



TESIS - K1142501

**SIMULASI KOREOGRAFI BERBASIS AGEN UNTUK MENGANALISIS
PERFORMA MULTI ORGANISASI DI TERMINAL PETIKEMAS**

ABD. CHARIS FAUZAN
05111650010063

DOSEN PEMBIMBING
Prof. Drs.Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.

**PROGRAM MAGISTER
BIDANG KEAHLIAN MANAJEMEN INFORMASI
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018**

[halaman ini sengaja dikosongkan]

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M.Kom.)

di

Institut Teknologi Sepuluh Nopember Surabaya

oleh:

Abd. Charis Fauzan

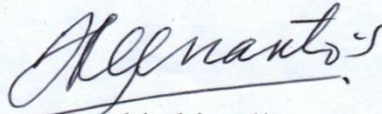
Nrp. 05111650010063

Dengan judul :
SIMULASI KOREOGRAFI BERBASIS AGEN
UNTUK MENGANALISIS PERFORMA MULTI ORGANISASI
DI TERMINAL PETIKEMAS

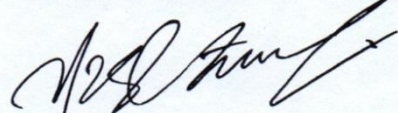
Tanggal Ujian : 8-1-2018
Periode Wisuda : 2017 Gasal

Disetujui oleh:

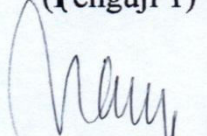
Prof.Ir.Drs.Ec. Riyanarto Sarno, M.Sc, Ph.D
NIP. 195908031986011001


(Pembimbing 1)

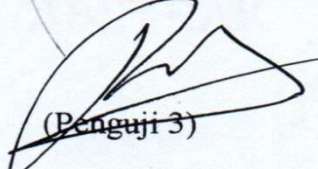
Waskitho Wibisono, S.Kom., M.Eng., Ph.D
NIP. 197410222000031001


(Penguji 1)

Dr. Ir. Raden Venantius Hari Ginardi, M.Sc
NIP. 196505181992031003

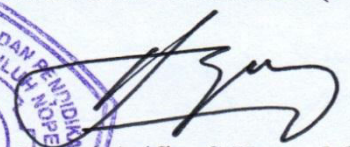

(Penguji 2)

Royyana Muslim I, S.Kom, M.Kom, Ph.D
NIP. 197708242006041001


(Penguji 3)

Dekan Fakultas Teknologi Informasi dan Komunikasi (FTIK),




Dr. Agus Zainal Arifin, S.Kom., M.Kom
NIP. 197208091995121001

[halaman ini sengaja dikosongkan]

SIMULASI KOREOGRAFI BERBASIS AGEN UNTUK MENGANALISIS PERFORMA MULTI ORGANISASI DI TERMINAL PETI KEMAS

Nama : Abd. Charis Fauzan
NRP : 05111650010063
Pembimbing 1 : Prof. Drs.Ec. Ir. Riyanarto Sarno M.Sc., Ph.D.

ABSTRAK

Koreografi pada proses bisnis mempunyai peran penting pada lingkungan komputing bisnis saat ini. Secara umum, koreografi melibatkan multi organisasi yang melakukan proses bisnis secara asinkron untuk mencapai tujuan bisnis secara bersama-sama. Koreografi pada multi organisasi dapat disimulasikan untuk mengetahui performa dari multi organisasi berdasarkan log data yang terekam pada sistem organisasi tersebut.

Penelitian ini mengusulkan simulasi koreografi pada multi organisasi yang bersifat asinkron. Simulasi memiliki metode tertentu untuk memetakan sistem nyata menjadi sistem simulasi, yaitu simulasi dinamis, simulasi discrete-event dan simulasi berbasis agen. Penentuan metode simulasi dipengaruhi oleh karakteristik kasus yang diselesaikan. Pada proses bisnis impor barang di PT. TPS, terjadi proses koreografi dengan melibatkan multi organisasi, sehingga terdapat beberapa organisasi yang saling berkomunikasi untuk mencapai tujuan proses. Komunikasi dalam koreografi tersebut membentuk karakteristik komunikasi organisasi yang bersifat asinkron ketika mengeksekusi proses bisnis sehingga mempengaruhi performa waktu dan biaya. Komunikasi asinkron ini dapat ditangani oleh agen dalam simulasi berbasis agen. Selain itu, penelitian ini juga memberikan kontribusi berkaitan dengan pembentukan log data di masa yang akan datang berdasarkan hasil *forecasting* jumlah container.

Penelitian ini menghasilkan evaluasi performa proses bisnis impor barang di PT. TPS untuk komunikasi yang bersifat asinkron dengan melibatkan beberapa agen. Evaluasi performa yang dihasilkan tidak hanya mengacu pada log data saat ini, melainkan juga mengacu pada log data di masa mendatang. Berdasarkan hasil simulasi didapatkan bahwa evaluasi *dwelling time* adalah 9,71 hari untuk simulasi berbasis agen dan 5,84 untuk simulasi discrete-event. Hal ini membuktikan bahwa komunikasi asinkron mempengaruhi waktu *dwelling time* di PT. TPS menjadi lebih tinggi.

Kata Kunci: Asinkron, Simulasi, Simulasi Berbasis Agen, Evaluasi Performa, Koreografi Proses Bisnis.

[halaman ini sengaja dikosongkan]

Agent Based Choreography Simulation for Analyzing the Performance of Multi Organization in Port Container Terminal

Student's Name : Abd. Charis Fauzan
NRP : 05111650010063
1st Advisor : Prof. Drs.Ec. Ir. Riyanarto Sarno M.Sc., Ph.D

ABSTRACT

Choreography on business processes has an important role in today's business computing environment. In general, choreography involves multiple organizations that perform asynchronous business processes to achieve business goals together. Choreography in multiple organizations can be simulated to determine the performance of multiple organizations based on log data recorded on the organization's system.

This study proposes choreographic simulations on multiple organizations that are asynchronous. The simulation has a specific method to map the real system into a simulation system, ie dynamic simulation, discrete-event simulation and agent-based simulation. The determination of the simulation method is influenced by the case characteristics completed. In the business process of import of goods in PT. TPS, there is a process of choreography involving multiple organizations, so there are several organizations that communicate with each other to achieve the purpose of the process. Communication in the choreography forms the characteristics of organizational communication that is asynchronous when executing business processes that affect the performance of time and cost. This asynchronous communication can be handled by agents in agent-based simulations. In addition, this study also contributes with the establishment of log data in the future based on the results of forecasting the number of containers.

This research resulted evaluation of business process performance of import goods in PT. TPS for communication is asynchronous with the involvement of several agents. The resulting performance evaluation not only refers to the current log of data, but also refers to the log data in the future. Based on the simulation results, the dwelling time evaluation is 9.71 days for agent-based simulation and 5.84 for discrete-event simulation. This proves that asynchronous communication affects the dwelling time in PT. TPS becomes higher.

Keywords: Business Process Simulation, Agent Based Simulation, Choreography, Performance Analysis,

[halaman ini sengaja dikosongkan]

KATA PENGANTAR

Alhamdulillahirabbil'alamin. Puji dan syukur penulis panjatkan kehadirat Allah SWT atas berkat, rahmat dan hidayah-Nya, penyusunan Tesis ini dapat diselesaikan. Tesis ini dibuat sebagai salah satu syarat dalam menyelesaikan Program Studi Magister di Institut Teknologi Sepuluh Nopember Surabaya. Penulis menyadari bahwa Tesis ini dapat diselesaikan karena dukungan dari berbagai pihak, baik dalam bentuk dukungan moral dan material. Melalui kesempatan ini dengan kerendahan hati penulis mengucapkan terima kasih dan penghargaan setinggi-tingginya kepada semua orang untuk semua bantuan yang telah diberikan, antara lain kepada:

1. Ibu Siti Nurul Mualimah, M.HI dan bapak Khoiroji tercinta yang tiada henti selalu mendukung anaknya, tetap sabar mendengar keluhannya, selalu mendoakan anaknya yang terbaik dan selalu menjadi panutan yang baik.
2. Bapak Prof. Drs. Ec. Ir. Riyanarto Sarno M.Sc., Ph.D selaku pembimbing yang senantiasa memberikan arahan dan bimbingan kepada penulis. Semoga Allah SWT senantiasa merahmati bapak dan keluarga.
3. Seluruh dosen S2 Teknik Informatika yang telah memberikan ilmu dan pengetahuan kepada penulis selama menempuh studi.
4. Teman seperjuangan dan sahabat serta teman lainnya yang tidak dapat disebutkan satu persatu, terutama TIM CAKDY terima kasih atas bantuan dan motivasi yang telah diberikan.

Akhirnya dengan segala kerendahan hati penulis menyadari masih banyak terdapat kekurangan pada Tesis ini. Oleh karena itu, segala tegur sapa dan kritik yang sifatnya membangun sangat penulis harapkan demi kesempurnaan Tesis ini. Penulis berharap bahwa perbuatan baik dari semua orang yang dengan tulus memberikan kontribusi terhadap penyusunan Tesis ini mendapatkan pahala dari Allah. Aamiin Alluhamma Aamiin.

Surabaya, Januari 2018

Penulis

[halaman ini sengaja dikosongkan]

DAFTAR ISI

HALAMAN PENGESAHAN	Error! Bookmark not defined.	iii
ABSTRAK		v
ABSTRACT		vii
KATA PENGANTAR		ix
DAFTAR ISI		xi
DAFTAR GAMBAR		xiii
DAFTAR TABEL		xv
BAB I PENDAHULUAN		1
1.1 Latar Belakang		1
1.2 Perumusan Masalah		3
1.3 Batasan Masalah		4
1.4 Tujuan Penelitian		4
1.5 Manfaat Penelitian		4
1.6 Kontribusi Penelitian		4
BAB II KAJIAN PUSTAKA		5
2.1 <i>Web Service Composition</i>		5
2.2 Koreografi Proses Bisnis		7
2.3 Simulasi Proses Bisnis		8
2.4 Simulasi Berbasis Agen		10
2.5 Log data		13
BAB III METODE PENELITIAN		15
3.1 Studi Literatur		15
3.2 Rancangan Penelitian		16
3.2.1 Pre-Processing		17
3.2.2 <i>Forecasting</i> Jumlah Kontainer		22
3.2.3 Membentuk <i>Forecasted</i> Log data		24
3.2.4 Simulasi Berbasis Agen		25
3.2.5 Evaluasi Performa		27
BAB IV HASIL PENELITIAN DAN PEMBAHASAN		31
4.1 Hasil Penelitian		31

4.1.1	Lingkungan Uji Coba	31
4.1.2	Pre-Processing.....	31
4.1.3	<i>Forecasting</i> Jumlah Container	38
4.1.4	<i>Generate</i> Log Data.....	44
4.1.5	Simulasi Berbasis Agen.....	52
4.1.6	Simulasi Discrete-Event	56
4.1.7	Paralelisasi	58
BAB V PENUTUP		61
5.1	Kesimpulan.....	61
5.2	Saran	62
DAFTAR PUSTAKA.....		63
Lampiran-Lampiran.....		65
BIOGRAFI PENULIS.....		77

DAFTAR GAMBAR

Gambar 2.1 Koreografi antara retail dan supplier	7
Gambar 2.2 Alur Simulasi dalam Penggalian Proses	8
Gambar 2.3 Prosedur Simulasi Proses Bisnis oleh (Rozinat et al., 2008)	9
Gambar 2.4 Contoh Log data	13
Gambar 3.1 Alur Metodologi Penelitian	15
Gambar 3.2 Rancangan Penelitian.....	16
Gambar 3.3 Contoh Log data Proses Impor pada PT. PTS.....	17
Gambar 3.4 Komunikasi antar agen	18
Gambar 3.5 Gambaran simulasi untuk aktivitas level 0	26
Gambar 3.6 Gambaran simulasi atribut komunikasi antar organisasi	26
Gambar 3.7 Model Koreografi Proses Impor di PT. TPS.....	27
Gambar 3.8 Algoritma evaluasi performa pada descrite-event simulation.....	28
Gambar 3.9 Metode modifikasi untuk menangani <i>single time stamp</i>	29
Gambar 4.1 <i>Forecasting</i> jumlah kontainer menggunakan <i>Moving Average</i>	41
Gambar 4.2 <i>Forecasting</i> jumlah kontainer menggunakan <i>simple exponential smoothing</i>	42
Gambar 4.3 <i>Forecasting</i> jumlah kontainer menggunakan <i>double exponential smoothing</i>	43
Gambar 4.4 Prosentase, <i>average</i> dan standar deviasi setiap <i>trace</i>	47
Gambar 4.5 Distribusi Normal Gaussian	48
Gambar 4.6 <i>Generated trace</i> menggunakan random distribusi normal.....	49
Gambar 4.7 <i>Generated trace</i> menggunakan prosentase	50
Gambar 4.8 total <i>Generated trace</i>	50
Gambar 4.9 <i>generated log data</i> berdasarkan <i>generated trace</i>	52
Gambar 4.10 Implementasi Activity pada <i>Agent Based Simulation</i>	53
Gambar 4.11 Implementasi <i>message</i> pada Agent Based Simulation	54
Gambar 4.7 Laporan <i>dwelling time</i> bulan Januari-April melalui simulasi berbasis agen.....	55
Gambar 4.8 Gantt Chart untuk menunjukkan komunikasi asinkron	55
Gambar 4.9 Implementasi Discrete Event Simulation Anylogic.....	57
Gambar 4.10 Laporan <i>dwelling time discrete-event simulation</i>	57
Gambar 4.11 Gantt Chart pada simulasi <i>discrete-event</i>	58
Gambar 4.12 Paralelisasi Bea Cukai dan Karantina	59
Gambar 4.13 <i>Dwelling time parallel ABS</i>	59

[halaman sengaja dikosongkan]

DAFTAR TABEL

Tabel 2.1 Penelitian terkait	10
Tabel 4.1. Dasar perhitungan sojourn time pada tiap aktivitas	33
Tabel 4.2 cost, sojourn time (ST), execution time (ET), dan waiting time (WT)..	37
Table 4.3 Jumlah kontainer harian	40
Table 4.4 Perbandingan MSE dan RE metode forecasting	43
Table 4.5 Jumlah kontainer hasil forecasting Bulan April 2016	45
Table 4.6 Frekuensi masing-masing trace Bulan Januari – Maret 2016	46
Table 4.8 Perbandingan Dwelling Time Parallel dan Non-Parallel ABS	59

[halaman sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1 Latar Belakang

Koreografi pada proses bisnis mempunyai peran penting pada lingkungan komputing bisnis saat ini untuk mengembangkan aplikasi yang terdistribusi di berbagai *web service*. Secara umum, koreografi pada proses bisnis merupakan suatu kombinasi antara dua atau lebih organisasi yang diterapkan dalam *web service*. Berbagai penelitian yang berkembang saat ini memosisikan koreografi pada proses bisnis untuk membentuk *web service* yang B2B (*business to business*) atau melibatkan berbagai organisasi dalam satu *service* (Solaiman & Molinajimenez, 2015). Implementasi nyata koreografi pada proses bisnis adalah membentuk *web service* guna mengkombinasi dua atau lebih organisasi proses bisnis (Ahmed & Srivastava, 2014). Kelemahan implementasi koreografi dalam *web service* adalah terbatas pada organisasi yang diimplementasikan saja, dan harus mengulang proses koreografi jika diimplementasikan untuk organisasi lain. Selain itu, implementasi koreografi pada *web service* juga membutuhkan usaha dan biaya karena melibatkan berbagai sumber daya (Milanovi, 2010). Penelitian ini mengusulkan sistem simulasi untuk koreografi pada lintas organisasi yang berguna untuk melakukan evaluasi performa pada multi organisasi melalui simulasi pada log data saat ini. Selain itu, dimungkinkan untuk mengevaluasi performa proses bisnis dimasa yang akan datang dengan membentuk suatu log data berdasarkan hasil *forecasting*.

Simulasi secara umum dapat diartikan sebagai tiruan dari aktivitas sebuah sistem, yang dibuat dengan tujuan untuk mengamati perubahan karakteristik pada sistem nyata (Buliali, 2013). Melalui simulasi, parameter-parameter pada koreografi proses bisnis dapat diubah untuk menentukan kinerja koreografi yang optimal sebagai masukan pada perusahaan untuk memperbaiki kinerja dimasa yang akan datang. Dalam penelitian ini, diusulkan suatu simulasi koreografi pada lintas organisasi yang terekam dalam bentuk log data. Sebagai studi kasus, log data yang diolah dalam penelitian ini adalah log data pada proses

bisnis impor barang di PT. Terminal Peti Kemas Surabaya (PT.TPS). Proses bisnis impor barang pada PT TPS dijalankan dengan melibatkan multi organisasi, yaitu *customer*, PT. TPS, karantina dan bea cukai. Masalah yang berkembang pada proses impor barang di terminal peti kemas adalah lamanya waktu bongkar muat kapal (*dwelling time*), sehingga melebihi waktu standar yang ditetapkan pemerintah selama 2-3 hari. Dwelling time yang lama membuat proses logistik terganggu. Dalam menjalankan proses bisnis, multi organisasi yang terlibat memiliki pola komunikasi yang asinkron. Komunikasi asinkron adalah komunikasi yang tidak terjadwal. Organisasi yang terlibat dapat berkomunikasi kapan saja walaupun terdapat organisasi yang sedang menjalankan aktivitas tertentu, sehingga dimungkinkan bahwa pola komunikasi asinkron akan mengganggu aktivitas organisasi lain (Park, Sutrisnowati, & Bae, 2016). Maka, tantangan yang perlu diselesaikan adalah melakukan simulasi untuk melakukan evaluasi pada proses logistik di terminal peti kemas. Selain permasalahan evaluasi performa, simulasi juga diharapkan dapat mendeteksi dan mengurangi waiting time pada proses bisnis (Rozinat, Mans, Song, & van der Aalst, 2008). Karena waiting time menjadi pemicu *dwelling time* pada terminal peti kemas. Simulasi mampu melakukan evaluasi pada log data saat ini (*existing event log*) dan juga mampu melakukan simulasi berdasarkan log data hasil peramalan (*forecasting*) pada suatu proses bisnis. *Forecasting* bermanfaat untuk memberikan masukan kepada para pemangku kebijakan untuk menentukan langkah strategis terhadap masa depan proses bisnis (Khodyrev & Popova, 2014).

Simulasi memiliki metode tertentu untuk memetakan sistem nyata menjadi sistem simulasi. (Grigoryev, 2016) membagi metode simulasi menjadi 3 metode utama, yaitu simulasi dinamis, simulasi *discrete-event* dan simulasi berbasis agen (*agent based simulation*). Penentuan metode simulasi dipengaruhi oleh karakteristik kasus yang diselesaikan. Umumnya, proses bisnis adalah kasus yang dapat diselesaikan dengan simulasi *discrete-event* karena perubahan variabel *state* pada kejadian berlangsung secara diskrit. Penelitian simulasi proses bisnis dengan metode *discrete-event* telah dilakukan dengan mengacu pada log data untuk menemukan model proses (A. Rozinat, R.S. Mans, M. Song, 2010). Pada proses bisnis impor barang di PT. TPS, terjadi proses koreografi dengan

melibatkan multi organisasi, sehingga ada beberapa organisasi yang saling berkomunikasi secara asinkron untuk mencapai tujuan proses. Pola komunikasi ini dapat ditangani oleh agen dalam simulasi berbasis agen. Hal ini dikuatkan dengan penelitian sebelumnya untuk menemukan perilaku agen berdasarkan penggalian log data (Ferreira, Szimanski, & Ralha, 2013), dan melakukan simulasi berbasis agen berdasarkan acuan perilaku komunikasi agen yang didapatkan dari log data untuk meningkatkan proses bisnis (Szimanski, Ralha, Wagner, & Ferreira, 2013). Oleh karenanya, kasus proses bisnis impor barang di PT. TPS memiliki karakteristik untuk diterapkan simulasi berbasis agen mengacu pada pola komunikasi koreografi yang bersifat asinkron.

Berdasarkan latar belakang yang telah diuraikan, peneliti mengusulkan untuk mengembangkan simulasi koreografi berbasis agen yang bersifat asinkron pada proses bisnis PT. TPS. Adapun fokus utama penelitian ini adalah melakukan simulasi untuk mengevaluasi kinerja setiap aktivitas pada multi organisasi. Simulasi tersebut tidak hanya diperuntukkan untuk log data saat ini, melainkan juga untuk simulasi log data dimasa yang akan datang berdasarkan hasil *forecasting* jumlah kontainer. Hasil simulasi berbasis agen akan dibandingkan dengan simulasi discrete event untuk membuktikan bahwa pola komunikasi asinkron mempengaruhi dwelling time di PT. TPS.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan perumusan masalah dalam penelitian ini adalah sebagai berikut.

1. Bagaimana melakukan *forecasting* jumlah kontainer?
2. Bagaimana membentuk *forecasted* log data mengacu pada hasil *forecasting* jumlah kontainer?
3. Bagaimana mengevaluasi performa multi organisasi pada terminal petikemas menggunakan simulasi berbasis agen?

1.3 Batasan Masalah

Untuk menghindari meluasnya permasalahan yang akan diselesaikan, maka batasan masalah dalam penelitian ini adalah:

1. Simulasi berbasis agen dan simulasi *discrete-event* dilakukan menggunakan kakas bantu *AnyLogic 8.0 Personal Learning Edition*.
2. Studi kasus yang diujikan adalah proses bisnis impor barang pada PT. TPS.
3. Multi organisasi yang terlibat antara lain: *customer*, karantina, bea cukai dan PT. TPS

1.4 Tujuan Penelitian

Tujuan pada penelitian ini adalah untuk mengevaluasi performa proses bisnis impor barang pada Terminal Petikemas Surabaya melalui simulasi koreografi berbasis agen. Evaluasi performa tidak hanya untuk log data saat ini, melainkan juga untuk log data di waktu yang akan datang (*forecasted log data*). Evaluasi performa ini berguna untuk memberikan masukan pada PT. TPS terkait *dwellling time* yang diperoleh saat simulasi.

