyAmt
                  Tank_UpdateStats(Tank_ID)
              }

Tank_FallTrigger None    Tank_ID      Integer // Waits until a Tank_ID falls to a specified level to trigger some action
                 Tank_FallLevel Real

                 Wait Until Tank_Level [tank_ID] > Tank_FallLevel // wait until level first rises above trigger level
                 Wait Until Tank_Level [tank_ID] <= Tank_FallLevel // wait until level drops to trigger level

Tank_RiseTrigger None    Tank_ID      Integer // Waits until a Tank_ID rises to a specified level to trigger some action
                 Tank_RiseLevel Real

                 Wait Until Tank_Level [tank_ID] < Tank_RiseLevel // wait until level first falls below trigger level
                 Wait Until Tank_Level [tank_ID] >= Tank_RiseLevel // wait until level rises to trigger level

Tank_Prep      None    Tank_ID      Integer // Prepares or cleans a Tank_ID before filling.
                 Tank_PrepTime Real      // Time is counted as Setup time.

                 Tank_Product[Tank_ID] = 0 // Clears out current product type
                 Tank_SetState (Tank_ID, Tank_Setup)
                 Wait Tank_PrepTime
                 Tank_SetState (Tank_ID, Tank_Idle)

Tank_SetLevel  None    Tank_ID      Integer // Instantly sets the Tank_ID level to the specified amount
                 Tank_SetQty Real      // If less than 0 or greater than the capacity, an error occurs.

                 If (Tank_SetQty < 0.0) or (Tank_SetQty > Tank_Cap(Tank_ID)) then
                     Stop "Set level " $ Tank_SetQty $ " is out of range for " $ Loc(Tank_ID)
                 Tank_Level[Tank_ID] = Tank_SetQty
                 Tank_UpdateStats(Tank_ID)

Tank_DoOperation None    Tank_ID      Integer // Delays some time for an operation such as mixing.
                 Tank_OperationTime Real // Time is counted as Operation time.

                 Tank_SetState (Tank_ID, Tank_Operation)
                 Wait Tank_OperationTime
                 Tank_SetState (Tank_ID, Tank_Blocked)

Tank_GoDown    None    Tank_ID      Integer // Sets Tank_ID state to Tank_Down for a specified time and then sets it to state prior to downtime.
                 Tank_DownTime Real      // Downtime statistics are updated.

                 int Tank_LastState = Tank_State [Tank_ID]
                 Inc Tank_Statistics[Tank_ID, 14] // Current number of DTs
                 if Tank_State [Tank_ID] <> Tank_Down
                 then
                 {
                     Inc Tank_Statistics[Tank_ID,3], Tank_Statistics[Tank_ID,1] * (Clock() - Tank_Statistics[Tank_ID,2]) // inc cum level
                     Tank_Statistics[Tank_ID,2] = Clock() // Update time since last content change
                     Tank_SetState (Tank_ID, Tank_Down)
                 }
                 Wait Tank_Downtime
                 Dec Tank_Statistics [Tank_ID,14]
                 If (Tank_Statistics[Tank_ID, 14] = 0) and (Tank_State[Tank_ID] = Tank_Down)
                 then
                 {

```

```

        Tank_Statistics[Tank_ID,2] = Clock() // Update time since last content change without changing contents
        Tank_SetState (Tank_ID, Tank_LastState)
    }

Tank_GoDownSched  None    Tank_ID      Integer // Sets Tank_ID state to Tank_Down for a specified time and then sets it to state prior to downtime.
                 Tank_DownTime Real // Downtime statistics are NOT updated.