1.5 Manfaat Penelitian

Pendekatan simulasi koreografi berbasis agen pada proses bisnis impor barang ini diharapkan dapat membantu para pemangku kebijakan (*stakeholder*) PT. TPS untuk menentukan kebijakan strategis dan perbaikan perusahaan di masa mendatang.

1.6 Kontribusi Penelitian

Kontribusi dari penelitian ini adalah mengusulkan :

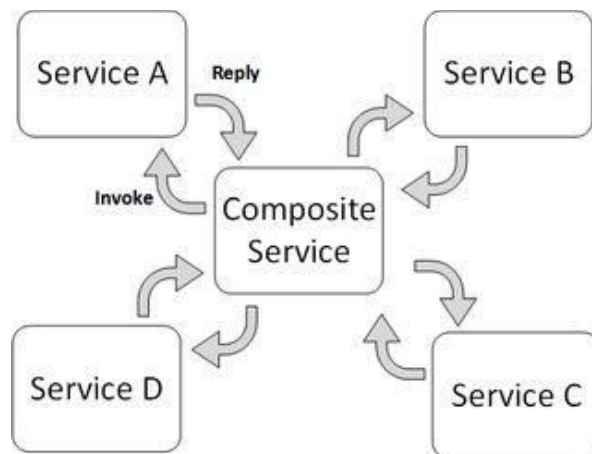
1. Mengusulkan metode untuk membentuk log data berdasarkan hasil *forecasting* jumlah kontainer dimasa yang akan datang, hasil pembentukan log data disebut sebagai *forecasted log data*.
2. Komparasi simulasi berbasis agen dan simulasi *discrete-event* untuk mengevaluasi performa multi organisasi menggunakan log data saat ini (*existing log data*) dan log data masa depan (*forecasted log data*).

BAB II KAJIAN PUSTAKA

Pada bab ini akan diuraikan mengenai konsep dasar tentang teori yang akan digunakan dalam penelitian ini. Pemaparan tersebut meliputi penjelasan tentang *web service composition*, koreografi proses bisnis, simulasi proses bisnis, simulasi berbasis agen serta log data.

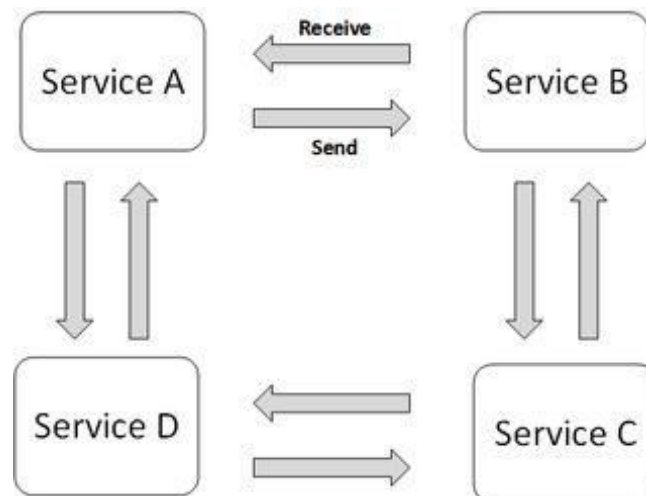
2.1 *Web Service Composition*

Komposisi mengacu pada bagaimana cara sesuatu dibangun, *web service composition* pada dasarnya membentuk suatu layanan (*service*) dengan memanfaatkan beragam layanan berbeda dalam aplikasi gabungan. Sehingga fungsi aplikasi yang berbeda bisa digunakan dalam satu aplikasi. Komposisi layanan mencakup semua proses yang menciptakan layanan bernilai tambah yang disebut sebagai layanan komposit atau agregat (Bai et al. 2009). *Web service composition* diimplementasikan menggunakan teknologi XML, SOAP, WSDL guna menyediakan sarana dalam kaitannya untuk mendeskripsikan, menemukan, dan memanggil layanan sebagai entitas dengan sendirinya. Namun, teknologi tersebut tidak memberikan detail perilaku tentang peran layanan dalam kolaborasi yang lebih kompleks. Kolaborasi ini mencakup serangkaian kegiatan dan hubungan antar aktivitas, yang membangun proses bisnis. Ada dua cara untuk membangun *web service composition*: yaitu orkestrasi *web service* dan koreografi *web service* (Peltz, 2003). Dalam orkestrasi, sebagaimana yang ditunjukkan oleh Gambar 2.1, terdapat sebuah proses terpusat yang mengatur semua *web service* yang terlibat didalamnya. *web service* tidak perlu tahu bahwa ia sebenarnya terlibat dalam bagian dari proses bisnis yang lebih tinggi tingkatannya karena hanya proses terpusat yang mengetahui ini. Hubungan antara semua layanan yang berpartisipasi dijelaskan oleh satu titik akhir, yaitu layanan gabungan. Orkestrasi mencakup pengelolaan transaksi antara layanan individual. Orkestrasi menggunakan pendekatan terpusat untuk komposisi layanan (Aalst, 2003).



Gambar 2.1 Orkestrasi pada *web service*

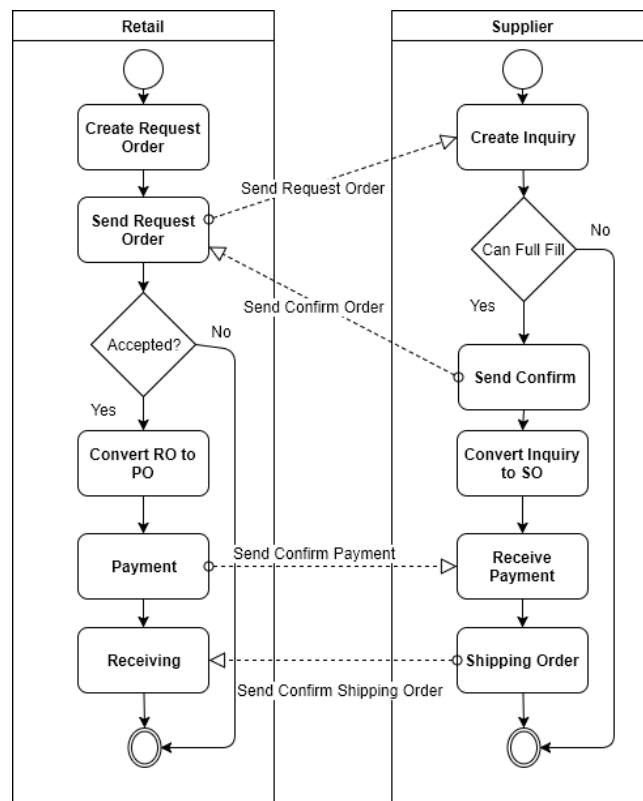
Namun berbeda halnya dengan orkestrasi, koreografi tidak bergantung pada suatu koordinator sentral. Sebagaimana ditunjukkan pada Gambar 2.2, Setiap *web service* yang terlibat akan mengetahui kapan harus mengeksekusi *service* masing-masing dan dengan siapa harus berinteraksi. Koreografi merupakan usaha yang bersifat kolaboratif yang berfokus pada pertukaran *message*. Hal ini berarti setiap *web service* yang terlibat perlu mengetahui proses bisnis, operasi yang dijalankan, organisasi yang terlibat, *message* yang dipertukarkan serta waktu/durasi pertukaran *message*. Logika sederhana untuk koreografi adalah seperti halnya tubuh manusia, di mana anggota yang berbeda saling bergantung dan bekerja untuk tujuan bersama, sehingga dalam koreografi tidak ada sistem terpusat seperti dalam orkestrasi.



Gambar 2.2 Koreografi pada *web service*

2.2 Koreografi Proses Bisnis

Proses bisnis multi organisasi adalah aktivitas bisnis yang didalamnya melibatkan lebih dari satu organisasi untuk mencapai tujuan bisnis (*business goal*). Maka dalam menjalankan proses bisnis multi organisasi diperlukan pola komunikasi antar organisasi untuk berinteraksi. Interaksi proses bisnis antar organisasi tersebut juga dikenal sebagai koreografi proses bisnis. Sebuah koreografi mendefinisikan urutan dan kondisi di mana banyak agen independen yang bekerja sama menukar pesan agar bisa melakukan suatu tugas untuk mencapai suatu tujuan Negara (El Kholly, Bentahar, El Menshawy, Qu, & Dssouli, 2014). Sebuah koreografi mendefinisikan aturan umum yang dapat digunakan kembali mengatur pemesanan pesan yang dipertukarkan, dan ketentuannya pola perilaku kolaboratif, sebagaimana disepakati antara dua atau lebih peserta yang berinteraksi. Aspek utama koreografi pada proses bisnis adalah pesan yang dipertukarkan di antara pihak yang berkolaborasi, yang menyetujui peraturan untuk memesan pesan (Jung, Kang, & Kim, 2004).

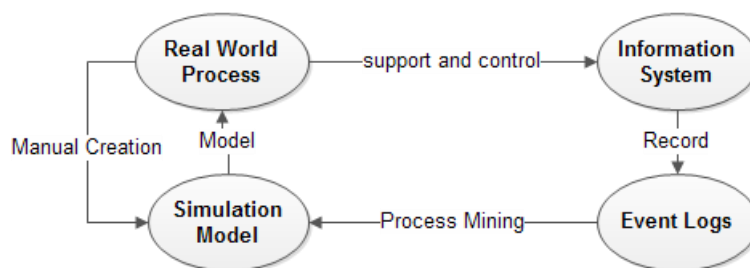


Gambar 2.3 Koreografi antara *retail* dan *supplier*

Koreografi pada proses bisnis mempunyai peran penting pada lingkungan komputing bisnis saat ini untuk mengembangkan aplikasi yang terdistribusi di berbagai *web service*. Gambar 2.3 menunjukkan gambaran koreografi pada dua organisasi. Secara umum, koreografi pada proses bisnis merupakan suatu gabungan antara dua atau lebih organisasi yang diterapkan dalam *web service*. Berbagai penelitian yang berkembang saat ini memposisikan koreografi pada proses bisnis untuk membentuk *web service* yang B2B (*business to business*) atau melibatkan berbagai organisasi dalam satu *service* (Solaiman & Molina-jimenez, 2015). Implementasi nyata koreografi pada proses bisnis adalah membentuk *web service* guna membentuk dua atau lebih organisasi proses bisnis.

2.3 Simulasi Proses Bisnis

Simulasi secara umum dapat diartikan sebagai tiruan dari aktivitas sebuah sistem, yang dibuat dengan tujuan untuk mengamati perubahan karakteristik pada sistem nyata (Buliali, 2013). Melalui simulasi, parameter-parameter pada koreografi proses bisnis dapat diubah untuk menentukan kinerja koreografi yang optimal sebelum diimplementasikan pada sistem nyata, salah satunya melalui *web service*.

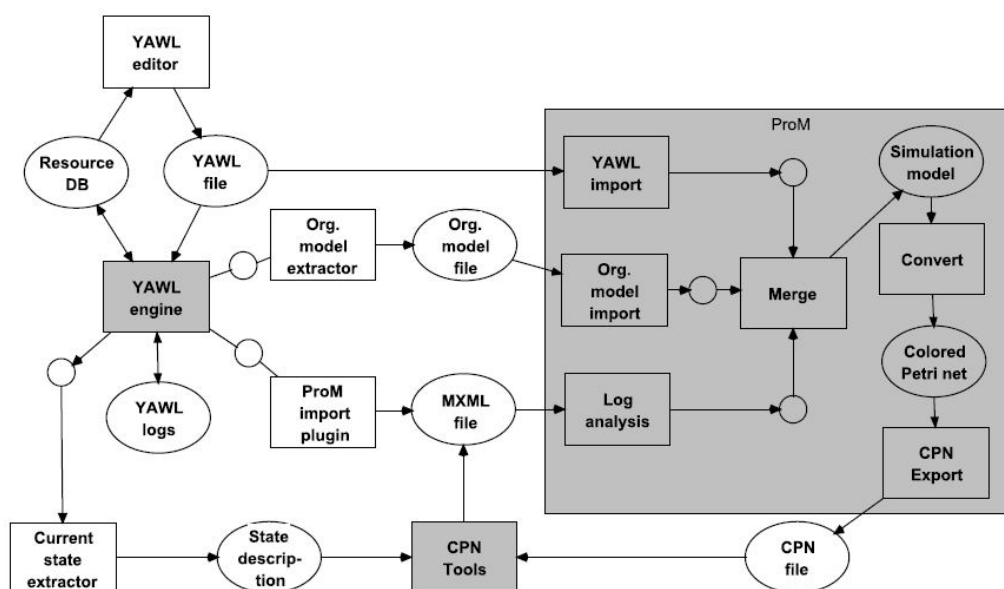


Gambar 2.4 Alur Simulasi dalam Penggalian Proses

Gambar 2.4 menjelaskan alur dasar mengapa simulasi proses bisnis diperlukan. Secara umum, proses bisnis dibentuk berdasarkan aktivitas yang terjadi di kehidupan nyata (*real-world process*), proses bisnis tersebut dikumpulkan dalam suatu sistem yang menggunakan teknologi untuk mendukung operasi dan manajemen, sistem tersebut dikenal populer sebagai sistem informasi

(*information system*). Melalui sistem informasi, setiap aktivitas yang berjalan dapat terekam secara otomatis jika sistem informasi tersebut memiliki mekanisme perekaman log data. Melalui log data tersebut, teknik proses mining dapat berkembang, yaitu metode untuk menggali proses dari log data untuk mendapatkan model proses. Model proses yang terbentuk dari log data dikembangkan untuk melakukan suatu simulasi.

Pada penelitian terdahulu, berkembang teknik simulasi proses bisnis menggunakan CPN Tools. Simulasi ini berguna untuk *estimating the effects of some redesign*, yaitu mampu memperkirakan bahkan meramalkan (*forecasting*) efek dari perubahan pada proses bisnis ketika dibuat skenario yang berbeda untuk simulasi (Rozinat et al., 2008). Simulasi dapat digambarkan secara manual menggunakan CPN Tools atau secara otomatis melalui penggalian model proses log data. Simulasi ini mampu mengestimasi atau memperkirakan kemungkinan yang terjadi seandainya dilakukan perubahan dalam proses bisnis. CPN Tools adalah perkakas lanjutan setelah ProM, karena ProM hanya mampu untuk menganalisis model proses berdasarkan log data yang ada. CPN Tools dan ProM adalah perkakas bantu yang saling terkait satu sama lain ditinjau dari proses simulasi pada Gambar 2.5.



Gambar 2.5 Prosedur Simulasi Proses Bisnis oleh (Rozinat et al., 2008)

2.4 Simulasi Berbasis Agen

Simulasi berbasis agen (ABS) berfokus pada analisis sistem bisnis melibatkan interaksi antar agen. ABS dapat digunakan untuk mewakili pengaturan organisasi secara alami, karena melibatkan pelaku usaha yang berkomunikasi dan berinteraksi satu sama lain. Secara khusus, dimungkinkan untuk menggunakan ABS untuk mensimulasikan pelaksanaan proses bisnis, sehingga menghasilkan suatu peristiwa log. Dalam pendekatan penelitian sebelumnya (Szimanski et al., 2013) digunakan suatu *framework* simulasi berbasis agen yang disebut dengan AOR. *Framework* ini yang menyediakan konstruksi tingkat tinggi (*high level*) seperti model proses yang berisikan aktivitas dan konstruksi tingkat rendah (*low level*) seperti acara pesan masuk dan keluar untuk memudahkan pemetaan proses bisnis (Ferreira et al., 2013).

Tabel 2.1 Penelitian terkait

Peneliti	Data	Metode	Peluang Kontribusi
(A. Rozinat, R.S. Mans, M. Song, 2010)	Log data proses bisnis penerimaan pasien di rumah sakit	Log data digali model prosesnya menggunakan algoritma proses mining, kemudian ditentukan organization perspektif dan data decision-nya. Hasilnya kemudian di export menggunakan plugin CPN Tools Export via ProM. Selanjutnya model proses disimulasikan menggunakan CPN Tools untuk dilakukan analisis terhadap waktu eksekusi, waktu tunggu	Simulasi hanya mengacu pada model proses terutama aktivitas proses bisnis. Oleh karenanya, simulasi ini berbasiskan <i>discrete-event</i> . Penelitian ini dapat dikembangkan dengan menambahkan perspektif organisasi, bukan hanya pada originator yang menjalankan aktivitas. Melainkan melibatkan komunikasi yang dilakukan antar

		dan waktu <i>sojourn</i> . Simulasi tersebut menjadi acuan untuk memperbaiki kinerja sistem penerimaan pasien di rumah sakit	organisasi.
(Onggo, 2012)	Proses bisnis <i>sugarcane</i> yang dimodelkan dalam bentuk notasi BPMN	Mengumpulkan organisasi yang terlibat dalam proses bisnis. Kemudian membentuk notasi pada BPMN untuk menggambarkan agen yang bisa berkomunikasi via pesan antar organisasi.	Penelitian ini hanya memberikan konsep menggambarkan agen dalam notasi BPMN. Belum melibatkan simulasi berbasis agen sehingga dapat menganalisis kinerja dari model agen yang telah dikonsepsi sebelumnya.
(Szimanski et al., 2013)	Proses bisnis purchasing dalam bentuk model high level (BPMN) dan low level (log data)	Penelitian ini mengusulkan metode untuk melakukan pemetaan dari dan low level (log data) menuju high level (model proses BPMN). Pemetaan perlu dilakukan karena data log data bukan disediakan dalam bentuk aktivitas, melainkan dalam bentuk <i>message</i> . Pemetaan dilakukan dengan membuat hirarki	Penelitian ini memberikan terobosan baru dalam melakukan pemetaan untuk log data yang didalamnya tidak mengandung atribut aktivitas sebagaimana log data pada umumnya. Penelitian ini perlu dikembangkan dengan melakukan simulasi berbasis agen guna mengetahui kinerja dari log data.

		<p>proses bisnis untuk membentuk keterkaitan antara log data dan model proses bisnis. Selanjutnya dilakukan perhitungan hidden markov untuk menganalisis keterkaitan.</p>	
(Park et al., 2016)	Log data di terminal petikemas	<p>Log data digali model prosesnya menggunakan algoritma proses mining, kemudian ditentukan <i>organization perspektif</i> dan data <i>decision</i>-nya. Hasilnya kemudian di ekspor menggunakan plugin CPN Tools Export via ProM. Selanjutnya model proses disimulasikan menggunakan CPN Tools untuk dilakukan analisis terhadap waktu eksekusi, waktu tunggu dan waktu <i>sojourn</i>. Simulasi tersebut dapat melakukan peramalan terhadap kemungkinan yang ditimbulkan di masa mendatang,</p>	<p>Penelitian ini dapat dikembangkan dengan menambahkan perspektif lintas organisasi dalam log data sebagai data simulasi, karena penelitian ini tidak mempertimbangkan organisasi dan mengacu hanya pada aktivitas terminal petikemas.</p>

		terutama untuk melakukan simulasi konfigurasi peralatan pelabuhan, seandainya terdapat kontainer yang jumlahnya bertambah	
--	--	---	--

2.5 Log data

Pada penambangan proses, untuk menganalisis suatu proses bisnis digunakan log data dari proses bisnis tersebut sebagai acuan. Log data merupakan suatu catatan histori yang memuat rangkaian aktivitas pada suatu sistem. Log data menjadi suatu bukti telah terjadinya suatu transaksi pada suatu proses yang sedang atau telah berlangsung. Log data berisi informasi mengenai rekam data di suatu kasus pada aktivitas tertentu. Kasus yang dimaksud yaitu suatu kejadian yang sedang berlangsung. Log data terdiri dari beberapa atribut diantaranya adalah *case id*, *activity id*, *timestamp* berisikan waktu mulai dan waktu selesai namun terkadang hanya terdapat satu waktu saja, dan originator. Gambar 2.6 menunjukkan contoh dari log data dari suatu kasus (Sarno, Haryadita, Kartini, Sarwosri, & Solichaha, 2016).

Case ID	Activity	Start Stamp	End Stamp
PP1	A	8/20/2016 10:32	8/20/2016 13:42
PP1	B	8/20/2016 13:42	8/20/2016 16:52
PP1	D	8/20/2016 18:27	8/20/2016 21:37
PP1	E	8/20/2016 21:37	8/21/2016 0:47
PP1	G	8/21/2016 0:47	8/21/2016 3:57
PP1	H	8/21/2016 3:57	8/21/2016 7:07
PP1	I	8/21/2016 7:07	8/21/2016 10:17
PP1	K	8/21/2016 10:17	8/21/2016 13:27
PP2	A	8/21/2016 13:27	8/21/2016 16:37
PP2	B	8/21/2016 16:37	8/21/2016 19:47
PP2	D	8/21/2016 21:22	8/22/2016 0:32
PP2	E	8/22/2016 0:32	8/22/2016 3:42
PP2	G	8/22/2016 3:42	8/22/2016 6:52
PP2	H	8/22/2016 6:52	8/22/2016 10:02
PP2	I	8/22/2016 10:02	8/22/2016 13:12
PP2	K	8/22/2016 13:12	8/22/2016 16:22

Gambar 2.6 Contoh Log data

Menurut (Aalst, 2016), log data didefinisikan sebagai suatu set proses eksekusi yang mengambil data aktivitas proses bisnis yang dilakukan dalam konteks tertentu. Atau dengan kata lain, log data merupakan catatan dari eksekusi aktivitas dalam suatu proses bisnis. Catatan eksekusi ini dapat menyimpan data berupa waktu dilaksanakannya suatu aktivitas, pelaku yang melaksanakan aktivitas, dan lain-lain sesuai dengan kebutuhan dari perusahaan yang menjalankan log datanya. Dalam log data dapat terdiri dari berbagai macam kasus, jejak, dan aktivitas .

- Kasus dan Jejak

Kasus merupakan suatu kasus tertentu yang ada pada log data. Kasus tersebut dapat berupa suatu kasus dalam memproduksi suatu barang tertentu, karena log data dapat terdiri dari catatan dari proses eksekusi pembuatan banyak barang atau proses eksekusi dari banyak kasus proses. Sedangkan jejak merupakan alur dari aktivitas yang dijalankan dalam suatu proses. Misal dalam suatu log data (E):

$$E = [< a, c, d >^{45}, < b, c, d >^{42}, < a, c, e >^{38}, < b, c, e >^{22}]$$

Dalam log data tersebut:

- Terdapat 4 jejak yaitu (a,c,d), (b,c,d), (a,c,e), (b,c,e).
- Terdapat 147 kasus karena (a, c, d) dilakukan sebanyak 45 kali, (b, c, d) sebanyak 42 kali, (a, c, e) sebanyak 38 kali, dan (b, c, e) sebanyak 22 kali.

- Aktivitas

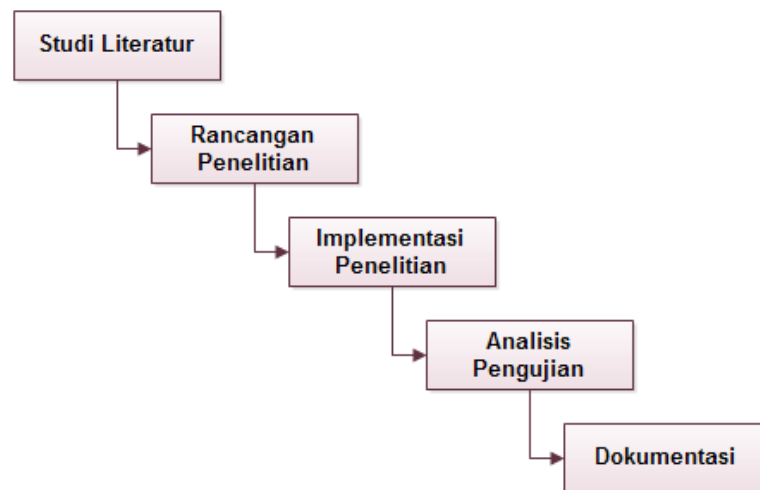
Merupakan bagian dari kasus yang merupakan sub proses dalam pembuatan suatu barang atau dalam suatu proses tertentu. Misal pada log data (E):

$$L = [< a, c, d >^{45}, < b, c, d >^{42}, < a, c, e >^{38}, < b, c, e >^{22}]$$

Dalam log data tersebut: Terdapat 5 aktivitas yaitu {a, b, c, d, e}.

BAB III METODE PENELITIAN

Bab ini memaparkan tentang metodologi penelitian yang digunakan pada penelitian ini. Adapun alur penelitian terdiri dari 5 tahap, meliputi (1) studi literatur, (2) rancangan penelitian, (3) implementasi penelitian, (4) analisis pengujian, (5) dokumentasi sistem. Ilustrasi alur metodologi penelitian ditunjukkan pada Gambar 3.1.



Gambar 3.1 Alur Metodologi Penelitian

Penjelasan dari tahapan metode penelitian pada Gambar 3.1 akan diterangkan secara terperinci pada sub bab berikutnya.

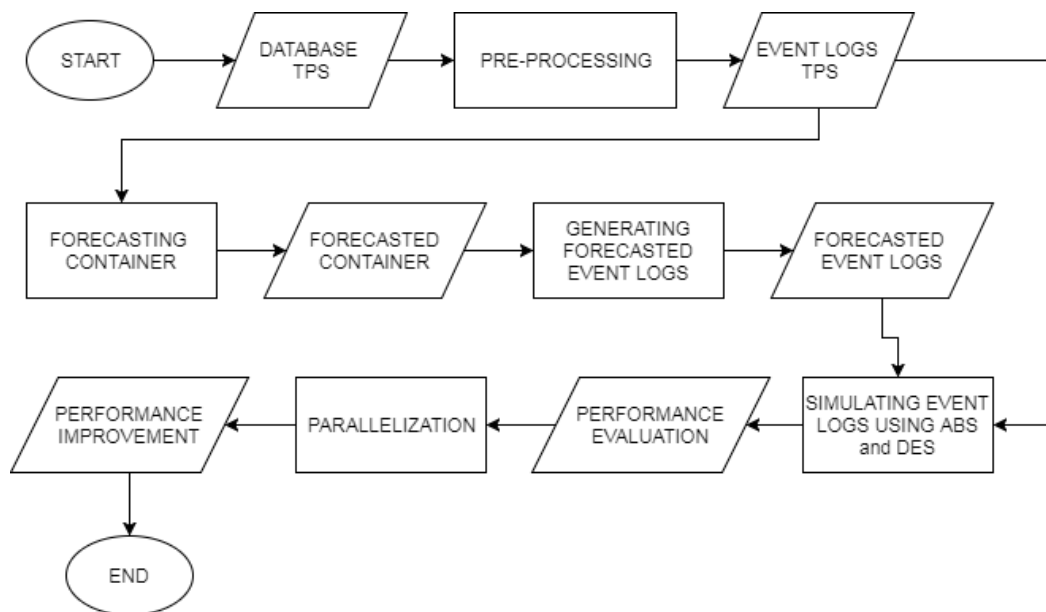
3.1 Studi Literatur

Penelitian selalu diawali dengan proses pengkajian yang berkaitan dengan topik penelitian yang diambil. Pada penelitian ini, referensi yang digunakan adalah literature-literatur yang berkaitan dengan simulasi berbasis agen untuk proses bisnis, terutama yang diterapkan pada proses bisnis multi organisasi (*cross organizations*). Dari tahap ini, diharapkan dapat memberikan gambaran lengkap dan memberikan dasar kontribusi pembuatan kerangka kerja untuk melakukan simulasi berbasis agen dalam upaya untuk meningkatkan proses bisnis pada terminal petikemas dengan studi kasus adalah proses bisnis impor barang di

PT. Terminal Petikemas Surabaya (PT.TPS). Pada tahap ini studi literatur dilakukan secara sistematis (*Systematic Literature Review*). Literatur diambil dari jurnal dan konferensi internasional yang dipublikasi melalui portal *sciencedirect*, IEEE atau *springer*. Jurnal yang diambil dengan mengambil kata kunci terkait dengan simulasi berbasis agen untuk proses bisnis, proses bisnis multi organisasi, analisis performa pada proses bisnis, *generating* log data serta paralelisasi proses bisnis.

3.2 Rancangan Penelitian

Kerangka kerja yang diusulkan pada penelitian ini terbagi atas 6 bagian yaitu: (1) Mendapatkan database; (2) transformasi menjadi log data; (3) *forecasting* jumlah container; (4) membentuk *forecasted* log data; (5) simulasi berbasis agen dan simulasi *discrete-event* (6) evaluasi performa (7) Paralelisasi proses bisnis. Alur rancangan kerangka kerja yang diusulkan pada penelitian ini dapat dilihat pada Gambar 3.2.



Gambar 3.2 Rancangan Penelitian

3.2.1 Pre-Processing

Untuk mendapatkan log data, peneliti melakukan pengambilan data secara langsung di tempat studi kasus, yaitu PT. Terminal Petikemas Surabaya (PT. TPS). Data yang terkumpul dalam database adalah data mentah yang membutuhkan pemrosesan lebih lanjut melalui pre-proses agar log data dapat digunakan sesuai dengan kebutuhan simulasi berbasis agen.

CONTAINER_NO	Start Time	End Time	Activity	CC_TT_No	HT_No	Ves_ID	NAMA_IMP
TRHU1782109		12/06/2015 17:15:03	Discharge	004	173	AJAI001	CV. CIPTA KARYA MAKMUR
WHLU8040720		12/06/2015 17:45:28	Discharge	003	212	AJAI001	CV. TRI JAYA MAKMUR
TRHU1782109	12/06/2015 17:49:50	02/07/2015	Yard	004	173	AJAI001	CV. CIPTA KARYA MAKMUR
WHLU8040720	12/06/2015 18:09:39	09/07/2015	Yard	003	212	AJAI001	CV. TRI JAYA MAKMUR
WHLU2801550		12/06/2015 20:36:28	Discharge	004	216	AJAI001	
WHLU2801550	12/06/2015 20:52:13	24/07/2015	Yard	004	216	AJAI001	
TRLU8819050		21/06/2015 19:06:29	Discharge	003	208	ALDI003	CV.YUDHA SOLUSI PRATAMA
DFSU1964638		21/06/2015 19:08:04	Discharge	003	208	ALDI003	PT. LAUTAN LUAS TBK
DFSU1964941		21/06/2015 19:14:21	Discharge	003	212	ALDI003	PT. JAKARANA TAMA
DFSU7474140		21/06/2015 19:19:49	Discharge	004	221	ALDI003	PT. GALANGCITRAMITRA MAJUMAPAN
DFSU1965043		21/06/2015 19:32:48	Discharge	004	230	ALDI003	PT. JAKARANA TAMA
CMAU2136921		21/06/2015 19:38:10	Discharge	004	187	ALDI003	PT. WUJAYA INDONESIA MAKMUR BICYCLE INDUSTRIES
TEMU2611128		21/06/2015 19:41:23	Discharge	004	217	ALDI003	PT. JINDAL STAINLESS INDONESIA
TRLU8819050	21/06/2015 19:25:02	02/07/2015	Yard	003	208	ALDI003	CV YUDHA SOLUSI PRATAMA
DFSU1964638	21/06/2015 19:26:30	06/07/2015	Yard	003	208	ALDI003	PT. LAUTAN LUAS TBK
DFSU7474141	21/06/2015 19:31:53	01/07/2015	Yard	004	221	ALDI004	PT. GALANGCITRAMITRA MAJUMAPAN
DFSU1965043	21/06/2015 19:46:18	30/06/2015	Yard	004	230	ALDI003	PT. JAKARANA TAMA
BSIU3004563		21/06/2015 19:57:03	Discharge	004	173	ALDI003	PT. GUDANG GARAM TBK

Gambar 3.3 Contoh Log data Proses Impor pada PT. PTS

Contoh log data ditunjukkan pada Gambar 3.3, adapun atribut yang telah disediakan pada log data adalah nomor kasus (*case id*), nomor container, waktu mulai, waktu selesai, dan aktivitas yang dijalankan. Selain itu atribut tambahan yang tersedia adalah nomor mesin crane, nomor truk untuk mengangkut container, nomor kapal dan *customer*. Namun, log data yang tersedia belum cukup digunakan pada simulasi berbasis agen, karena membutuhkan atribut tambahan untuk agen, yaitu pengirim pesan (*sender*), pesan (*message*) dan penerima pesan (*receiver*). Oleh karena itu, dilakukan proses transformasi pada log data dengan menambahkan atribut *sender*, *message* dan *receiver* pada log data. Ketiga atribut tersebut yang dalam proses selanjutnya berperan untuk digunakan pada setiap agen.

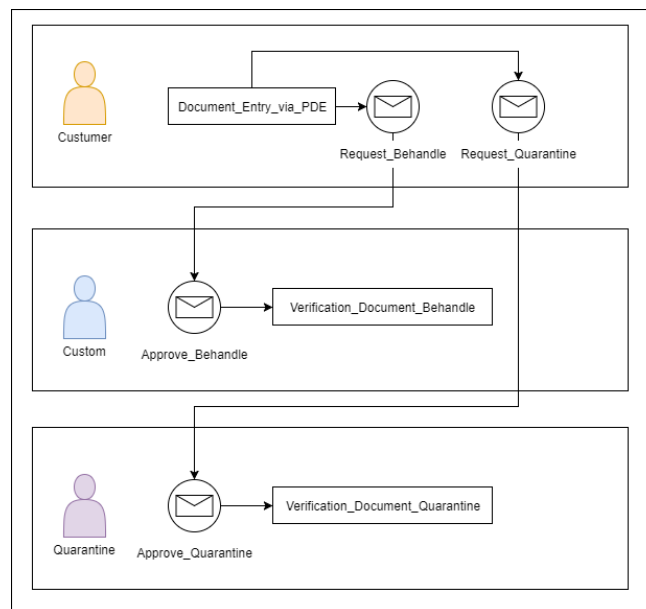
Kemudian, pada tahapan selanjutnya, dilakukan pemodelan proses bisnis untuk menggambarkan proses impor barang di PT. TPS. Untuk memodelkan proses bisnis, yang perlu dilakukan adalah melakukan survei lapangan untuk memastikan bagaimana *standart operating procedure* (SOP) untuk menjalankan proses bisnis impor barang tersebut. Pemodelan proses bisnis dibagi menjadi level

0 dan level 1, level 0 menangani proses secara umum. Proses bisnis yang terdapat pada level 0 adalah

1. *Pre Clearance*
2. *Discharge*
3. *Yard Operation*
4. *Custom Inspection*
5. *Quarantine Inpection*
6. *Delivery*

Adapun organisasi yang terlibat dalam proses bisnis impor barang dibagi menjadi 4 organisasi. Keempat organisasi tersebut untuk selanjutnya disebut sebagai agen dalam simulasi berbasis agen, antara lain:

1. *Customer*
2. PT. Terminal Petikemas Surabaya (TPS)
3. Karantina
4. Bea Cukai



Gambar 3.4 Komunikasi antar agen

Gambar 3.4 menunjukkan gambaran komunikasi yang dilakukan beberapa agen. *Customer* mengirimkan *message request_behandle* untuk *custom* dan

request_quarantine untuk karantina. Agar sesuai dengan kebutuhan simulasi berbasis agen, perlu diketahui bahwa terdapat organisasi yang memiliki karakteristik *asynchronous* sehingga diduga akan mempengaruhi *dwelling time*, yaitu karantina dan bea cukai.

1. Karantina

Pihak karantina memiliki karakteristik yang *asynchronous*. Sebelum pihak karantina melakukan proses, diharuskan menerima message terlebih dahulu dari pihak *customer*, yakni *requestQuarantine*. Pihak karantina tidak memiliki jadwal untuk mengatur kapan permintaan karantina oleh *customer* akan diterima. Hal ini mengakibatkan kontainer menumpuk pada *yard* untuk menunggu proses karantina. Dengan demikian, pola komunikasi *asynchronous* ini mempengaruhi meningkatnya *dwelling time* tiap container.

2. Bea Cukai

Pihak bea cukai juga memiliki karakteristik yang *asynchronous*. Bea cukai akan menerima message *requestBehandle* dari *customer*. Namun, bea cukai tidak memiliki jadwal yang baku untuk menerima permintaan *handle* dari *customer*. Pola komunikasi *asynchronous* ini mengakibatkan kontainer menumpuk pada *yard* untuk menunggu proses *handle*.

Berikut ini adalah prosedur standar yang dilakukan pada proses bisnis impor barang di PT. TPS mengacu pada penelitian di lapangan:

Prosedur layanan pembongkaran petikemas (*discharge*)

1. Perencanaan ; pelanggan harus melengkapi dokumen :
 - a. *Master Cable*
 - b. CVIA (*Container Vessel Identification Advice* = Pemberitahuan Identifikasi Kapal Petikemas)
 - c. *Statement of Fact* (Surat Pernyataan Keadaan)
 - d. *Statement Letter* (email *baplie* file)
 - e. *Import Summary List* (ISL = Daftar Ringkasan Impor)
 - f. *Dangerous Cargo List* (Daftar Kargo Berbahaya)

- g. *Approval from Harbor Master* (Surat Ijin dari Syahbandar)
 - h. *Reefer List* (Daftar Reefer)
 - i. *Crane Sequence List* (Daftar Urutan Crane)
 - j. *Discharge Stowage Plan* (Rencana Penyimpanan Pembongkaran)
 - k. *Discharge Bay Plan* (Rencana Bay Pembongkaran)
 - l. *Manifest*
 - m. *Special Cargo List* (Daftar Kargo Khusus)
2. *Yard and Berth Planning Sub-department* (Sub-departemen Perencanaan Lapangan dan Dermaga) memeriksa dokumen. Mereka mengadakan rapat harian, bersama dengan Departemen Teknik, dengan Perusahaan Pelayaran, untuk merencanakan jadwal layanan penanganan petikemas.
 3. *Vessel Berth Planning Sub-department* (Sub-departemen Perencanaan Lapangan dan Dermaga) memproses rencana pembongkaran ke dalam sistem komputer berdasarkan data yang dikirimkan oleh Perusahaan Pelayaran lewat email, dan mencetak *Discharge List* (Daftar Pembongkaran) dan menyerahkannya kepada *Berth Operations* (Operasi Dermaga).
 4. Berdasarkan *Discharge List* (Daftar Pembongkaran), *Berth Operations Superintendent* (Superintenden Operasi Dermaga) memerintahkan Operator CC, lewat Petugas *Tally* Dermaga, untuk membongkar petikemas dari atas kapal dan memuatnya ke atas *chassis Head Truck*, dan membawanya ke Lapangan Penumpukan Petikemas, dan mengkonfirmasi posisi pembongkaran ke dalam sistem komputer (HHT/Teklogix)
 5. Setelah *Head Truck* tiba di Lapangan Penumpukan Petikemas, *Yard Operations Superintendent* (Superintenden Operasi Lapangan) memerintahkan Operator RTG, lewat Petugas *Tally* Lapangan, untuk menumpuk petikemas, dan mengkonfirmasi posisi petikemas ke dalam sistem komputer (HHT/Teklogix). Petugas *Tally* Lapangan memerintahkan pengemudi *Head Truck* untuk kembali ke Dermaga untuk mengambil petikemas selanjutnya yang akan dibongkar.
 6. Pada akhir shift, Petugas *Tally* Lapangan melaporkan hasil pekerjaan kepada Superintenden Operasi Lapangan, sedangkan Petugas *Tally*

Dermaga melaporkan hasil pekerjaan kepada Superintenden Operasi Dermaga.

Prosedur Pemeriksaan Bea Cukai (*Behandle*)

1. Pelanggan menyerahkan Surat Permohonan *Behandle* Barang kepada TPS lewat Petugas Layanan Administrasi dilengkapi dengan Dokumen Asli, Warkat Dana, dan Perintah Pengeluaran.
2. Petugas Administrasi memeriksa dan mencetak Job Order dan menyerahkan kepada Pelanggan, dengan salinan Perintah Pengeluaran, dan menyerahkan 2 lembar salinan kepada Petugas Layanan Operasi.
3. Pelanggan menyerahkan *Job Order* kepada *CFS Operations Assistant Manager*.
4. Asisten Manajer Operasi CFS atau Staff yang ditunjuk memeriksa dokumen dan menerbitkan *Container Movement Job* (Pekerjaan Pergerakan Petikemas) untuk menarik petikemas dari Lapangan Penumpukan ke CFS.
5. Setelah petikemas telah dipindahkan ke CFS, keadaan fisik petikemas akan diperiksa sebelum pemeriksaan *Behandle* dilakukan.
6. Setelah pengeluaran barang telah selesai dilakukan, laporan harus disiapkan dan diketahui oleh Petugas CFS, dan disetujui oleh Pelanggan.

Prosedur layanan pengeluaran petikemas (*delivery*)

1. Perencanaan pelanggan harus melengkapi dokumen:
 - a. Surat Permohonan Pengeluaran Petikemas
 - b. Surat Asli Perintah Pengeluaran (DO = Delivery Order)
 - c. Penyediaan Warkat Dana (Pembayaran di Depan) (masing-masing 4 lembar) untuk diserahkan kepada Import Service Staff (Petugas Layanan Impor).
 - d. SPPB = Surat Persetujuan Pengeluaran Barang dan Surat Pernyataan PP (Pencekalan dan Pencegahan) dari Bea Cukai
 - e. Surat Kuasa dari Importir

2. Petugas Layanan Impor mencetak CEIR/Job Order yang telah disetujui oleh *Import Superintendent* (Superintenden Impor). Lembar ke 1, 2, dan 3 CEIR diserahkan kepada Pelanggan. Pelanggan menyerahkan kepada pengemudi *Head Truck*.
3. Pengemudi *Head Truck* menuju ke *In-Gate* (Gerbang Masuk) dan menyerahkan *Job Order/CEIR* kepada *In-Gate Staff* (Petugas Gerbang Masuk).
4. *In-Gate Staff* mencetak *In-Gate Terminal Job Slip* berdasarkan *Job Order/CEIR* dan mengembalikan lembar ke 1 dan 2 kepada pengemudi *Head Truck*.
5. Pengemudi *Head Truck* menyerahkan *In-Gate Terminal Job Slip* dan *Job Order/CEIR* kepada Petugas *Tally Lapangan*.
6. Petugas *Tally Lapangan* memerintahkan Operator RTG untuk mengangkat petikemas dari Lapangan Penumpukan ke atas *chassis Head Truck* sesuai dengan posisi yang tercantum dalam *In-Gate Terminal Job Slip*.
7. Pengemudi *Head Truck* menerima *Job Order/CEIR* dan *In-Gate Terminal Job Slip* dari Petugas *Tally Lapangan* bergerak menuju *Out-Gate* (Gerbang Keluar) dan menyerahkan *In-Gate Terminal Job Slip* dan *Job Order/CEIR* lembar ke 3 kepada Petugas *Out-Gate*, dan Surat Pernyataan Pecekalan dan Pencegahan (PP) kepada Petugas Bea Cukai.
8. Petugas *Out-Gate* mengkonfirmasi nomor polisi *Head Truck* dan nomor referensi kerja *Head Truck* berdasarkan *In-Gate Terminal Job Slip* ke dalam sistem computer dengan dilampiri lembar ke 1 CEIR kepada pengemudi *Head Truck*.