    int Tank_LastState = Tank_State [Tank_ID]
    Inc Tank_Statistics[Tank_ID, 14] // Current number of DTs
    if Tank_State [Tank_ID] <> Tank_Down then
    {
        Tank_State[Tank_ID] = Tank_Down
        Tank_Statistics [Tank_ID, 6] = Clock() // update last state change time
    }
    Wait Tank_Downtime
    Dec Tank_Statistics [Tank_ID,14]
    If (Tank_Statistics[Tank_ID, 14] = 0) and (Tank_State[Tank_ID] = Tank_Down) then
    {
        // Tank_SetState (Tank_ID, Tank_LastState) --> Removed this line and replaced it with the following two
lines
        Tank_State[Tank_ID] = Tank_LastState
        Tank_Statistics[Tank_ID, 6] = Clock() // update last state change time
    }
Tank_SetState  None    Tank_ID      Integer // Updates Tank_Statistics and sets the Tank_State.
              SetState Integer
    Int Stat_Index = Tank_State[Tank_ID] + 7
    Inc Tank_Statistics [Tank_ID, Stat_Index], Clock() - Tank_Statistics [Tank_ID, 6] // inc cum state time
    Tank_Statistics [Tank_ID,6] = Clock() // update last state change time
    Tank_State[Tank_ID] = SetState

Tank_FreeCap   Real    Tank_ID      Integer // Returns the available capacity of a Tank_ID

Return Cap(Loc(Tank_ID)) - Tank_Level [Tank_ID]
Tank_Cap       Real    Tank_ID      Integer // Returns the defined capacity of a Tank_ID

Return Cap(Loc(Tank_ID))
Tank_UpdateStats  None    Tank_ID      Integer // Updates the statistics for a Tank_ID whenever the Level changes.
// It is called automatically in all of the pre-defined subroutines that fill, empty and transfer.

// Tank_Statistics array column numbers referenced:
// 1      Last level
// 2      Last change time
// 3      Cum time-weighted level
// 4      Entries
// 5      Max contents

Inc Tank_Statistics[Tank_ID,3], Tank_Statistics[Tank_ID,1] * (Clock() - Tank_Statistics[Tank_ID,2]) // inc cum level
Tank_Statistics[Tank_ID,2] = Clock() // Update time since last change
If Tank_Level[Tank_ID] > Tank_Statistics[Tank_ID,1] then // test for increase in level
{
    Inc Tank_Statistics[Tank_ID,4], Tank_Level[Tank_ID] - Tank_Statistics[Tank_ID,1] // increment unit entries
    If Tank_Level[Tank_ID] > Tank_Statistics[Tank_ID,5] Then // test for new max conts
        Tank_Statistics[Tank_ID,5] = Tank_Level[Tank_ID] // assign new max conts
    if Tank_Level[Tank_ID] = Cap(Loc(Tank_ID)) Then // test to see if now full
    {
        if Tank_State[Tank_ID] <> Tank_Down then
            Tank_SetState (Tank_ID, Tank_Blocked)
    }
    Else if Tank_State [Tank_ID] <> Tank_Filling then
        Tank_SetState (Tank_ID, Tank_Filling)
    }
    Else if Tank_Level[Tank_ID] < Tank_Statistics[Tank_ID,1] then // must be a drop in level
    {
        If Tank_Level[Tank_ID] = 0 Then // test to see if now empty
            Tank_SetState (Tank_ID, Tank_Idle)
        Else if Tank_State [Tank_ID] <> Tank_Emptying Then
            Tank_State [Tank_ID] = Tank_Emptying
    }
    Tank_Statistics[Tank_ID,1] = Tank_Level[Tank_ID] // Update last level to current level

Tank_Rate      Real    Tank_FromID Integer // Used to dynamically change a flow rate for an emptying FROM tank_ID and/or a filling TO tank_ID.
              Tank_ToID Integer // The user must modify this routine to return the desired rate.

// For example, if the flow rate from TankA to TankB decreases from 150 gpm to 100 gpm when TankB rises above
4000,
// the following code might be entered:

// If (Tank_FromID = TankA) and (Tank_ToID = TankB)
// then if Tank_Level[Tank_FromID] > 4000
// then return 100
// else return 150

// Replace the following code with your code.
if Tank_FromID = 0
then Stop "Passing 0 as the fill rate for " $ loc(Tank_ToID) $ " requires that you return a rate value using the subroutine
called Tank_Rate."
else Stop "Passing 0 as the empty or transfer rate for " $ loc(Tank_FromID) $ " requires that you return a rate value using
the subroutine called Tank_Rate."