3.2.2 *Forecasting* Jumlah Kontainer

Pada fase ini, dilakukan peramalan terhadap jumlah kontainer dimasa yang akan datang. Penelitian yang dilakukan oleh (Park et al., 2016) berhasil melakukan *forecasting* kontainer dan melakukan simulasi proses bisnis yang bersifat sinkron. Pada penelitian ini, kontribusi yang diunggulkan adalah melakukan *forecasting* jumlah kontainer dimasa yang akan datang, untuk

kemudian membentuk log data dimasa yang akan datang. Untuk melakukan peramalan terhadap jumlah kontainer. Digunakan beberapa metode yang populer untuk peramalan pada data time series, antara lain: *Simple exponential smoothing*, *double exponential smooting*, dan *moving average*.

Simple exponential smooting adalah metode yang menggunakan alpha konstan (α) antara 0 sampai 1 untuk menentukan bobot data sebelumnya. Semakin bobot (α) mendekati 1 yang ditugaskan pada metode ini, semakin banyak bobot yang diberikan pada data terbaru dibandingkan data yang terdahulu. Persamaan untuk *Simple exponential smooting* ditunjukkan pada Persamaan (1).

$$F_t = \alpha A_{t-1} + (1 - \alpha)F_{t-1} \quad (1)$$

F_{t-1} = Forecast result of last period

A_{t-1} = Actual data of last period

Kemudian, *double exponential smooting* adalah metode statistik yang berbasis pada smoothing eksponensial sederhana. Dalam eksponensial ganda, ada dua metode yang biasa digunakan. Metode pertama hanya menggunakan satu konstanta bobot yaitu alpha. Metode kedua menggunakan alpha dan beta sebagai konstanta bobot. Yang pertama juga disebut *smoothing eksponensial brown's*. *Double exponential smooting* dihitung menggunakan Persamaan (2).

$$F_t = \alpha_t + b_t$$

$$a_t = 2A_t - A''_t$$

$$b_t = \left(\frac{\alpha}{1 - \alpha}\right)(A_t - A''_t)$$

$$A_t = \alpha y_t + (1 - \alpha)A_{t-1}$$

$$A''_t = \alpha A_t + (1 - \alpha)A''_{t-1}$$

(2)

y_t = Actual data of the current period

F_t = Forecast for the current period

Metode selanjutnya adalah *moving average*. Metode ini adalah teknik *forecasting* yang menghitung data rata-rata beberapa periode sebelum perkiraan target. Perhitungan menggunakan *moving average* ditunjukkan pada Persamaan (3)

$$MA = \frac{1}{K}(Y_{t-1} + Y_{t-2} + \dots + Y_{t-K+1}) \quad (3)$$

Forecasting jumlah kontainer ditentukan menggunakan 3 metode tersebut. Namun pada fase selanjutnya, dipilih hasil *forecasting* dengan nilai error yang paling kecil. Data yang digunakan untuk *forecasting* adalah data jumlah kontainer pada Bulan Januari hingga Maret 2016 sebanyak 90 data. *Forecasting* ditentukan hanya untuk meramalkan kontainer dalam satuan hari. Didapatkan hasil kontainer selama 30 hari kedepan, yakni Bulan April 2016. Berdasarkan 90 data kontainer, 60 data dijadikan sebagai data training, 30 sisanya sebagai data testing.

3.2.3 Membentuk *Forecasted* Log data

Pada fase ini, dibentuk log data berdasarkan hasil *forecasting* jumlah kontainer dimasa yang akan datang. Adapun metode untuk membentuk log data berdasarkan hasil *forecasting* adalah :

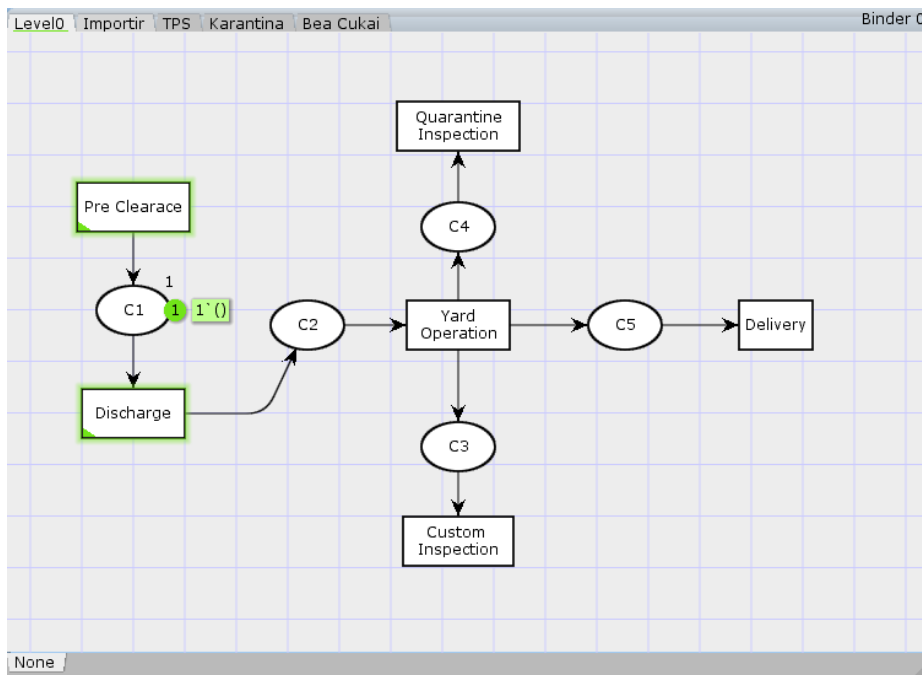
1. Menghitung *forecasted* jumlah kontainer setiap harinya selama satu bulan.
2. Menghitung frekuensi jumlah kontainer setiap *trace* pada setiap bulan.
3. Menghitung prosentase frekuensi tiap *trace*, *average* kontainer per hari per *trace*, dan standar deviasi kontainer per hari per *trace*.
4. Membangkitkan bilangan acak untuk jumlah kontainer tiap *trace* menggunakan distribusi normal dengan cara:
 $trace\ n = NORMINV(RAND());\ average\ container\ trace\ n ;\ st_dev\ container\ trace\ n).$
5. Jika hasil *forecasted* kontainer – jumlah kontainer hasil *generate* acak semua $trace > 0$, maka sisa jumlah kontainer akan di-*generate* ulang menggunakan prosentase frekuensi tiap *trace*.

6. Jumlah kontainer untuk *trace n* = prosentase *trace n* * sisa jumlah kontainer pada proses ke-5.
7. Menjumlahkan hasil masing-masing *trace* pada langkah ke-4 dan ke-6. Hasil dari proses ini adalah *generated trace*.
8. Membentuk atribut *case_id*, aktivitas/*message*, *sojourn time*, *waiting time* dan *execution time* untuk membentuk *forecasted* log data mengacu pada *generated trace*.
9. Menambahkan log data untuk *case_id*, aktivitas/*message*, *sojourn time*, *waiting time* dan *execution time* mengacu pada hasil jumlah kontainer pada *generated trace*.

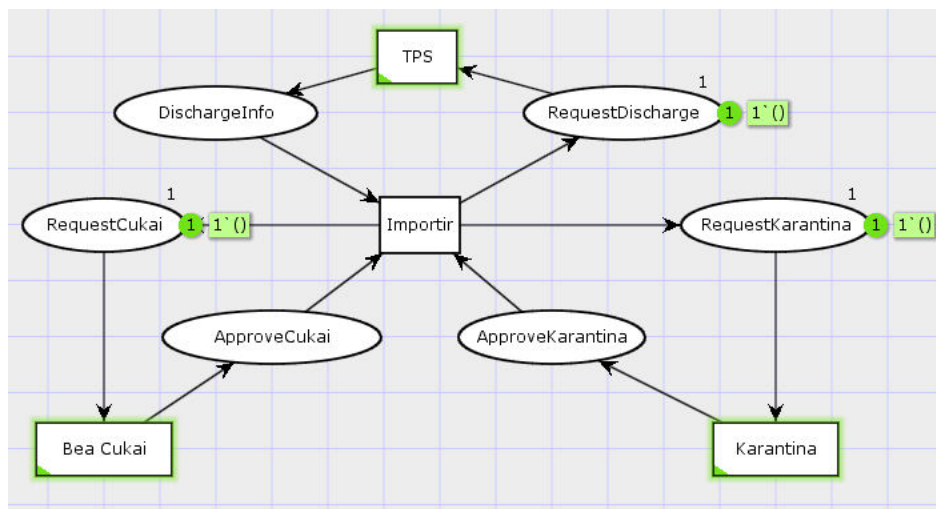
3.2.4 Simulasi Berbasis Agen

Pada tahap ini, aktivitas dan atribut komunikasi (*sender, message dan receiver*) telah terpetakan. Agen yang terlibat dalam simulasi adalah seluruh organisasi yang terlibat, yaitu *customer*, karantina, bea cukai dan PT. TPS. Dasar yang menjadi acuan simulasi adalah log data. Tidak semua data atribut pada log data digunakan dalam simulasi berbasis agen. Atribut pada data log yang mempengaruhi simulasi berbasis agen adalah aktivitas dan *message*, untuk membedakan keduanya maka digunakan ariginator pada aktivitas dan atribut komunikasi (*sender, receiver*) pada *message*. Kemudian untuk melakukan perhitungan evaluasi performa, dibutuhkan adanya *sojourn time, waiting time dan execution time*.

Gambar 3.5 menunjukkan gambaran simulasi menggunakan perkakas bantu CPN Tools sebatas untuk memodelkan proses bisnis impor barang pada PT. TPS di level 0. Sehingga proses bisnis dipadatkan menjadi 6 aktivitas. Selanjutnya Gambar 3.6 menunjukkan gambaran pertukaran *message* pada agen-agen yang terdapat pada proses bisnis impor barang di PT. TPS. Realitanya, penggunaan CPN Tools dikhususkan untuk simulasi discrete-event, sehingga dapat implementasi simulasi digunakan perkakas bantu khusus untuk simulasi berbasis agen.



Gambar 3.5 Gambaran simulasi untuk aktivitas level 0

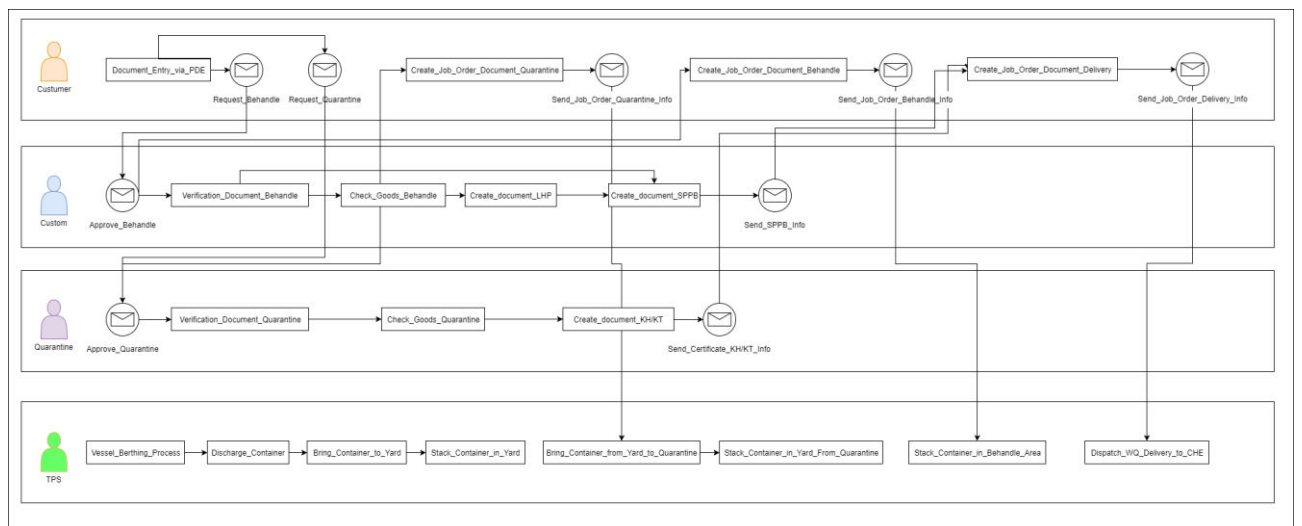


Gambar 3.6 Gambaran simulasi atribut komunikasi antar organisasi

Kemudian untuk melakukan simulasi berbasis agen, digunakan perangkat bantu *Anylogic 8.0 Personal Learning Edition*. Perangkat bantu ini dipilih karena mendukung untuk dilakukan simulasi berbasis agen. Selain itu, memungkinkan juga untuk melakukan simulasi *multi method*. Oleh karenanya, pada implementasinya nanti, simulasi berbasis agen dapat dibandingkan dengan

simulasi discrete-event menggunakan perkakas bantu yang sama. Langkah-langkah untuk melakukan simulasi berbasis agen adalah

- a. Membuat model proses bisnis multi organisasi berdasarkan acuan SOP pada Gambar 3.7.
- b. Membuat agen sebagai representasi organisasi
- c. Konfigurasi *message* antar agen
- d. Konfigurasi log data sebagai data input
- e. Menyiapkan output log simulasi



Gambar 3.7 Model Koreografi Proses Impor di PT. TPS

3.2.5 Evaluasi Performa

Pada tahap ini, dilakukan analisis performa berdasarkan hasil simulasi. Mengacu pada penelitian terdahulu oleh (Rozinat et al., 2008), performa didapatkan melalui perhitungan waktu tunggu (*waiting time*), waktu eksekusi (*execution time*) dan *sojourn time*.

- a. Waktu eksekusi (*execution time*) adalah waktu yang dibutuhkan selama berlangsungnya perintah eksekusi dari awal mulai proses hingga selesainya proses.

- b. Waktu tunggu (*waiting time*) adalah waktu yang dibutuhkan oleh suatu proses selama menunggu di *ready queue*.
- c. Waktu sojourn (*sojourn time*) adalah total waktu untuk keseluruhan pelaksanaan proses dari awal hingga akhir dan termasuk *waiting time*, atau dengan kata lain sojourn time adalah *execution time* ditambahkan dengan *waiting time*.

Pada penelitian terdahulu, metode untuk menghitung performa didasarkan pada simulasi *discret-event*. Artinya, simulasi bergantung pada aktivitas proses bisnis yang direpresentasikan dalam bentuk log data. Proses bisnis pada simulasi *discrete-event* hanya memiliki single organisasi. Sehingga tidak terdapat *message* untuk komunikasi antar agen ketika simulasi dijalankan.

```

Input: event log
Output: sojourn time (st), execution time (et), waiting time (wt)

Inisialisasi case_id
for iter = 1 to N_iter do
  for case_id to Ncase_id do
    if label == activity
      et_activityN = (t2_activityN - t1_activityN)
      wt_activityN = (t1_activityN+1 - t2_activityN)
      st_activityN = wt_activityN + et_ActivityN
    end
  end
end

```

Gambar 3.8 Algoritma evaluasi performa pada descrite-event simulation

Jika menggunakan algoritma pada Gambar 3.8, maka tidak dapat diterapkan pada kasus yang sedang diteliti saat ini. Oleh karena itu, penelitian ini mengusulkan untuk membuat metode yang modifikasi agar dapat menghitung performa multi organisasi yang bersifat *asynchronous*. Oleh karena itu, memungkinkan untuk memilah antara *message* dan aktivitas pada log data yang hanya memiliki *single time stamp*.

```

Input: event log
Output: sojourn time (st), execution time (et), waiting time (wt)

Inisialisasi case_id
for iter = 1 to Niter do
    for caseid to Ncaseid do
        if label == message
            st_message = t_message+.....n
        elseif label == activity
            st_activity = t_activity+.....n
        end
    end
end

hitung
total_st = st_message + st_activity
average st_message ; average st_activity
st_dev st_message ; st_dev st_activity

if label ==message
then
    et_message=RandomNormal(min st_message, st_dev
message)
    wt_message= st_message-et_message

if label == activity
then
    et_activity=RandomNormal(min st_activity, st_dev
activity)
    wt_activity= st_activity-et_activity

```

Gambar 3.9 Metode modifikasi untuk menangani *single time stamp*

Gambar 3.9 menunjukkan usulan metode modifikasi untuk memperoleh *execution time* dan *waiting time* pada aktivitas/*message* pada *single time stamp* log data. Penentuan *execution time* dan *waiting time* dilakukan per aktivitas/*message* mengacu pada hasil *sojourn time* tiap aktivitas/*message*. Berikut ini adalah metode untuk membentuk *execution time* dan *waiting time*:

- a. Tentukan nilai minimum (MIN) *sojourn time* aktivitas/*message* n
- b. Tentukan standar deviasi dari *sojourn time* untuk aktivitas/*message* n
- c. *Execution time* = NormalRandom(MIN *sojourn time* aktivitas/*message* n, standar deviasi dari *sojourn time* untuk aktivitas/*message* n)

$$d. \text{Waiting time} = \text{Sojourn time} - \text{execution time}$$

Metode yang diusulkan didasarkan bahwa nilai minimum *sojourn time* adalah nilai *execution time* yang memiliki *waiting time* = 0, sehingga *sojourn time* pasti akan memiliki *execution time* dan *waiting time* tidak sama dengan 0 ketika *sojourn time* lebih dari nilai minimumnya. Kemudian, pembentukan bilangan random berbasis distribusi normal berdasarkan 2 parameter, yaitu nilai minimum *sojourn time* dan standar deviasi dari *sojourn time*. Adapun syarat pembentukan bilangan random tersebut adalah (a) *execution time* tidak boleh kurang atau sama dengan 0; (b) *execution time* tidak boleh lebih besar dari *sojourn time*.

BAB IV HASIL PENELITIAN DAN PEMBAHASAN

4.1 Hasil Penelitian

4.1.1 Lingkungan Uji Coba

Data log diujikan menggunakan komputer dengan spesifikasi *processor Intel® Core™ i3 CPU M380 @ 2.53 GHz*, memori 4 GB, sistem operasi yang digunakan adalah *Windows 10 Professional 64-bit* dan bahasa pemrograman yang digunakan adalah Java dengan bantuan perkakas bantu *Anylogic Personal Learning Edition 8.0* untuk *Agent Based Simulation* dan *Discrete Event Simulation*.

4.1.2 Pre-Processing

Data yang digunakan adalah log data Terminal Petikemas Surabaya, Indonesia. Log data pada terminal petikemas mewakili rekaman proses bisnis yang terjadi di terminal petikemas pada rentang waktu yang ditentukan. Peneliti membatasi rentang waktu log data yang digunakan sebagai data penelitian adalah selama periode Bulan Januari hingga Bulan Maret Tahun 2016. Kemudian data log bulan selanjutnya akan di-*generate* untuk mendapatkan *forecasted* log data. Sehingga log data yang digunakan adalah *existing* log data yang menandakan kondisi log data saat ini yakni pada Bulan Januari hingga Maret 2016 dan *forecasted* log data yang menunjukkan kondisi log data di masa mendatang, yakni pada Bulan April 2016.

Log data memiliki beberapa atribut dasar seperti id kasus, aktivitas dan *time stamp*. Namun, khusus untuk log data yang digunakan dalam penelitian ini, digunakan atribut tambahan yang belum pernah digunakan pada penelitian terdahulu, yaitu penambahan atribut pesan (*message*). Berbeda dengan atribut aktivitas yang memiliki *originator*, atribut pesan memiliki *sender* dan *receiver* untuk membedakan dengan atribut aktivitas. Peneliti mengusulkan bahwa log data yang mengandung atribut pesan menandakan adanya aktivitas yang dijalankan oleh beberapa organisasi yang berbeda.

Log data pada terminal petikemas dibatasi hanya pada proses bisnis untuk mengimpor barang di terminal peti kemas. Proses impor barang di terminal petikemas melibatkan beberapa organisasi, antara lain *customer*, PT. TPS, bea cukai dan karantina. Keberadaan pesan menjadi penanda adanya komunikasi antar organisasi, atau dijadikan *trigger* karena keberhasilan aktivitas dipengaruhi oleh pesan antar organisasi. Sebelumnya telah disepakati bahwa jumlah organisasi yang terlibat dalam proses bisnis impor barang di Terminal Petikemas Surabaya adalah 4 organisasi. Oleh karena itu, ketika disimulasikan menggunakan simulasi berbasis agen, organisasi tersebut berubah menjadi 4 agen yang saling berkomunikasi. Pesan adalah pemicu untuk menjalankan aktivitas pada agen lain. Jika tidak ada pesan, maka aktivitas di agen lain tidak bisa berjalan.

Pada kasus yang sedang diteliti saat ini, atribut yang digunakan dalam log data adalah *case id*, *sender*, *receiver*, *activity*, *message*, *originator*, *single timestamp* dan *cost*. *Pre-processing* dilakukan untuk melakukan transformasi dari bentuk database menjadi bentuk log data. Hal ini dilakukan karena log data adalah *input* untuk implementasi metode sedangkan data yang tersedia pada Terminal Petikemas Surabaya adalah dalam bentuk *database*. Bagian yang terpenting dalam *database* yang diambil adalah kolom yang memiliki tanggal, karena tanggal tersebut digunakan sebagai timestamp untuk atribut aktivitas pada log data. Intervensi untuk transformasi database menjadi log data meliputi intervensi waktu (*time*) dan biaya (*cost*). Tabel intervensi untuk *time* dan *cost* ditunjukkan pada Lampiran A dan B.

Pre-processing selanjutnya dilakukan untuk memperoleh nilai *sojourn time* dan *execution time* pada setiap aktivitas. *Sojourn time* adalah waktu yang ditempuh untuk menjalankan aktivitas termasuk waktu tunggu sebelum aktivitas dijalankan. Maka secara sederhana, *sojourn time* adalah waktu eksekusi (*execution time*) ditambahkan dengan waktu tunggu (*waiting time*). Untuk memperoleh *sojourn time*, dilakukan perhitungan dengan mencari selisih tanggal antar aktivitas. Namun, perhitungan tersebut tetap didasarkan atas petunjuk *expert* di Terminal Petikemas Surabaya. Tabel 4.1 menunjukkan dasar perhitungan *sojourn time* untuk setiap aktivitas dan pesan.