Tank_SelectOutput Integer Tank_Start Integer // Selects an output tank_ID according to a selection rule. Works for up to 10 outputs.
                 Tank_Qty Integer
                 Tank_Rule Integer int Tank_Index
                 Tank_Limit Real Int Tank_WithLongestIdleTime
                 Tank_Prod Integer Real Tank_LongestIdleTime

```

```

Tank_Loop
{
  If Tank_Limit > 0 then // test for OK to fill a partial tank_ID
  {
    Tank_Index = Tank_Start
    while Tank_Index < Tank_Start + Tank_Qty do
    {
      //Test for partially full tank_ID with level equal to or less than the maximum level and same product
      // To select a partially full tank_ID only when it is filling and not emptying, add test for
      Tank_State[Tank_Index] <> Tank_Emptying. (see added change further below)
      if (Tank_Level [Tank_Index] > 0) and (Tank_Level [Tank_Index] <= Tank_Limit) and
      (Tank_Product[Tank_Index] = Tank_Prod)
      Then Return Tank_Index
      Else Inc Tank_Index
    }
  }

  // Test for idle tank_ID that meets selection rule
  Tank_Index = Tank_Start
  If Tank_Rule = Tank_InOrder
  Then while Tank_Index < Tank_Start + Tank_Qty do
  If Tank_State[Tank_Index] = Tank_Idle
  Then Return Tank_Index
  Else Inc Tank_Index
  Else if Tank_Rule = Tank_LongestIdle then
  {
    Tank_WithLongestIdleTime = 0
    Tank_LongestIdleTime = 999999999.0
    while Tank_Index < Tank_Start + Tank_Qty do
    {
      If Tank_State [Tank_Index] = Tank_Idle
      Then if Tank_Statistics[Tank_Index, 6] < Tank_LongestIdleTime then
      {
        Tank_LongestIdleTime = Tank_Statistics[Tank_Index, 6]
        Tank_WithLongestIdleTime = Tank_Index
      }
      Inc Tank_Index
    }
    If Tank_WithLongestIdleTime > 0 then Return Tank_WithLongestIdleTime
  }
  Else Stop "Invalid output selection rule defined for "$ Loc(Tank_Start)

  // Wait for a tank_ID to become available
  If Tank_Limit > 0 // Eliminate this test if a partial tank_ID can only fill when already filling and not emptying.
  then Wait 1 min // if waiting on a limit, check every minute since it is dependent on product type as well
  else // Wait for the first tank_ID to become idle
  If Tank_Qty < 2
  then Wait Until Tank_State[Tank_Start] = Tank_Idle
  else If Tank_Qty < 3
  then Wait Until (Tank_State[Tank_Start] = Tank_Idle) or (Tank_State[Tank_Start+1] = Tank_Idle)
  else If Tank_Qty < 4
  then Wait Until (Tank_State[Tank_Start] = Tank_Idle) or (Tank_State[Tank_Start+1] = Tank_Idle) or
  (Tank_State[Tank_Start+2] = Tank_Idle)
  else If Tank_Qty < 5
  then Wait Until (Tank_State[Tank_Start] = Tank_Idle) or (Tank_State[Tank_Start+1] = Tank_Idle) or
  (Tank_State[Tank_Start+2] = Tank_Idle)
  or (Tank_State[Tank_Start+3] = Tank_Idle)
  else If Tank_Qty < 6
  then Wait Until (Tank_State[Tank_Start] = Tank_Idle) or (Tank_State[Tank_Start+1] = Tank_Idle) or
  (Tank_State[Tank_Start+2] = Tank_Idle)
  or (Tank_State[Tank_Start+3] = Tank_Idle) or (Tank_State[Tank_Start+4] = Tank_Idle)
  else If Tank_Qty < 7
  then Wait Until (Tank_State[Tank_Start] = Tank_Idle) or (Tank_State[Tank_Start+1] = Tank_Idle) or
  (Tank_State[Tank_Start+2] = Tank_Idle)
  or (Tank_State[Tank_Start+3] = Tank_Idle) or (Tank_State[Tank_Start+4] = Tank_Idle) or
  (Tank_State[Tank_Start+5] = Tank_Idle)
  else If Tank_Qty < 8
  then Wait Until (Tank_State[Tank_Start] = Tank_Idle) or (Tank_State[Tank_Start+1] = Tank_Idle) or
  (Tank_State[Tank_Start+2] = Tank_Idle)
  or (Tank_State[Tank_Start+3] = Tank_Idle) or (Tank_State[Tank_Start+4] = Tank_Idle) or
  (Tank_State[Tank_Start+5] = Tank_Idle)
  or (Tank_State[Tank_Start+6] = Tank_Idle)
  else If Tank_Qty < 9
  then Wait Until (Tank_State[Tank_Start] = Tank_Idle) or (Tank_State[Tank_Start+1] = Tank_Idle) or
  (Tank_State[Tank_Start+2] = Tank_Idle)
  or (Tank_State[Tank_Start+3] = Tank_Idle) or (Tank_State[Tank_Start+4] = Tank_Idle) or
  (Tank_State[Tank_Start+5] = Tank_Idle)
  or (Tank_State[Tank_Start+6] = Tank_Idle) or (Tank_State[Tank_Start+7] = Tank_Idle)
  else If Tank_Qty < 10
  then Wait Until (Tank_State[Tank_Start] = Tank_Idle) or (Tank_State[Tank_Start+1] = Tank_Idle) or
  (Tank_State[Tank_Start+2] = Tank_Idle)
  or (Tank_State[Tank_Start+3] = Tank_Idle) or (Tank_State[Tank_Start+4] = Tank_Idle) or
  (Tank_State[Tank_Start+5] = Tank_Idle)
  or (Tank_State[Tank_Start+6] = Tank_Idle) or (Tank_State[Tank_Start+7] = Tank_Idle) or
  (Tank_State[Tank_Start+8] = Tank_Idle)
  or (Tank_State[Tank_Start+9] = Tank_Idle)
  }
  Tank_SelectInput Integer Tank_Start Integer // Selects an input tank_ID according to a rule. Works for up to 10 inputs.
}

```