Tabel 4.1. Dasar perhitungan sojourn time pada tiap aktivitas

AKTIVITAS/MESSAGE	SOJOURN TIME
<i>Approve_Behandle</i>	Tanggal <i>Approve_Behandle</i> – Tanggal <i>Request_Behandle</i>
<i>Approve_Quarantine</i>	Tanggal <i>Approve_Quarantine</i> – Tanggal <i>Request_Behandle</i>
<i>Bring_Container_from_Yard_to_Quarantine</i>	Tanggal <i>Bring_Container_from_Yard_to_Quarantine</i> – Tanggal <i>Send_Job_Order_Quarantine_Info</i>
<i>Bring_Container_to_Yard</i>	Tanggal <i>Bring_Container_to_Yard</i> – Tanggal <i>Discharge_Container</i>
<i>Bring_Container_from_Yard_to_Behandle</i>	Tanggal <i>Bring_Container_from_Yard_to_Behandle</i> – Tanggal <i>Send_Job_Order_Behandle_Info</i>
<i>Check_Goods_Behandle</i>	Tanggal <i>Check_Goods_Behandle</i> - <i>Stack_Container_in_Behandle_Area</i>
<i>Create_Job_Order_Document_Behandle</i>	Tanggal <i>Create_Job_Order_Document_Behandle</i> – Tanggal <i>Verification_Document_Behandle</i>
<i>Check_Container_before_Truck_out</i>	Tanggal <i>Check_Container_before_Truck_out</i> – Tanggal <i>Truck_Go_To_Gate_Out</i>
<i>Check_Goods_Quarantine</i>	Tanggal <i>Check_Goods_Quarantine</i> – Tanggal <i>Stack_Container_in_Quarantine_Area</i>
<i>Create_Job_Order_Document_Delivery</i>	Tanggal <i>Create_Job_Order_Document_Delivery</i> – tanggal <i>Send_SPPB_Info</i>
<i>Create_Job_Order_Document_Quarantine</i>	Tanggal <i>Create_Job_Order_Document_Quarantine</i> – Tanggal <i>Verification_Document_Quarantine</i>
<i>Create_document_KH/KT</i>	Tanggal <i>Create_document_KH/KT</i> – Tanggal <i>Check_Goods_Quarantine</i>
<i>Create_document_SPPB</i>	Tanggal <i>Create_document_SPPB</i> – Tanggal <i>Stack_Container_in_Yard_From_Behandle</i>
<i>Create_document_LHP</i>	Tanggal <i>Create_document_LHP</i> – Tanggal <i>Check_Goods_Behandle</i>
<i>Decide_Task_Before_Lift_Container</i>	Tanggal <i>Decide_Task_Before_Lift_Container</i> – Tanggal <i>Determining_Dry</i>
<i>Determining_Dry</i>	Tanggal <i>Determining_Dry</i> – Tanggal <i>Determine_Container_Type</i>
<i>Discharge_Container</i>	Tanggal <i>Discharge_Container</i> – Tanggal <i>Vessel_Berthing_Process</i>

<i>Dispatch_WQ_Delivery_to_CHE</i>	<i>Tanggal Dispatch_WQ_Delivery_to_CHE – Tanggal Truck_In</i>
<i>Document_Entry_via_PDE</i>	<i>Random antara 2-3 jam (Berdasarkan intervensi oleh pakar)</i>
<i>Determine_Container_Type</i>	<i>Tanggal Determine_Container_Type – Tanggal Dispatch_WQ_Delivery_to_CHE</i>
<i>Determining_Refeer</i>	<i>Tanggal Determining_Refeer - Tanggal Determine_Container_Type</i>
<i>Determining_Uncontainer</i>	<i>Tanggal Determining_Uncontainer – Tanggal Determine_Container_Type</i>
<i>Lift_on_Container_Truck</i>	<i>Tanggal Lift_on_Container_Truck – Tanggal Decide_Task_Before_Lift_Container</i>
<i>Prepare_Tools</i>	<i>Tanggal Prepare_Tools – Tanggal Decide_Task_Before_Lift_Container</i>
<i>Request_Behandle</i>	<i>Tanggal Request_Behandle – Tanggal Entry Document</i>
<i>Request_Quarantine</i>	<i>Tanggal Request_Quarantine - Tanggal Entry Document</i>
<i>Send_Job_Order_Behandle_Info</i>	<i>Tanggal Send_Job_Order_Behandle_Info – Tanggal Create_Job_Order_Document_Behandle</i>
<i>Send_Certificate_KH/KT_Info</i>	<i>Tanggal Send_Certificate_KH/KT_Info – Tanggal Create_document_KH/KT</i>
<i>Send_Job_Order_Delivery_Info</i>	<i>Tanggal Send_Job_Order_Delivery_Info – Tanggal Create_Job_Order_Document_Delivery</i>
<i>Send_SPPB_Info</i>	<i>Tanggal Send_SPPB_Info – Tanggal Create_document_SPPB</i>
<i>Send_Job_Order_Quarantine_Info</i>	<i>Tanggal Send_Job_Order_Quarantine_Info – Tanggal Create_Job_Order_Document_Quarantine</i>
<i>Stack_Container_in_Quarantine_Area</i>	<i>Tanggal Stack_Container_in_Quarantine_Area – Tanggal Bring_Container_from_Yard_to_Quarantine</i>
<i>Stack_Container_in_Yard</i>	<i>Tanggal Stack_Container_in_Yard – Tanggal Bring_Container_to_Yard</i>
<i>Stack_Container_in_Yard_From_Quarantine</i>	<i>Tanggal Stack_Container_in_Yard_From_Quarantine – Tanggal Send_Certificate_KH/KT_Info</i>
<i>Stack_Container_in_Behandle_Area</i>	<i>Tanggal Stack_Container_in_Behandle_Area – Tanggal Send_Job_Order_Behandle_Info</i>

<i>Stack_Container_in_Yard_From_Behandle</i>	Tanggal <i>Stack_Container_in_Yard_From_Behandle</i> – Tanggal <i>Bring_Container_from_Yard_to_Behandle</i>
<i>Truck_Go_To_Gate_Out</i>	Tanggal <i>Truck_Go_To_Gate_Out</i> – Tanggal <i>Lift_on_Container_Truck</i>
<i>Truck_Out</i>	Tanggal <i>Truck_Out</i> – Tanggal <i>Check_Container_before_Truck_out</i>
<i>Truck_in</i>	Tanggal <i>Truck_in</i> - Tanggal <i>Send_Job_Order_Delivery_Info</i>
<i>Unplug_Refeer_Cable</i>	Tanggal <i>Unplug_Refeer_Cable</i>
<i>Verification_Document_Behandle</i>	Tanggal <i>Verification_Document_Behandle</i> – Tanggal <i>Approve Behandle</i>
<i>Verification_Document_Quarantine</i>	Tanggal <i>Verification_Document_Quarantine</i> – Tanggal <i>Approve Quarantine</i>
<i>Vessel_Berthing_Process</i>	Random antara 5 – 12 jam (Berdasarkan intervensi oleh pakar)

Berdasarkan Tabel 4.1, tidak semua *sojourn time* pada aktivitas/*message* ditentukan berdasarkan selisih tanggal aktivitas sebelumnya. Terdapat juga aktivitas yang *sojourn time*-nya ditentukan berdasarkan intervensi oleh pakar, dikarenakan aktivitas tersebut merupakan awal proses pada organisasi tertentu. Misalkan, pada aktivitas *Vessel_Berthing_Process* merupakan awal aktivitas pada organisasi PT. TPS sehingga tidak memungkinkan untuk menghitung selisih dengan aktivitas sebelumnya. Kemudian, berdasarkan hasil perhitungan *sojourn time*, didapatkan nilai *sojourn time* tiap aktivitas/*message*. Oleh karena itu, dapat dihitung pula rata-rata (*average*) *sojourn time* tiap aktivitas/*message*. Kontribusi baru dalam penelitian ini adalah mengusulkan metode untuk membentuk *execution time* dan *waiting time* berdasarkan hasil *sojourn time*. Penentuan *execution time* dan *waiting time* dilakukan per aktivitas/*message* mengacu pada hasil *sojourn time* tiap aktivitas/*message*. Berikut ini adalah metode untuk membentuk *execution time* dan *waiting time*:

- a. Tentukan nilai minimum (MIN) *sojourn time* aktivitas/*message* n
- b. Tentukan standar deviasi dari *sojourn time* untuk aktivitas/*message* n

- c. $Execution\ time = NormalRandom(MIN\ sojourn\ time\ aktivitas/message\ n,$
standar deviasi dari $sojourn\ time$ untuk aktivitas/message n)
- d. $Waiting\ time = Sojourn\ time - execution\ time$

Metode yang diusulkan didasarkan bahwa nilai minimum $sojourn\ time$ adalah nilai $execution\ time$ yang memiliki $waiting\ time = 0$, sehingga $sojourn\ time$ pasti akan memiliki $execution\ time$ dan $waiting\ time$ tidak sama dengan 0 ketika $sojourn\ time$ lebih dari nilai minimumnya. Kemudian, pembentukan bilangan random berbasis distribusi normal berdasarkan 2 parameter, yaitu nilai minimum $sojourn\ time$ dan standar deviasi dari $sojourn\ time$. Adapun syarat pembentukan bilangan random tersebut adalah (a) $execution\ time$ tidak boleh kurang atau sama dengan 0; (b) $execution\ time$ tidak boleh lebih besar dari $sojourn\ time$. Jika tidak memenuhi syarat tersebut. Maka hasil pembentukan bilangan random tersebut tidak dapat digunakan untuk menghitung $waiting\ time$. Oleh karenanya, didapatkan $execution\ time$ dengan rentang tertinggi adalah $MIN\ sojourn\ time +$ maksimum standar deviasi dari aktivitas/message n, begitu pula sebaliknya. Namun pembentukan $execution\ time$ tetap didasarkan dengan syarat yang telah ditentukan.

$$- \text{Standar deviasi} \leftarrow MIN\ Sojourn\ Time \rightarrow + \text{Standar Deviasi}$$

Table 4.2 adalah hasil perolehan nilai $execution\ time$ dan $waiting\ time$ untuk setiap aktivitas/message mengacu pada pembentukan bilangan acak sebagaimana yang telah dijelaskan sebelumnya. Selain itu, ditambahkan pula perhitungan $cost$ yang diperoleh ketika intervensi database ke log data. Adapun untuk $sojourn\ time$, $execution\ time$ dan $waiting\ time$ memiliki satuan waktu dalam bentuk detik dan $cost$ memiliki satuan dalam bentuk dollar. Hasil yang telah diperoleh pada Tabel 4.2 adalah rata-rata $cost$, $sojourn\ time$, $execution\ time$, dan $waiting\ time$ untuk setiap aktivitas/message mengacu pada log data saat ini, yaitu log data pada Bulan Januari hingga Bulan Maret 2016. Hasil tersebut juga akan digunakan sebagai acuan untuk menentukan rata-rata dwelling time pada Terminal Petikemas Surabaya menggunakan simulasi berbasis agen untuk proses asinkron dan menggunakan simulasi *discrete-event* untuk proses sinkron.

Tabel 4.2 *cost, sojourn time (ST), execution time (ET), dan waiting time (WT)*

AKTIVITAS/MESSAGE	AVG COST	AVG ST	AVG ET	AVG WT
<i>Approve_Behandle</i>	1,85	177458,37	91063,22	86395,15
<i>Approve_Quarantine</i>	11,81	156190,63	82277,67	73912,96
<i>Bring_Container_from_Yard_to_Quarantine</i>	177,17	6348,12	2932,99	3415,13
<i>Bring_Container_to_Yard</i>	61,71	1078,12	767,58	310,54
<i>Bring_Container_from_Yard_to_Behandle</i>	465,76	14239,11	8156,98	6082,13
<i>Check_Goods_Behandle</i>	1104,64	3885,42	1868,76	2016,66
<i>Create_Job_Order_Document_Behandle</i>	465,54	179032,60	107667,73	71364,87
<i>Check_Container_before_Truck_out</i>	2,70	170,01	67,60	102,41
<i>Check_Goods_Quarantine</i>	354,89	3605,18	1542,72	2062,46
<i>Create_Job_Order_Document_Delivery</i>	40,59	46712,39	24435,42	22276,97
<i>Create_Job_Order_Document_Quarantine</i>	177,13	2780,39	1216,85	1563,55
<i>Create_document_KH/KT</i>	413,16	6579,01	2995,78	3583,23
<i>Create_document_SPPB</i>	2656,99	44463,16	23193,21	21269,95
<i>Create_document_LHP</i>	22445,20	6730,35	3520,59	3209,76
<i>Decide_Task_Before_Lift_Container</i>	2,71	245,43	71,50	173,93
<i>Determining_Dry</i>	18,04	329,55	106,64	222,92
<i>Discharge_Container</i>	502,81	599,87	438,71	161,16
<i>Dispatch_WQ_Delivery_to_CHE</i>	1,20	162,02	60,43	101,59
<i>Document_Entry_via_PDE</i>	0,00	7201,26	5268,59	1932,67
<i>Determine_Container_Type</i>	2,70	462,39	137,71	324,68
<i>Determining_Refeer</i>	22,85	254,84	112,10	142,74
<i>Determining_Uncontainer</i>	18,30	116,52	98,22	18,30
<i>Lift_on_Container_Truck</i>	19,49	368,09	117,25	250,85
<i>Prepare_Tools</i>	54,29	809,23	592,90	216,32
<i>Request_Behandle</i>	0,00	89,95	63,73	26,22
<i>Request_Quarantine</i>	0,00	179,85	140,87	38,97
<i>Send_Job_Order_Behandle_Info</i>	0,00	119,74	87,30	32,44
<i>Send_Certificate_KH/KT_Info</i>	11,82	120,08	87,84	32,24

<i>Send_Job_Order_Delivery_Info</i>	0,00	145,50	112,54	32,95
<i>Send_SPPB_Info</i>	0,00	99,94	66,81	33,13
<i>Send_Job_Order_Quarantine_Info</i>	11,79	96,66	64,64	32,02
<i>Stack_Container_in_Quarantine_Area</i>	177,09	203,55	71,54	132,01
<i>Stack_Container_in_Yard</i>	23,36	600,19	438,89	161,30
<i>Stack_Container_in_Yard_From_Quarantine</i>	177,34	4204,46	1588,79	2615,67
<i>Stack_Container_in_Behandle_Area</i>	468,89	14254,76	8260,85	5993,90
<i>Stack_Container_in_Yard_From_Behandle</i>	459,68	212,01	83,75	128,26
<i>Truck_Go_To_Gate_Out</i>	2,70	6646,93	2363,98	4282,95
<i>Truck_Out</i>	13,54	179,28	72,34	106,93
<i>Truck_in</i>	13,52	153168,08	68025,93	85142,15
<i>Unplug_Refeer_Cable</i>	14,92	601,42	312,72	288,70
<i>Verification_Document_Behandle</i>	6,78	359,34	245,64	113,70
<i>Verification_Document_Quarantine</i>	177,33	5025,35	2242,03	2783,32
<i>Vessel_Berthing_Process</i>	54330,79	64789,07	12449,86	43200,00

4.1.3 Forecasting Jumlah Container

Peramalan (*forecasting*) adalah proses untuk memperkirakan beberapa kebutuhan di masa datang yang meliputi kebutuhan dalam ukuran kuantitas, kualitas, waktu dan lokasi yang dibutuhkan dalam rangka memenuhi permintaan barang ataupun jasa. Pada bab ini dijelaskan *forecasting* yang dilakukan untuk mengetahui perkiraan jumlah container yang masuk di Terminal Petikemas Surabaya dimasa yang akan datang. Berdasarkan log data yang telah diperoleh, 1 *case id* merepresentasikan 1 kontainer. Misal, jika dalam satu bulan diperoleh 23.000 kontainer, otomatis didapatkan bahwa dalam bulan itu terdapat 23.000 kontainer yang masuk. Mengacu pada data log, didapatkan jumlah kontainer pada Bulan Januari, Februari dan Maret 2017 yang ditunjukkan pada Tabel 4.3. Data kontainer bulanan dipecah menjadi data kontainer dalam rentang waktu harian. Alasannya, jika dilakukan *forecasting* bulanan, maka hanya ada 3 bulan sebagai data training. Hal ini tentu akan membuat hasil *forecasting* menjadi kurang

maksimal. Oleh karena itu, data jumlah kontainer selama 3 bulan yang tersedia dipecah menjadi data jumlah kontainer dengan rentang waktu harian, sehingga didapatkan data kontainer sebanyak 92 hari. Untuk melakukan *forecasting*, data kontainer selama 92 hari dibagi menjadi 2 bagian, 60 data kontainer digunakan sebagai data *training* dan sisanya digunakan sebagai data *testing*. Tabel 4.3 menunjukkan bahwa jumlah kontainer tertinggi terdapat pada tanggal 20 pada Bulan Februari sebanyak 2104 penerimaan kontainer, sedangkan terdapat pula beberapa tanggal yang pada hari itu penerimaan kontainer sama dengan 0. Hal ini menunjukkan bahwa pada data log hasil transformasi, memang tidak ditemukan ada proses bongkar muat kontainer (*discharge*) dari kapal sandar. Kosongnya jumlah kontainer pada beberapa tanggal tentu akan mempengaruhi *mean square error* (MSE) dan *error rate* (ER) pada hasil *forecasting*.

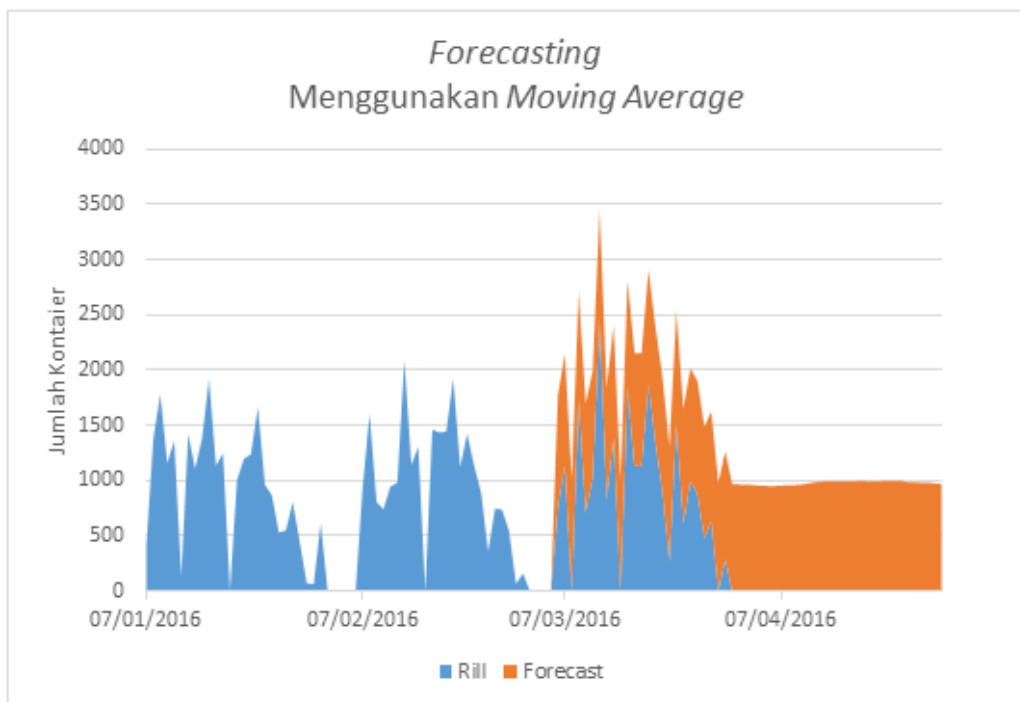
Table 4.3 Jumlah kontainer harian

Tanggal	Jumlah Kontainer	Tanggal	Jumlah Kontainer
01/01/2016	873	16/02/2016	0
02/01/2016	1964	17/02/2016	1533
03/01/2016	936	18/02/2016	1525
04/01/2016	1221	19/02/2016	1629
05/01/2016	620	20/02/2016	2104
06/01/2016	1226	21/02/2016	1210
07/01/2016	857	22/02/2016	1640
08/01/2016	1381	23/02/2016	1357
09/01/2016	1802	24/02/2016	1192
10/01/2016	1248	25/02/2016	630
11/01/2016	1370	26/02/2016	1144
12/01/2016	127	27/02/2016	1345
13/01/2016	1448	28/02/2016	1342
14/01/2016	1124	29/02/2016	939
15/01/2016	1414	01/03/2016	1070
16/01/2016	2001	02/03/2016	1287
17/01/2016	1203	03/03/2016	823
18/01/2016	1350	04/03/2016	1202
19/01/2016	0	05/03/2016	1516
20/01/2016	1058	06/03/2016	1846
21/01/2016	1327	07/03/2016	1488
22/01/2016	1305	08/03/2016	384
23/01/2016	1900	09/03/2016	1305

24/01/2016	1083	10/03/2016	820
25/01/2016	1047	11/03/2016	1041
26/01/2016	692	12/03/2016	2044
27/01/2016	896	13/03/2016	1368
28/01/2016	1355	14/03/2016	1389
29/01/2016	1374	15/03/2016	331
30/01/2016	1122	16/03/2016	1312
31/01/2016	1847	17/03/2016	941
01/02/2016	1513	18/03/2016	1451
02/02/2016	1121	19/03/2016	1565
03/02/2016	1435	20/03/2016	1415
04/02/2016	953	21/03/2016	939
05/02/2016	1239	22/03/2016	734
06/02/2016	1883	23/03/2016	1256
07/02/2016	1585	24/03/2016	728
08/02/2016	1651	25/03/2016	955
09/02/2016	839	26/03/2016	1061
10/02/2016	763	27/03/2016	725
11/02/2016	961	28/03/2016	476
12/02/2016	992	29/03/2016	304
13/02/2016	2104	30/03/2016	230
14/02/2016	1190	31/03/2016	62
15/02/2016	1361		

Setelah data jumlah kontainer disiapkan, selanjutnya dilakukan proses *forecasting*. Data jumlah kontainer yang dihasilkan melalui proses *forecasting* adalah jumlah kontainer pada Bulan April Tahun 2016 dengan rentang waktu harian. Sehingga, didapatkan jumlah kontainer yang masuk setiap harinya pada Bulan April Tahun 2016. Metode untuk *forecasting* menggunakan 3 metode yang berbeda, dengan maksud untuk membandingkan MSE dan ER pada masing-masing metode. Metode dengan nilai MSE dan ER yang paling kecil dapat dijadikan acuan untuk menggunakan hasil *forecasting* metode tersebut pada proses *generating trace* dan *generating log data*. 3 metode yang digunakan untuk melakukan *forecasting* adalah *moving average*, *simple exponential smoothing* dan *double exponential smoothing*. Ketiga metode tersebut kompatibel untuk digunakan dalam masalah *forecasting* pada data berbasis *time series* sebagaimana data jumlah kontainer yang tersedia untuk *forecasting* pada penelitian ini.