```

Tank_Qty      Integer
Tank_Rule     Integer   int Tank_Index
Tank_Limit    Real      Int Tank_WithLongestBlockedTime
Tank_Prod     Integer   Real Tank_LongestBlockedTime

Tank_Loop
{
If Tank_Limit > 0 then // test for OK to draw from a partial tank_ID
{
Tank_Index = Tank_Start
while Tank_Index < Tank_Start + Tank_Qty do
{
// test for partially full tank_ID with level equal to or greater than the minimum level and same product
// To select only a tank_ID that is not being filled, add test for Tank_State[Tank_Index] <> Tank_Filling.
if (Tank_FreeCap(Tank_Index) > 0) and (Tank_Level [Tank_Index] >= Tank_Limit) and
(Tank_Product[Tank_Index] = Tank_Prod)
Then Return Tank_Index
Else Inc Tank_Index
}
}

// Test for blocked tank_ID that meets selection rule
Tank_Index = Tank_Start
If Tank_Rule = Tank_InOrder
Then while Tank_Index < Tank_Start + Tank_Qty do
If Tank_State[Tank_Index] = Tank_Blocked
Then Return Tank_Index
Else Inc Tank_Index
Else if Tank_Rule = Tank_LongestBlocked then
{
Tank_WithLongestBlockedTime = 0
Tank_LongestBlockedTime = 99999999.0
while Tank_Index < Tank_Start + Tank_Qty do
{
If Tank_State [Tank_Index] = Tank_Blocked
Then if Tank_Statistics[Tank_Index, 6] < Tank_LongestBlockedTime then
{
Tank_LongestBlockedTime = Tank_Statistics[Tank_Index, 6]
Tank_WithLongestBlockedTime = Tank_Index
}
}
Inc Tank_Index
}
If Tank_WithLongestBlockedTime > 0 then Return Tank_WithLongestBlockedTime
}
Else Stop "Invalid input selection rule defined for "$ Loc(Tank_Start)

// Wait for a tank_ID to have material for input
If Tank_Limit > 0 // If partial tanks can only selected if currently emptying (not filling), eliminate this test.
then Wait 1 min // if waiting on a limit, check every minute since it is dependent on product type as well
else // Wait for the first tank_ID to become blocked (full)
If Tank_Qty < 2
then Wait Until Tank_State[Tank_Start] = Tank_Blocked
else If Tank_Qty < 3
then Wait Until (Tank_State[Tank_Start] = Tank_Blocked) or (Tank_State[Tank_Start+1] = Tank_Blocked)
else If Tank_Qty < 4
then Wait Until (Tank_State[Tank_Start] = Tank_Blocked) or (Tank_State[Tank_Start+1] = Tank_Blocked) or
(Tank_State[Tank_Start+2] = Tank_Blocked)
else If Tank_Qty < 5
then Wait Until (Tank_State[Tank_Start] = Tank_Blocked) or (Tank_State[Tank_Start+1] = Tank_Blocked) or
(Tank_State[Tank_Start+2] = Tank_Blocked)
or (Tank_State[Tank_Start+3] = Tank_Blocked)
else If Tank_Qty < 6
then Wait Until (Tank_State[Tank_Start] = Tank_Blocked) or (Tank_State[Tank_Start+1] = Tank_Blocked) or
(Tank_State[Tank_Start+2] = Tank_Blocked)
or (Tank_State[Tank_Start+3] = Tank_Blocked) or (Tank_State[Tank_Start+4] = Tank_Blocked)
else If Tank_Qty < 7
then Wait Until (Tank_State[Tank_Start] = Tank_Blocked) or (Tank_State[Tank_Start+1] = Tank_Blocked) or
(Tank_State[Tank_Start+2] = Tank_Blocked)
or (Tank_State[Tank_Start+3] = Tank_Blocked) or (Tank_State[Tank_Start+4] = Tank_Blocked)
or (Tank_State[Tank_Start+5] = Tank_Blocked)
else If Tank_Qty < 8
then Wait Until (Tank_State[Tank_Start] = Tank_Blocked) or (Tank_State[Tank_Start+1] = Tank_Blocked) or
(Tank_State[Tank_Start+2] = Tank_Blocked)
or (Tank_State[Tank_Start+3] = Tank_Blocked) or (Tank_State[Tank_Start+4] = Tank_Blocked)
or (Tank_State[Tank_Start+5] = Tank_Blocked)
or (Tank_State[Tank_Start+6] = Tank_Blocked)
else If Tank_Qty < 9
then Wait Until (Tank_State[Tank_Start] = Tank_Blocked) or (Tank_State[Tank_Start+1] = Tank_Blocked) or
(Tank_State[Tank_Start+2] = Tank_Blocked)
or (Tank_State[Tank_Start+3] = Tank_Blocked) or (Tank_State[Tank_Start+4] = Tank_Blocked)
or (Tank_State[Tank_Start+5] = Tank_Blocked)
or (Tank_State[Tank_Start+6] = Tank_Blocked) or (Tank_State[Tank_Start+7] = Tank_Blocked)
else If Tank_Qty < 10
then Wait Until (Tank_State[Tank_Start] = Tank_Blocked) or (Tank_State[Tank_Start+1] = Tank_Blocked) or
(Tank_State[Tank_Start+2] = Tank_Blocked)
or (Tank_State[Tank_Start+3] = Tank_Blocked) or (Tank_State[Tank_Start+4] = Tank_Blocked)
or (Tank_State[Tank_Start+5] = Tank_Blocked)
or (Tank_State[Tank_Start+6] = Tank_Blocked) or (Tank_State[Tank_Start+7] = Tank_Blocked)
or (Tank_State[Tank_Start+8] = Tank_Blocked)
else If Tank_Qty < 11
then Wait Until (Tank_State[Tank_Start] = Tank_Blocked) or (Tank_State[Tank_Start+1] = Tank_Blocked) or
(Tank_State[Tank_Start+2] = Tank_Blocked)
or (Tank_State[Tank_Start+3] = Tank_Blocked) or (Tank_State[Tank_Start+4] = Tank_Blocked)
or (Tank_State[Tank_Start+5] = Tank_Blocked)

```



```

(Tank_State[Tank_Start+8] = Tank_Blocked)          or (Tank_State[Tank_Start+6] = Tank_Blocked) or (Tank_State[Tank_Start+7] = Tank_Blocked) or
                                                    or (Tank_State[Tank_Start+9] = Tank_Blocked)
}

```

```

*****
*           External Files           *
*****

```

ID	Type	File Name	Prompt
(null)	Shift	C:\Program Files\ProModel\Shifts\shift fix.sft	