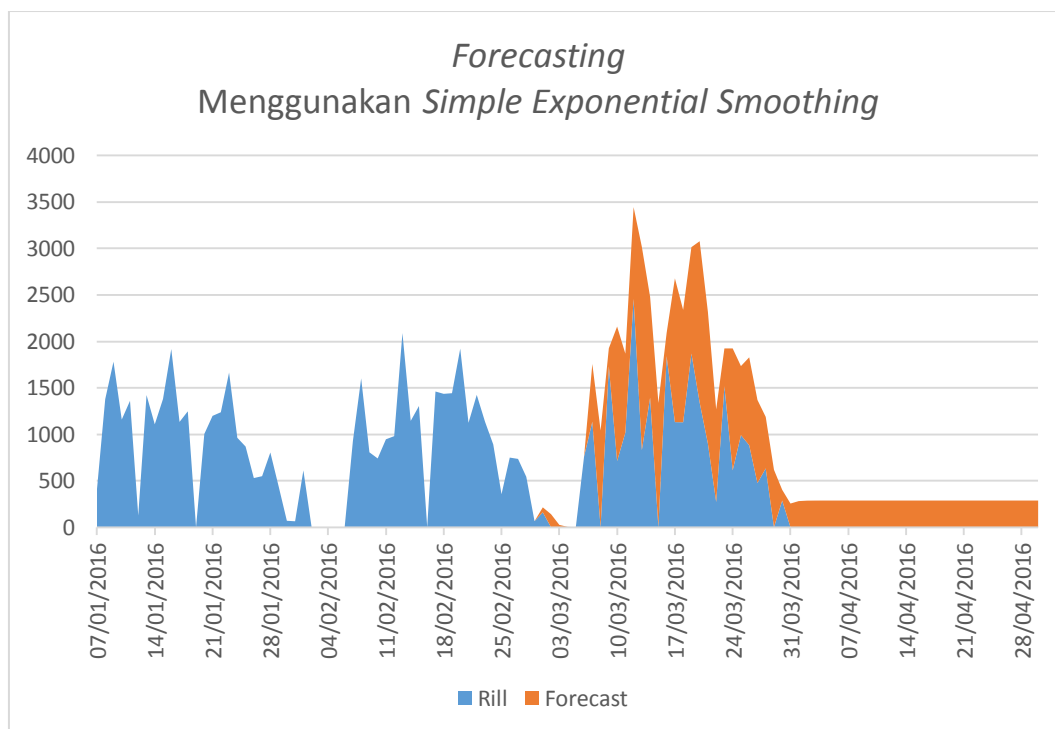
Proses *forecasting* pertama dilakukan menggunakan metode *moving average*, untuk menghasilkan jumlah kontainer tiap harinya selama Bulan April 2016. Perhitungan *forecasting moving average* menggunakan Persamaan 3 sebagaimana ditunjukkan Bab 3. Gambar 4.1 menunjukkan grafik hasil perhitungan *forecasting* menggunakan *moving average*. Grafik berwarna biru sebagai data rill dan grafik berwarna oranye sebagai hasil *forecasting*. Hasil *forecasting* menunjukkan bahwa pada Bulan April 2016, setiap harinya PT. TPS akan menerima kurang dari sama dengan 1000 kontainer. Hasil *forecasting* menggunakan *moving average* menunjukkan jumlah kontainer yang stabil dan tidak fluktuatif secara signifikan. Selanjutnya dihitung nilai MSE dan ER untuk metode *moving average* sehingga didapatkan nilai MSE adalah 253946,7 dan nilai ER sebesar 51,16618.



Gambar 4.1 *Forecasting* jumlah kontainer menggunakan *Moving Average*

Proses *forecasting* kedua menggunakan metode *simple exponential smoothing* untuk menghasilkan jumlah kontainer tiap harinya selama Bulan April 2016. Perhitungan *forecasting* metode ini menggunakan Persamaan 1 sebagaimana ditunjukkan Bab 3. Gambar 4.2 menunjukkan grafik hasil perhitungan *forecasting*

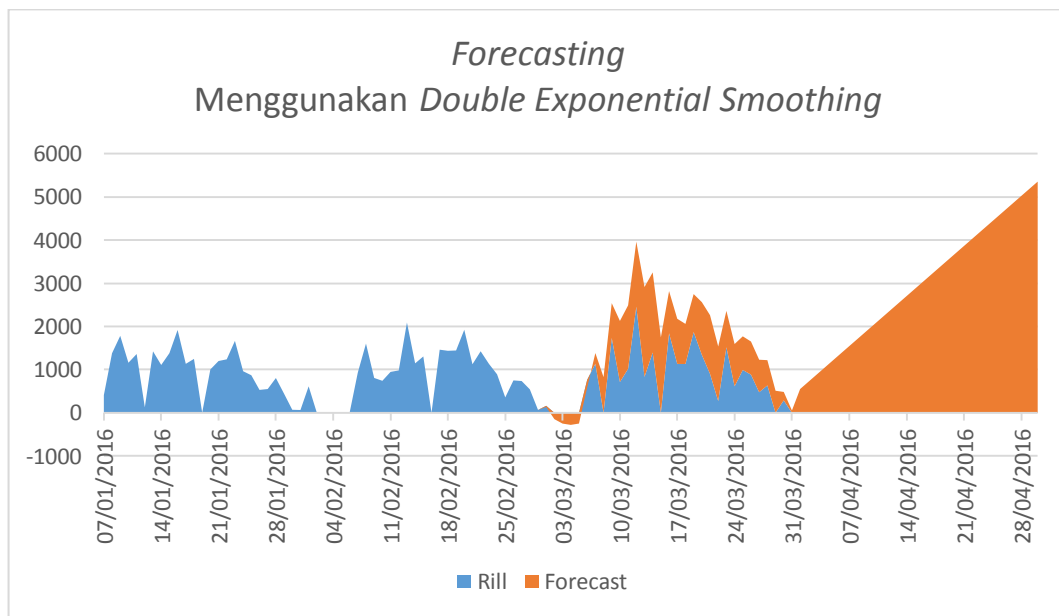
menggunakan *simple exponential smoothing*. Sama halnya dengan penjelasan sebelumnya, bahwa grafik berwarna biru sebagai data rill dan grafik berwarna oranye sebagai hasil *forecasting*. Hasil *forecasting* menunjukkan bahwa pada Bulan April 2016, setiap harinya PT. TPS akan menerima jumlah kontainer yang lebih sedikit dibandingkan dengan hasil metode *moving average*. Rata-rata jumlah kontainer yang tiba setiap harinya pada Bulan April 2016 adalah 288 kontainer bahkan cenderung stagnan di angka 288 kontainer. Selanjutnya dihitung nilai MSE dan ER untuk metode *simple exponential smoothing* sehingga didapatkan nilai MSE adalah sebesar 259155 dan nilai ER sebesar 52,92851.



Gambar 4.2 *Forecasting* jumlah kontainer menggunakan *simple exponential smoothing*

Proses *forecasting* terakhir menggunakan metode *double exponential smoothing* untuk menghasilkan jumlah kontainer tiap harinya selama Bulan April 2016. Perhitungan *forecasting* metode ini menggunakan Persamaan 2 sebagaimana ditunjukkan Bab 3. Gambar 4.3 menunjukkan grafik hasil perhitungan *forecasting* menggunakan *double exponential smoothing*. Hasil *forecasting* metode ini menunjukkan perbedaan yang signifikan dibandingkan hasil *forecasting* menggunakan *moving average* dan *simple exponential*

smoothing. Jika metode sebelumnya menghasilkan jumlah kontainer yang cenderung stabil, maka metode ini menghasilkan jumlah kontainer yang cenderung mengalami peningkatan setiap harinya, bahkan melampaui jumlah tertinggi *data training*. Selanjutnya dihitung nilai MSE dan ER untuk metode *double exponential smoothing* sehingga didapatkan nilai MSE adalah sebesar 428117,2 dan nilai ER sebesar 53,01028.



Gambar 4.3 *Forecasting* jumlah kontainer menggunakan *double exponential smoothing*

Setelah dilakukan perhitungan *forecasting* menggunakan 3 metode berbeda, antara lain metode *moving average*, *simple exponential smoothing*, dan *double exponential smoothing*. Maka didapatkan nilai mean square error (MSE) dan relative error (RE) setiap metode. Oleh karena itu, dilakukan perbandingan untuk mendapatkan nilai MSE dan RE yang paling kecil diantara ketiga metode *forecasting*.

Table 4.2 Perbandingan MSE dan RE metode *forecasting*

Metode	MSE	RE
Moving Average	253946,7	51,16618
Simple Exponential Smoothing	259155	52,92851
Double Exponential Smoothing	428117,2	53,01028

Perbandingan MSE dan RE setiap metode *forecasting* ditunjukkan melalui Tabel 4.4. Didapatkan bahwa metode *moving average* memiliki nilai MSE dan RE paling rendah. MSE dan RE yang rendah menunjukkan bahwa hasil *forecasting* jumlah kontainer mendekati nilai sebenarnya sesuai dengan *data training*. Sehingga, hasil *forecasting* menggunakan metode *moving average* dijadikan acuan untuk membentuk *trace* dan membentuk *forecasted* log data.

4.1.4 Generate Log Data

Generate log data dilakukan untuk membentuk log data hasil *forecasting*, atau disebut sebagai *forecasted* log data untuk membedakan dengan *existing* log data. Sebagaimana yang telah dibahas pada subbab sebelumnya bahwa hasil *forecasting* yang dijadikan acuan untuk membentuk log data adalah metode *forecasting* yang memiliki nilai MSE dan RE terendah, dan telah didapatkan bahwa metode *moving average* memiliki nilai MSE dan RE terendah. Oleh karenanya, hasil *forecasting* metode *moving average* dijadikan acuan membentuk *trace* dan *forecasted* log data. *Forecasted* log data diperlukan untuk mengetahui performa *time* dan *cost* di waktu yang akan datang. Langkah-langkah untuk melakukan *generate* guna membentuk *forecasted* log data adalah

1. Menghitung *forecasted* jumlah kontainer setiap harinya selama satu bulan.
2. Menghitung frekuensi jumlah kontainer setiap *trace* pada setiap bulan.
3. Menghitung prosentase frekuensi tiap *trace*, *average* kontainer per hari per *trace*, dan standar deviasi kontainer per hari per *trace*.
4. Membangkitkan bilangan acak untuk jumlah kontainer tiap *trace* menggunakan distribusi normal dengan cara:
 $trace\ n = NORMINV(RAND());\ average\ container\ trace\ n ;\ st_dev\ container\ trace\ n$.
5. Jika hasil *forecasted* kontainer – jumlah kontainer hasil *generate* acak semua *trace* > 0 , maka sisa jumlah kontainer akan di-*generate* ulang menggunakan prosentase frekuensi tiap *trace*.

6. Jumlah kontainer untuk *trace* n = prosentase trace n * sisa jumlah kontainer pada proses ke-5.
7. Menjumlahkan hasil masing-masing *trace* pada langkah ke-4 dan ke-6. Hasil dari proses ini adalah *generated trace*.
8. Membentuk atribut *case_id*, aktivitas/message, *sojourn time*, *waiting time* dan *execution time* untuk membentuk *forecasted* log data mengacu pada *generated trace*.
9. Menambahkan log data untuk *case_id*, aktivitas/message, *sojourn time*, *waiting time* dan *execution time* mengacu pada hasil jumlah kontainer pada *generated trace*.

Berdasarkan langkah-langkah yang telah disebutkan, log data dibentuk sesuai dengan hasil *forecasting* yang memperoleh jumlah kontainer selama 30 hari di bulan April 2016. Langkah pertama telah terselesaikan dengan mendapatkan hasil *forecasting* menggunakan metode *moving average*. Tabel 4.5 menunjukkan jumlah kontainer hasil *forecasting* pada Bulan April 2016 berdasarkan metode *moving average*. Hasil *forecasting* menunjukkan jumlah kontainer yang tiba di Terminal Petikemas Surabaya setiap harinya berubah namun tidak pernah lebih dari 1000 kontainer per hari. Hasil pada Tabel 4.5 tersebut menjadi acuan untuk membentuk *forecasted trace* dan *forecasted* log data. Misalkan, jika dilakukan *generating forecasted trace* dan *forecasted* log data, pada Tanggal 1 April 2016, maka akan dibentuk sejumlah log data sebanyak 965 *case_id* dengan pembagian *case_id* tiap *trace* memiliki jumlah yang berbeda.

Table 4.5 Jumlah kontainer hasil *forecasting* Bulan April 2016

Tanggal	Jumlah Kontainer	Tanggal	Jumlah Kontainer
01/04/2016	965	16/04/2016	994
02/04/2016	963	17/04/2016	995
03/04/2016	960	18/04/2016	996
04/04/2016	953	19/04/2016	997
05/04/2016	952	20/04/2016	992

06/04/2016	952	21/04/2016	994
07/04/2016	955	22/04/2016	998
08/04/2016	959	23/04/2016	999
09/04/2016	960	24/04/2016	1000
10/04/2016	966	25/04/2016	987
11/04/2016	975	26/04/2016	986
12/04/2016	986	27/04/2016	982
13/04/2016	991	28/04/2016	982
14/04/2016	992	29/04/2016	975
15/04/2016	993	30/04/2016	969

Langkah kedua adalah menghitung frekuensi masing-masing *trace* pada Bulan Januari hingga Maret 2016. Pada log data, terdapat 12 *trace* yang menjadi jalur atau lintasan suatu proses impor pada PT. TPS. *Trace* tersebut didapatkan dengan mendeteksi jalur setiap *case_id* pada log data. Cara mendeteksi jumlah *trace* dapat dilakukan dengan bantuan perangkat bantu *Disco Fluxicon*. Tabel 4.6 menunjukkan 12 *trace* pada log data PT.TPS. Kemudian, dihitung frekuensi tiap *trace* pada Bulan Januari hingga Bulan Maret 2016. Proses menghitung frekuensi tiap *trace* juga menggunakan perangkat bantu *Disco Fluxicon*.

Table 4.6 Frekuensi masing-masing *trace* Bulan Januari – Maret 2016

Nama Trace	Keterangan	Januari	Februari	Maret	Total
Trace 1	Quarantine;Dry;Green Line	4286	3823	4844	12953
Trace 2	Quarantine;Dry;Red Line	201	162	215	578
Trace 3	Quarantine;Reefer;Green Line	584	1380	757	2721
Trace 4	Quarantine;Reefer;Red Line	24	53	19	96
Trace 5	Quarantine;Uncontainer;Green Line	14	0	0	14
Trace 6	Quarantine;Uncontainer;Red Line	0	0	0	0
Trace 7	Dry;Green Line	16334	16992	17102	50428
Trace 8	Dry;Red Line	773	587	444	1804
Trace 9	Reefer;Green Line	1604	379	480	2463
Trace 10	Reefer;Red Line	96	6	4	106
Trace 11	Uncontainer; Green Line	54	48	72	174
Trace 12	Uncontainer; Red Line	3	1	6	10
	TOTAL	23973	23431	23943	71347

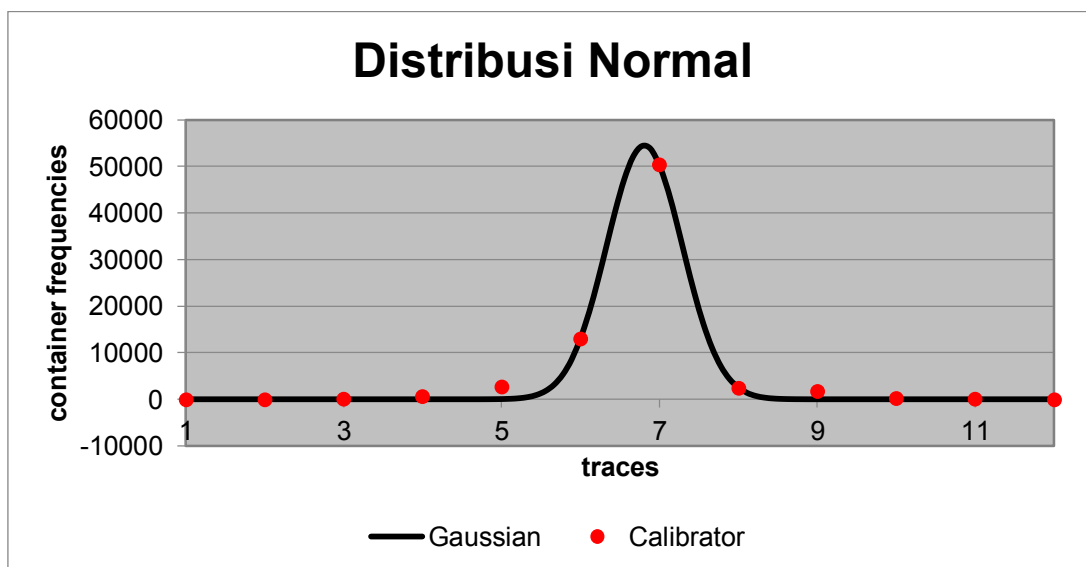
Tabel 4.4 juga menunjukkan bahwa terdapat *trace* yang memiliki frekuensi 0, yaitu pada *trace* 6, sehingga dapat dipastikan bahwa *trace* 6 juga tidak memiliki frekuensi jumlah kontainer ketika dilakukan *generate*. Selain itu, *trace* dengan frekuensi paling banyak terdapat pada *trace* 7, karena mayoritas kontainer yang masuk di PT. TPS memiliki jalur hijau pada saat pemeriksaan bea cukai dan jenisnya bermuatan bahan kering (*dry*). Oleh karena itu, ketika melakukan *generate trace*, tentu frekuensi yang terbanyak akan didapatkan oleh *trace* 7.

Kemudian, langkah selanjutnya sesuai dengan usulan metode adalah menghitung prosentase frekuensi tiap *trace*, *average* kontainer per hari per *trace*, dan standar deviasi kontainer per hari per *trace*. Gambar 4.4 menunjukkan hasil prosentase frekuensi tiap *trace*, *average* kontainer per hari per *trace*, dan standar deviasi kontainer per hari per *trace*. Prosentase didapatkan dengan cara membandingkan frekuensi total tiap *trace* dan frekuensi total semua *trace* kemudian dikalikan dengan 100%. Prosentase terbanyak didapatkan oleh *trace* 7 dan frekuensi terkecil oleh *trace* 6. Untuk *average* tiap harinya didapatkan dari *average* per bulan dibagi dengan 30 hari. Begitu pula untuk mencari standar deviasi per hari, didapatkan dengan menghitung standar deviasi per bulan dibagi dengan 30 hari.

Trace	Prosentase	Average Per Bulan	Average Per Hari	Stdev Per Bulan	St Dev Per Hari
1	18,15	4318	144	511	17
2	0,81	193	6	27	1
3	3,81	907	30	419	14
4	0,13	32	1	18	1
5	0,02	5	0	8	0
6	0,00	0	0	0	0
7	70,68	16809	560	415	14
8	2,53	601	20	165	5
9	3,45	821	27	680	23
10	0,15	35	1	53	2
11	0,24	58	2	12	0
12	0,01	3	0	3	0

Gambar 4.4 Prosentase, *average* dan standar deviasi setiap *trace*

Setelah menghitung prosentase frekuensi tiap *trace*, *average* kontainer per hari per *trace*, dan standar deviasi kontainer per hari per *trace*, langkah berikutnya adalah membangkitkan bilangan acak untuk jumlah kontainer tiap *trace* menggunakan distribusi normal. Penentuan pembangkitan bilangan acak menggunakan distribusi normal didasarkan pada *curve fitting* untuk distribusi normal (*gaussian*) memiliki nilai error terkecil dibandingkan dengan *curve fitting* menggunakan metode distribusi yang lainnya. Gambar 4.5 menunjukkan *curve fitting* menggunakan ditribusi normal (*gaussian*).



Gambar 4.5 Distribusi Normal Gaussian

Kemudian, Gambar 4.6 menunjukkan hasil pembentukan *generated trace* menggunakan *generate* bilangan acak berbasis distribusi normal dengan cara:

$$trace\ n = NORMINV(RAND(); avg\ container\ trace\ n ; st_dev\ container\ trace\ n).$$

Nilai *trace n* didapatkan dari random distribusi normal dengan parameter rata-rata kontainer per hari pada *trace n* dan standar deviasi per hari untuk *trace n*. Gambar 4.6 menunjukkan hasil dari pembentukan *trace*. Setelah itu, nilai semua *generated trace* dijumlahkan untuk mengetahui jumlah total *generated trace* per harinya yang terbentuk dari Tanggal 1 hingga 30 April. Jumlah dari total *generated trace* per hari dibandingkan dengan hasil *forecasting* jumlah kontainer. Hasilnya, seluruh nilai total *generated trace* < hasil *forecasting* jumlah kontainer. Oleh karena itu dilakukan langkah ke-5 dan ke-6 sesuai dengan usulan metode, yaitu

melakukan generated trace berdasarkan hasil pengurangan hasil *forecasting* jumlah kontainer - total *generated trace* untuk setiap harinya.

GENERATED TRACES PAKAI RANDOM DISTRIBUSI NORMAL															
Date	Predictive Container	Trace 1	Trace 2	Trace 3	Trace 4	Trace 5	Trace 6	Trace 7	Trace 8	Trace 9	Trace 10	Trace 11	Trace 12	GENERATE	SISA
01/04/2016	965	156	7	33	2	0	0	543	12	36	3	2	0	794	171
02/04/2016	963	114	6	10	1	0	0	553	20	-24	3	1	0	685	278
03/04/2016	960	109	6	60	2	0	0	562	22	39	1	1	0	802	158
04/04/2016	953	155	7	35	1	0	0	568	12	-5	2	2	0	777	176
05/04/2016	952	136	6	5	0	0	0	548	24	13	2	2	0	737	215
06/04/2016	952	136	7	22	2	1	0	557	35	40	3	2	0	804	148
07/04/2016	955	188	7	81	1	0	0	554	12	5	1	2	0	850	105
08/04/2016	959	146	6	12	1	0	0	575	18	53	0	2	0	812	146
09/04/2016	960	142	7	27	0	0	0	545	19	26	2	1	0	769	191
10/04/2016	966	105	7	37	2	0	0	571	21	1	0	2	0	746	220
11/04/2016	975	146	6	56	0	0	0	546	15	30	2	2	0	803	172
12/04/2016	986	132	6	16	1	0	0	571	19	37	-1	2	0	782	204
13/04/2016	991	172	5	39	2	0	0	538	17	69	1	2	0	844	147
14/04/2016	992	141	5	23	0	0	0	566	16	4	-1	1	0	756	236
15/04/2016	993	157	6	27	2	0	0	566	24	23	0	2	0	806	187
16/04/2016	994	150	7	19	0	0	0	563	12	14	-1	2	0	767	227
17/04/2016	995	107	7	53	0	0	0	559	17	34	3	2	0	782	213
18/04/2016	996	135	6	39	1	0	0	565	20	-12	2	2	0	758	237
19/04/2016	997	133	7	36	1	0	0	537	27	6	1	2	0	750	247
20/04/2016	992	133	6	24	1	0	0	552	22	0	3	3	0	744	247
21/04/2016	994	153	7	31	1	0	0	560	17	10	2	2	0	785	209
22/04/2016	998	131	9	16	0	1	0	547	22	16	0	2	0	744	254
23/04/2016	999	162	6	18	2	0	0	554	15	25	1	2	0	785	215

Gambar 4.6 *Generated trace* menggunakan random distribusi normal

Pada metode yang diusulkan, langkah ke-5 yang harus dilakukan jika hasil *forecasted* kontainer – jumlah total *generated trace* > 0 adalah sisa jumlah kontainer akan di-*generate* ulang menggunakan prosentase frekuensi tiap *trace*. Cara untuk mendapatkan *generate trace* terbaru yakni jumlah kontainer untuk $trace\ n = \text{prosentase trace } n * \text{sisa jumlah kontainer pada proses ke-4}$. Gambar 4.7 menunjukkan hasil yang diperoleh ketika meng-*generate* jumlah kontainer tiap *trace* menggunakan prosentase. Dengan menggunakan cara ini, sisa jumlah kontainer pada langkah ke-5 dapat terbagi rata ke semua *trace*. *Trace* yang memiliki prosentase terbanyak seperti *trace 7* tentu akan memiliki hasil *generate* yang banyak pula.

GENERATED TRACES USING PROSENTASE													
Date	Sisa Generate	Trace 1	Trace 2	Trace 3	Trace 4	Trace 5	Trace 6	Trace 7	Trace 8	Trace 9	Trace 10	Trace 11	Trace 12
01/04/2016	171	31	1	7	0	0	0	121	4	6	0	0	0
02/04/2016	278	51	2	11	0	0	0	197	7	10	0	1	0
03/04/2016	158	29	1	6	0	0	0	112	4	5	0	0	0
04/04/2016	176	32	1	7	0	0	0	124	4	6	0	0	0
05/04/2016	215	39	2	8	0	0	0	152	5	7	0	1	0
06/04/2016	148	27	1	6	0	0	0	104	4	5	0	0	0
07/04/2016	105	19	1	4	0	0	0	74	3	4	0	0	0
08/04/2016	146	27	1	6	0	0	0	103	4	5	0	0	0
09/04/2016	191	35	2	7	0	0	0	135	5	7	0	0	0
10/04/2016	220	40	2	8	0	0	0	156	6	8	0	1	0
11/04/2016	172	31	1	7	0	0	0	122	4	6	0	0	0
12/04/2016	204	37	2	8	0	0	0	144	5	7	0	0	0
13/04/2016	147	27	1	6	0	0	0	104	4	5	0	0	0
14/04/2016	236	43	2	9	0	0	0	167	6	8	0	1	0
15/04/2016	187	34	2	7	0	0	0	132	5	6	0	0	0
16/04/2016	227	41	2	9	0	0	0	161	6	8	0	1	0
17/04/2016	213	39	2	8	0	0	0	150	5	7	0	1	0
18/04/2016	237	43	2	9	0	0	0	168	6	8	0	1	0
19/04/2016	247	45	2	9	0	0	0	175	6	9	0	1	0

Gambar 4.7 *Generated trace* menggunakan prosentase

Langkah selanjutnya adalah menjumlahkan *generated trace* menggunakan random distribusi normal dengan *generated trace* menggunakan prosentase untuk mendapatkan total *generated trace*, yang pada proses selanjutnya digunakan sebagai acuan untuk meng-generate log data.

TOTAL GENERATED FORECASTED EVENT LOG													
Date	Total Kontainer	Trace 1	Trace 2	Trace 3	Trace 4	Trace 5	Trace 6	Trace 7	Trace 8	Trace 9	Trace 10	Trace 11	Trace 12
01/04/2016	965	164	9	34	0	1	0	712	31	11	1	3	0
02/04/2016	963	166	7	47	2	0	0	649	35	51	4	2	0
03/04/2016	960	177	7	50	1	1	0	677	28	14	3	2	0
04/04/2016	953	193	7	50	1	0	0	660	25	17	-1	3	0
05/04/2016	952	156	7	38	2	1	0	682	23	41	0	2	0
06/04/2016	952	164	7	28	1	0	0	664	33	53	1	2	0
07/04/2016	955	198	7	50	1	0	0	644	22	29	2	2	0
08/04/2016	959	202	8	54	0	0	0	670	23	-1	0	2	0
09/04/2016	960	164	8	41	1	1	0	647	13	81	2	2	0
10/04/2016	966	149	8	38	2	0	0	696	21	51	0	2	0
11/04/2016	975	162	7	47	1	0	0	711	22	20	4	1	0
12/04/2016	986	165	7	55	1	0	0	673	16	64	3	2	0
13/04/2016	991	192	8	46	2	0	0	687	22	30	2	2	0
14/04/2016	992	156	9	58	2	0	0	718	19	24	3	2	0
15/04/2016	993	193	8	27	1	0	0	700	23	36	1	3	0
16/04/2016	994	160	8	75	1	0	0	673	27	47	1	3	0
17/04/2016	995	160	8	38	1	0	0	664	30	87	3	3	0
18/04/2016	996	170	8	58	2	0	0	689	30	34	1	2	0
19/04/2016	997	189	7	42	2	0	0	729	31	-9	3	2	0
20/04/2016	992	164	8	55	2	0	0	662	24	69	5	3	0
21/04/2016	994	174	9	29	2	0	0	736	28	9	5	2	0
22/04/2016	998	177	8	73	1	0	0	678	25	33	1	2	0

Gambar 4.8 total *Generated trace*

Gambar 4.8 menunjukkan total *generated trace*. Keseluruhan *trace* tiap harinya jika dijumlahkan akan sama dengan hasil *forecasting* jumlah kontainer per harinya. Total *generated trace* inilah yang menjadi acuan untuk membentuk log data pada Bulan April 2016. Jika terdapat *trace* yang memiliki nilai negatif maka

dilakukan proses absolut sehingga tidak akan ada nilai negatif untuk setiap trace, namun dengan konsekuensi bahwa nilai total *generated trace* akan berbeda dengan hasil *forecasting* jumlah kontainer. Oleh karena itu, peneliti menyarankan agar dalam meng-*generate* bilangan random pada proses *generated trace* dilakukan pembatasan bahwa angka hasil generate tidak boleh < 0 .

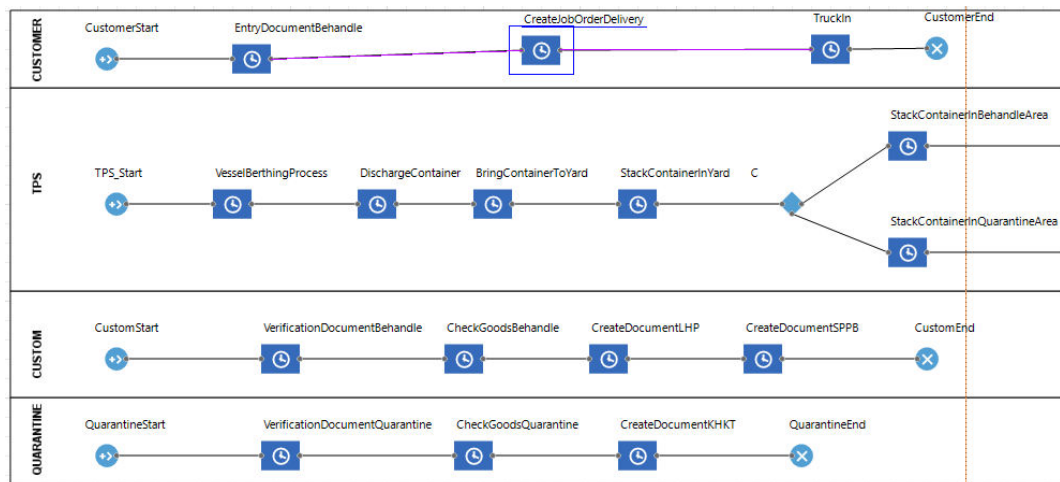
Langkah terakhir sesuai dengan usulan metode adalah melakukan *generating* log data untuk mendapatkan log data terbaru berdasarkan acuan *generated trace*. Berbeda dengan log data transformasi, *generated* log data dibuat berdasarkan atribut *sojourn time*. Oleh karena itu, setiap *trace* di-*generate* sesuai dengan aktivitas/*message* yang terdapat pada *trace* tersebut. Kemudian setiap aktivitas/*message* pada semua trace di-*generate* *sojourn time*-nya, dengan cara membangkitkan bilangan acak menggunakan distribusi normal dengan parameter *average sojourn time* dan standar deviasi *sojourn time* mengacu pada log data Bulan Januari hingga Maret 2016. Hasil yang didapatkan adalah nilai *sojourn time* pada setiap aktivitas/*message* untuk *generated* log data pada Bulan April 2016. Melalui *sojourn time* pada *generated* log data, dapat dihitung pula *execution time* dan *waiting time*, dengan cara yang sama seperti pada *Pre-processing*. Sehingga *generated* log data dapat digunakan untuk menghitung *dwelling time* pada Bulan April 2016. Hasil *generated* log data ditunjukkan pada Gambar 4.9. Setiap aktivitas/*message* sudah memiliki atribut *time* secara lengkap untuk evaluasi kinerja, yakni *sojourn time*, *waiting time* dan *execution time*.

Activity	SojournTime	AverageST	ST_DEV	ExTime	WaitTime	MIN Sojourn
Document_Entry_via_PDE	6648	7162,957258	2052,277103	5898	750	3603
Request_Behandle	60	89,61290323	29,9975025	38	22	60
Vessel_Berthing_Process	65335	64990,4125	12522,73563	51263	14072	43233
Discharge_Container	387	598,5092742	172,7009066	315	72	300
Bring_Container_to_Yard	1140	1104,708871	348,860568	982	158	600
Stack_Container_in_Yard	746	599,5157258	170,8643815	440	306	300
Approve_Behandle	99840	118141,8387	115346,4038	85408	14432	10980
Verification_Document_Behandle	496	361,9318548	138,0746022	441	55	120
Create_document_SPPB	24780	29968,1371	46418,00623	16328	8452	60
Send_SPPB_Info	68	91,21653226	38,23638658	52	16	60
Create_Job_Order_Document_Delivery	27900	34014,31452	50643,67174	17586	10314	60
Send_Job_Order_Delivery_Info	120	150,353629	29,94478028	119	1	120
Truck_in	172740	123267,5565	332560,8434	119249	53491	540
Dispatch_WQ_Delivery_to_CHE	60	96,96774194	863,7544635	42	18	60
Determine_Container_Type	180	304,016129	2441,721101	120	60	60
Determining_Dry	120	256,1361457	2306,667278	92	28	60
Decide_Task_Before_Lift_Container	120	139,6209677	1498,193003	88	32	60
Lift_on_Container_Truck	120	318,0241935	2733,772438	75	45	60
Truck_Go_To_Gate_Out	2760	4383,798387	9332,013764	2541	219	60
Check_Container_before_Truck_out	60	87,19354839	31,79868059	22	38	60
Truck_Out	120	140,9274194	1500,247381	90	30	60
Document_Entry_via_PDE	5487	7162,957258	2052,277103	4836	651	3603
Request_Behandle	120	89,61290323	29,9975025	86	34	60
Vessel_Berthing_Process	58570	64990,4125	12522,73563	46465	12105	43233

Gambar 4.9 *generated log data* berdasarkan *generated trace*

4.1.5 Simulasi Berbasis Agen

Langkah-langkah pada fase *pre-processing* membentuk suatu transformasi mengubah database menjadi log data. Log data yang dihasilkan adalah dalam rentang Bulan Januari hingga Maret 2016. Simulasi berbasis agen dilakukan untuk mensimulasikan *existing* log data yaitu Bulan Januari hingga Maret 2016 serta untuk mensimulasikan *forecasted* log data yakni pada bulan April 2016. Untuk mengimplementasikan simulasi berbasis agen, digunakan perkakas bantu *Anylogic Personal Learning Edition 8.0*. Awalnya, dibuat model simulasi berbasis agen mengacu pada proses bisnis multi organisasi yang terdapat pada Terminal Petikemas Surabaya. Gambaran proses bisnis secara menyeluruh dilampirkan pada Lampiran C. Lampiran C juga menjadi acuan untuk memodelkan proses bisnis multi organisasi untuk simulasi berbasis agen di *AnyLogic*.

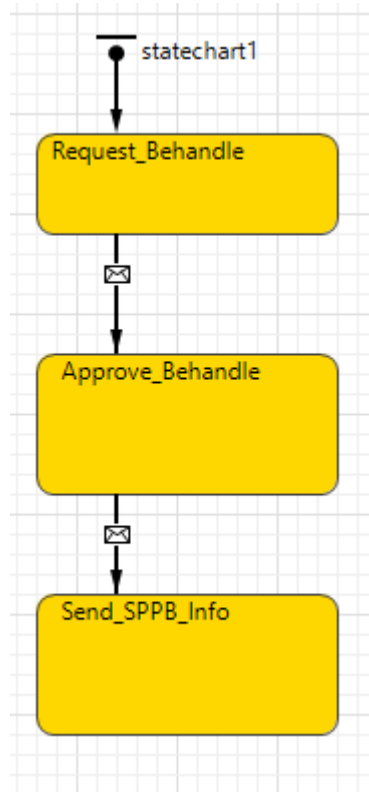


Gambar 4.10 Implementasi Activity pada *Agent Based Simulation*

Gambar 4.10 adalah implementasi pemodelan multi organisasi di Anylogic. Terdapat 4 agen yang digambarkan, yaitu *customer*, *TPS*, *custom* dan *quarantine*. Masing-masing agen memiliki model untuk *start* dan *end*. Inilah yang membedakan simulasi berbasis agen dengan simulasi discrete-event. Pada discrete-event, hanya terdapat satu *start* dan satu *end*, sehingga multi organisasi hanya digambarkan sebagai suatu kesatuan model proses yang sinkron. Namun, pada simulasi berbasis agen, model proses bersifat asinkron karena masing-masing agen tidak memiliki jadwal yang tetap untuk memulai proses. Bahkan, membutuhkan perantara *message* untuk menjadi pemicu (*trigger*) agar aktivitas pada agen yang lain dapat dijalankan. Proses asinkron inilah yang terjadi di Terminal Petikemas Surabaya, sehingga simulasi dimaksudkan untuk membuktikan bahwa komunikasi asinkron antar agen dapat mempengaruhi *dwelling time* menjadi lebih besar dibandingkan dengan proses bisnis yang sinkron menggunakan *discrete-event simulation*.

Gambar 4.11 menunjukkan implementasi message pada simulasi berbasis agen. *Message* pada Anylogic dimodelkan melalui *statechart* kemudian dihubungkan dengan model simulasi berbasis agen pada Gambar 4.10. Adapun message yang dimodelkan antara lain *Approve_Behandle*, *Approve_Quarantine*, *Request_Behandle*, *Request_Quarantine*, *Send_Job_Order_Behandle_Info*, *Send_Certificate_KH/KT_Info*, *Send_Job_Order_Delivery_Info*, *Send_SPPB_Info*, *Send_Job_Order_Quarantine_Info*. Untuk menghubungkan

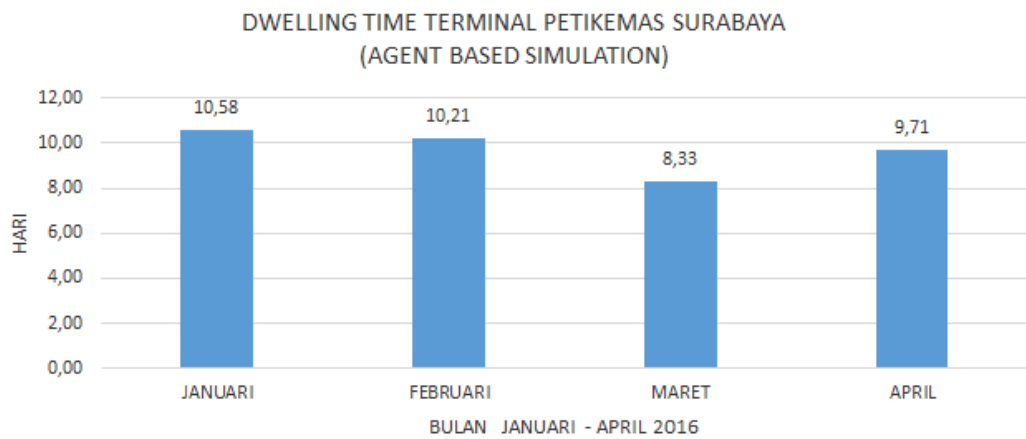
message yang dimodelkan melalui *statechart* dengan model proses bisnis untuk simulasi berbasis agen, maka disesuaikan dengan aktivitas pada setiap agen.



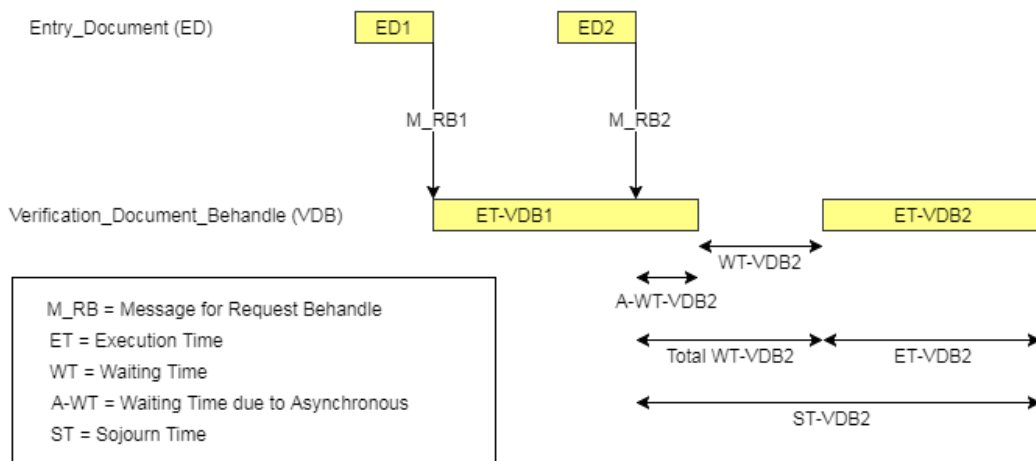
Gambar 4.11 Implementasi *message* pada Agent Based Simulation

Melalui hasil simulasi berbasis agen, diharapkan performa *time* dan *cost* pada tiap aktivitas akan diketahui. Terutama untuk mengetahui nilai *dwelling time* pada *existing* log data pada bulan Januari hingga Maret 2016 dan *forecasted* log data pada Bulan April 2016. *Dwelling time* mencakup proses yang dilakukan oleh 4 agent (*customer*, *custom*, *quarantine* dan TPS) sehingga nilai *time* per aktivitas per agen dapata dihitung. Oleh karena itu dapat diketahui agen yang memiliki performa *time* paling lama dibanding dengan agen yang lain. Acuan perhitungan *dwelling time* adalah berdasarkan waktu ketika kapal *discharge* (*start dwelling time*) hingga diakhiri ketika *truck* keluar dari area PT. TPS (*end dwelling time*). *Dwelling time* mengacu pada *sojourn time* serta menggunakan *agent based simulation*, didapatkan nilai *dwelling time* sebagaimana Gambar 4.7, yakni nilai *dwelling time* pada rentang bulan Januari hingga April 2016. Didapatkan bahwa *dwelling time* PT. TPS memiliki waktu yang tinggi, bahkan melampaui target

yang ditetapkan pemerintah, yakni maksimal 3 hari. Berdasarkan Gambar 4.7, *dwelling time* tertinggi terdapat pada Bulan Januari 2016 sebesar 10.58 hari, sedangkan *dwelling time* terendah terdapat pada Bulan Maret 2016 selama 8,33 hari. Faktor yang menyebabkan lamanya *dwelling time* adalah adalah komunikasi asinkron yang mempengaruhi waktu tunggu beberapa aktivitas. Gambaran komunikasi asinkron ditunjukkan oleh Gambar 4.8.



Gambar 4.12 Laporan *dwelling time* bulan Januari-April melalui simulasi berbasis agen



Gambar 4.13 Gantt Chart untuk menunjukkan komunikasi asinkron

Gambar 4.13 menunjukkan *ganttt chart* yang merepresentasikan komunikasi asinkron. Komunikasi asinkron ditunjukkan bahwa ketika ED1 selesai melakukan aktivitas maka ia mengeluarkan *message request behandle* sebagai *trigger* aktivitas VDB. Setelah itu, VDB1 mulai menjalankan eksekusi (ET-VDB1).

Namun, ditengah perjalanannya, VDB1 mendapatkan message *request handle* yang berasal dari ED2, sehingga VDB2 tidak langsung menjalankan eksekusi, melainkan menunggu VDB1 menyelesaikan eksekusi hingga tuntas. Oleh karena itu, penelitian ini menemukan adalah *waiting time* yang berasal dari komunikasi asinkron. Pada Gambar 4.13 *waiting time* karena pengaruh asinkron disebut A-WT. Total *waiting time* yang dimiliki oleh VDB2 adalah penjumlahan dari *waiting time* normal (WT) dan *waiting time* karena asinkron (A-WT). Pada kenyataannya, komunikasi asinkron ini dialami oleh aktivitas yang berhubungan langsung dengan *message approve handle* dan *approve quarantine* yang berdampak pada tingginya *dwelling time* pada Bulan Januari hingga April 2016 di PT. TPS sebagaimana ditunjukkan oleh Gambar 4.13.

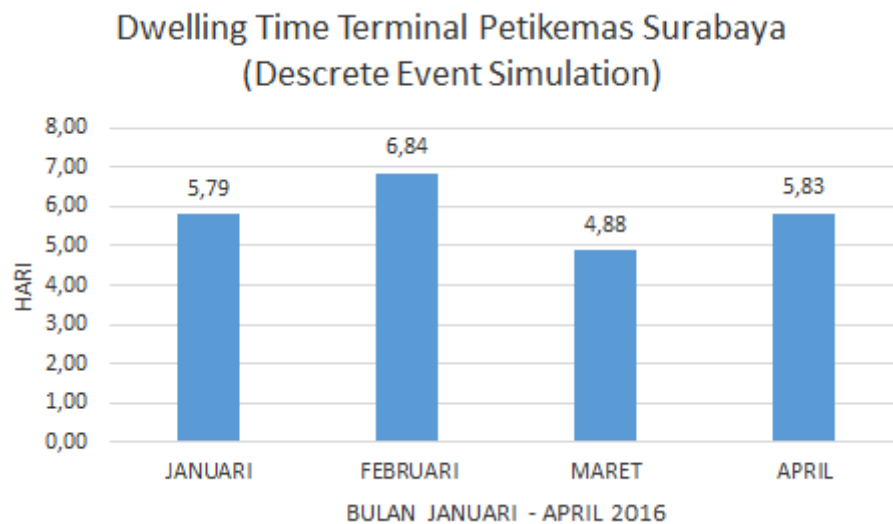
4.1.6 Simulasi Discrete-Event

Simulasi *discrete-event* dilakukan untuk mensimulasikan *existing* log data yaitu Bulan Januari hingga Maret 2016 serta untuk mensimulasikan *forecasted* log data yakni pada bulan April 2016. Untuk mengimplementasikan simulasi *discrete-event*, juga digunakan perangkat bantu *Anylogic Personal Learning Edition 8.0*. Awalnya, dibuat model simulasi *discrete-event* mengacu pada proses bisnis multi organisasi yang terdapat pada Terminal Petikemas Surabaya. Gambaran proses bisnis secara menyeluruh dilampirkan pada Lampiran C. Lampiran C juga menjadi acuan untuk memodelkan proses bisnis multi organisasi untuk simulasi berbasis agen di *AnyLogic*. Namun berbeda dengan simulasi berbasis agen, simulasi *discrete-event* ini memodelkan proses bisnis impor barang di PT. TPS secara menyeluruh dengan menggabungkan beberapa agen menjadi satu kesatuan proses bisnis, sehingga simulasi *discrete-event* hanya memiliki satu *start* dan satu *end*. Gambar 4.9 menunjukkan model proses bisnis untuk simulasi *discrete-event*. Simulasi *discrete-event* ini berguna untuk mensimulasikan proses bisnis yang tidak memiliki komunikasi asinkron. Dengan kata lain, simulasi *discrete-event* adalah simulasi untuk proses bisnis yang sinkron. Hasil dari simulasi *discrete-event* ini berguna untuk menunjukkan suatu perbandingan selisih yang didapatkan terhadap simulasi berbasis agen. Simulasi *discrete-event* mewakili proses bisnis

dengan karakteristik sinkron karena tidak melibatkan *message*, sedangkan simulasi berbasis agen melibatkan *message* yang bersifat asinkron.



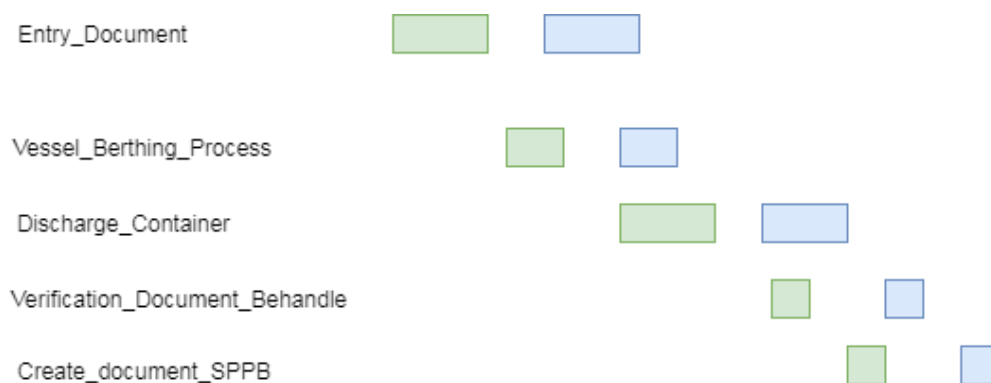
Gambar 4.14 Implementasi Descrete Event Simulation Anylogic



Gambar 4.15 Laporan *dwelling time discrete-event simulation*

Sementara itu, hasil *dwelling time* menggunakan *discrete-event simulation* memiliki hasil lebih rendah karena tidak melibatkan *message* pada saat simulasi, sehingga tiap aktivitas yang dijalani oleh masing-masing agen langsung berjalan tanpa ada gangguan *message*. Gambar 4.15 menunjukkan grafik *dwelling time* pada PT. TPS dalam rentang Bulan Januari hingga April 2016. *Dwelling time* tertinggi terdapat pada Bulan Februari sedangkan *dwelling time* terendah terdapat

Pada Bulan Maret 2016. Hasil ini memberikan bukti bahwa *message* asinkron pada simulasi berbasis agen memberikan pengaruh signifikan pada penambahan waktu *dwelling time*. Pada Gambar 4.16 gantt chart menunjukkan bahwa proses bisnis yang bersifat sinkron langsung mengeksekusi proses tanpa harus menunggu *message* sebagai pemicu aktivitas, sehingga seluruh aktivitas pada proses bisnis sinkron dapat dieksekusi secara berurutan.



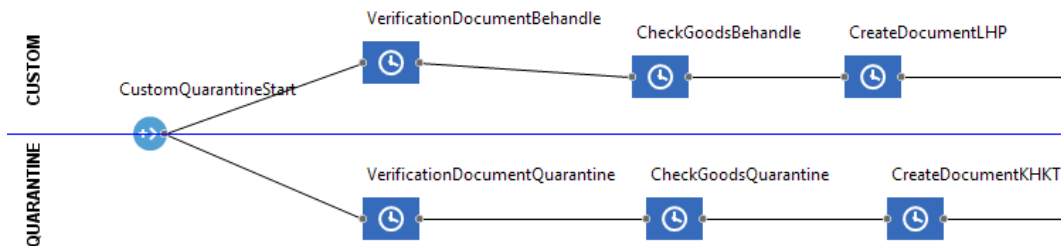
Gambar 4.16 Gantt Chart pada simulasi *discrete-event*

4.1.7 Paralelisasi

Paralelisasi dimaksudkan sebagai upaya untuk mengurangi *dwelling time* pada PT. TPS. Paralelisasi ini dimaksudkan sebagai bahan masukan untuk PT. TPS guna mengurangi *dwelling time*. Proses paralelisasi didasarkan pada *message* yang asinkron yaitu *Approve Behandle* dan *Approve Quarantine* dijadikan sebagai satu *message* yaitu *ApproveBehandleQuarantine*, sehingga agen bea cukai dan agen karantina dalam menjalankan prosesnya dimulai secara bersamaan karena cukup didahului oleh satu *message*. Pada kenyataan di lapangan, bea cukai dan karantina memiliki waktu mulai yang berbeda, sehingga *dwelling time* menjadi lama karena tidak terjadwal. Adapun kriteria paralelisasi yang diusulkan adalah:

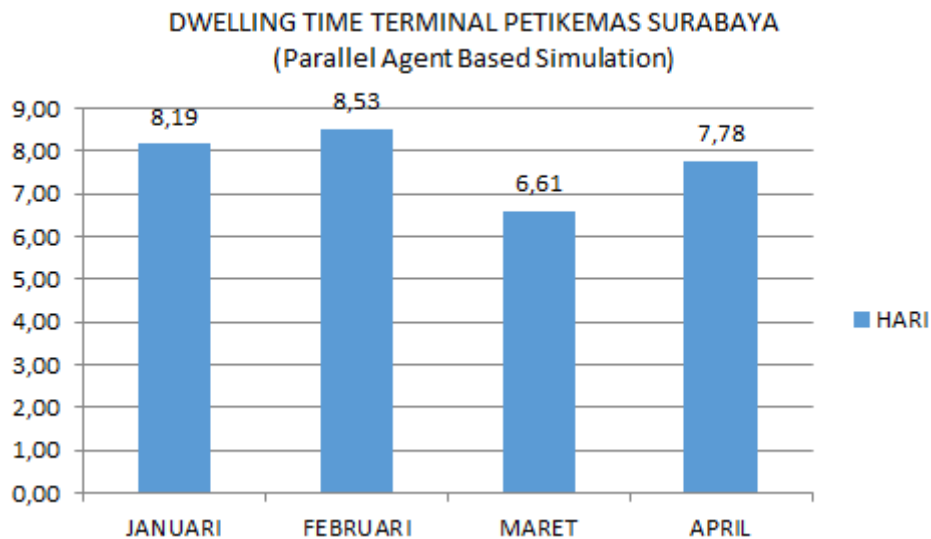
- Aktivitas yang independen harus diidentifikasi terlebih dahulu
- Aktivitas dapat di paralelisasi jika aktivitas pada entitas simulasi yang sama dijalankan dalam urutan *timestamp*
- Jika dua aktivitas pada entitas simulasi yang sama mengakses item data yang sama dengan cara yang berbeda

- Waktu dan tempat eksekusi berbeda
- Resource yang menangani setiap aktivitas berbeda
- *Message* bersifat asinkron



Gambar 4.17 Paralelisasi Bea Cukai dan Karantina

Pada Gambar 4.17, karantina dan bea cukai memiliki waktu mulai yang sama, sehingga perbedaan aktivitas yang bergantung pada message pada karantina dan bea cukai dapat berjalan bersamaan. Jika dilakukan simulasi menggunakan *agent based simulation*, didapatkan hasil sebagaimana Gambar 4.18.



Gambar 4.18 Dwelling time parallel ABS

Table 4.7 Perbandingan Dwelling Time Parallel dan Non-Parallel ABS

	Januari	Februari	Maret	April
Non Parallel	10,58	10,21	8,33	9,71

Parallel	8,19	8,53	6,61	7,78
Selisih	2,39	1,68	1,72	1,93

Tabel 4.7 menunjukkan selisih yang didapatkan ketika menggunakan parallel bea cukai dan karantina. Perhitungan *dwelling time parallel* dan *non-parallel* menggunakan simulasi berbasis agen. Selisih paling banyak didapatkan pada Bulan Januari, yaitu *dwelling time* berkurang 2,39 hari. Rata-rata selisih yang didapatkan adalah 1,93 hari jika menggunakan model proses bisnis parallel. Hasil ini diharapkan dapat memberikan rekomendasi perbaikan pada PT. TPS agar mengurangi waktu *dwelling time*. 4. Paralelisasi terbukti dapat mengurangi waktu tunggu dan memberikan dampak untuk memperkecil *dwelling time*.

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

1. *Forecasting* jumlah kontainer dilakukan menggunakan 3 metode yang berbeda, yaitu *moving average*, *simple exponential smoothing* serta *double exponential smoothing*. Perbandingan *mean square error* (MSE) dan *relative error* (RE) pada ketiga metode menunjukkan bahwa metode *moving average* memiliki MSE dan RE paling kecil, sehingga hasil *forecasting* dijadikan acuan untuk *generating* log data.
2. Penelitian ini menghasilkan metode untuk melakukan pembentukan log data melalui hasil *forecasting* jumlah container, dengan cara melakukan *generate* bilangan random distribusi normal pada setiap *trace* sehingga menghasilkan *generated* log data. *Generated* log data merepresentasikan log data di waktu yang akan datang.
3. Penentuan *dwelling time* dihitung menggunakan simulasi berbasis agen dan simulasi *discrete-event* pada *existing* log data dan *forecasted* log data. Didapatkan hasil bahwa *dwelling time* pada *existing* log data dan *forecasted* log melebihi batas yang ditetapkan pemerintah, yakni maksimal 3 hari. Sedangkan pada log data didapatkan rata-rata *dwelling time* adalah 9,7 hari berdasarkan simulasi berbasis agen dan *dwelling time* 5,4 hari menggunakan simulasi *discrete-event*. Mengacu pada perbandingan kedua simulasi didapatkan bahwa perbedaan waktu *dwelling time* pada simulasi berbasis agen dan simulasi *discrete-event* dipengaruhi oleh adanya *message* yang bersifat asinkron pada simulasi berbasis agen, sehingga menghasilkan *dwelling time* yang lebih lama. Rekomendasi perbaikan dilakukan dengan melakukan paralelisasi agen bea cukai dan karantina. Paralelisasi ditentukan berdasarkan adanya agen yang dikenai datangnya *message* asinkron sehingga paralelisasi diharapkan dapat mengurangi waktu tunggu dan memberikan dampak untuk memperkecil *dwelling time*. Hasil paralelisasi menunjukkan bahwa *dwelling time* pada Log data Bulan Januari – April

berkurang dengan selisih rata-rata 1,93 hari dibandingkan dengan proses bisnis non-parallel yang digunakan sebelumnya.

5.2 Saran

- 1 Peneliti menyarankan agar penelitian yang akan datang menggunakan acuan database yang lebih detail, sehingga proses intervensi *expert* pada saat transformasi tidak terlalu banyak. Karena pada dasarnya semakin minim intervensi, maka data penelitian semakin akurat. Kemudian agar proses simulasi itu mencerminkan keadaan PT. TPS yang terus berkembang dengan berbagai kebijakan terbaru. Serta mampu membandingkan simulasi berbasis agen dengan beberapa *framework agent based* yang berbeda, sehingga hasil simulasi dapat ditelaah lebih dalam.

DAFTAR PUSTAKA

- A. Rozinat, R.S. Mans, M. Song, and W. M. P. van der A. (2010). Discovering Simulation Models - Tutorial. *Information Systems*, 34(3), 1–3. <https://doi.org/http://dx.doi.org/10.1016/j.is.2008.09.002>.
- Aalst, W. Van Der. (2016). *Process Mining: Data Science in Action*, Springer Verlag, ISBN 978-3-662-49850-7.
- Aalst, W. Van der, M. Dumas, and A.H.M. ter Hofstede. (2003). Web service composition languages: old wine in new bottles? In *Euromicro Conference*. 298–305.
- Ahmed, T., & Srivastava, A. (2014). 2014 IEEE International Conference on Services Computing Service Choreography: Present and Future, 3–4. <https://doi.org/10.1109/TSC.2013.27>
- Buliali, J. L. (2013). *Dasar Pemodelan dan Simulasi Sistem*. Surabaya: ITS PRESS.
- El Kholy, W., Bentahar, J., El Menshawy, M., Qu, H., & Dssouli, R. (2014). Modeling and verifying choreographed multi-agent-based web service compositions regulated by commitment protocols. *Expert Systems with Applications*, 41(16), 7478–7494. <https://doi.org/10.1016/j.eswa.2014.05.046>
- Ferreira, D. R. (2013). Mining the low-level behaviour of agents in high-level business processes, 6(2), 146–166.
- Ferreira, D. R., Szimanski, F., & Ralha, C. G. (2013). A hierarchical Markov model to understand the behaviour of agents in business processes. *Lecture Notes in Business Information Processing*, 132 LNBIP, 150–161. <https://doi.org/10.1007/978-3-642-36285-9-16>
- Grigoryev, I. (2016). *AnyLogic 7 in Three Days*. <https://doi.org/10.1007/s007690000247>
- Jianbo Bai, Hong Xiao, Xianghua Yang, and Guofang Zhang. (2009). Study on integration technologies of building automation systems based on web services. In *CCCM 2009*, Vol. 4. 262–266.
- Jung, J., Kang, S., & Kim, H. (2004). Choreography for B2B Collaboration, (February), 37–45.
- Khodyrev, I., & Popova, S. (2014). Discrete modeling and simulation of business processes using event logs. *Procedia Computer Science*, 29, 322–331. <https://doi.org/10.1016/j.procs.2014.05.029>
- Milanovi, M. (2010). Modeling Service Choreographies with Rule-enhanced Business Processes. <https://doi.org/10.1109/EDOC.2010.18>
- Onggo, B. S. S. (2012). Bpmn Pattern for Agent-Based Simulation Model Representation, 3657–3666.

- Park, S., Sutrisnowati, R. A., & Bae, H. (2016). Port Logistics Simulation Using CPN Tools with Yard Truck and Gantry Crane Configuration, 2–9.
- Peltz, Chris. (2003). Web services orchestration and choreography. *Computer* 36, 10, 46–52
- Rozinat, A., Mans, R. S., Song, M., & van der Aalst, W. M. P. (2008). Discovering colored Petri nets from event logs. *International Journal on Software Tools for Technology Transfer*, 10(1). <https://doi.org/10.1007/s10009-007-0051-0>
- Sarno, R., Haryadita, F., Kartini, Sarwosri, & Solichaha, A. (2016). Business process optimization from single timestamp event log. *Proceeding - 2015 International Conference on Computer, Control, Informatics and Its Applications: Emerging Trends in the Era of Internet of Things, IC3INA 2015*, 50–55. <https://doi.org/10.1109/IC3INA.2015.7377745>
- Solaiman, E., & Molina-jimenez, C. (2015). 2015 IEEE International Conference on Services Computing A Tool for the Automatic Verification of BPMN Choreographies. <https://doi.org/10.1109/SCC.2015.103>
- Szimanski, F., Ralha, G., Wagner, G., & Ferreira, D. R. (2013). Improving Business Process Models with Agent-Based Simulation and Process Mining. *Lecture Notes in Business Information Processing*, 147 LNBIP, 124–138. https://doi.org/10.1007/978-3-642-38484-4_10

LAMPIRAN

LAMPIRAN A

Intervensi Perhitungan Time

Perhitungan Time			
Activity	Database	Intervensi Complete	Keterangan
Entry Document via PDE		2-3 jam sebelum Vessel Berthing Process	
RequestBehandle		1-2 menit setelah Entry Document via PDE	
RequestQuarantine		1-2 menit setelah RequestBehandle	Aktivitas tersebut ada jika HAS_QUARANTINE_FLAG == YES
Vessel Berthing Process	VESSEL_ATB	12-24 jam setelah VESSEL_ATB	
Discharge Container	DISC_DATE	DISC_DATE	
Bring Container to Yard		2-3 menit sebelum Stack Container in Yard	
Stack Container in Yard	STACK_DATE	STACK_DATE	
Approve Behandle	CUSTOM_DEL_DATE	CUSTOM_DEL_DATE	jam 8.00-16.30 dan sebelum JOB_DEL_DATE
Verification Document Behandle	SP_JALUR_MERAH_PIB_DATE	5-10 Menit Setelah Approve Behandle	Jika jalur hijau -> 1-2 menit setelah Approve Behandle. Jika jalur merah, sesuai dengan SP_JALUR_MERAH_PIB_DATE
Create Job Order Document Behandle	FIRST_JOB_CUSTOMS_BEHANDLE_TS	FIRST_JOB_CUSTOMS_BEHANDLE_TS	Aktivitas-aktivitas tersebut ada kalau CUSTOMS_BEHANDLE_COUNT == 1
SendJobOrderBehandle Info		1-2 Menit Setelah Start SendJobOrderBehandle Info	
Stack Container in Behandle Area	FIRST_STACK_BEHANDLE_TS	FIRST_STACK_BEHANDLE_TS	

Check Goods Behandle		1 jam sebelum Create document LHP	
Create document LHP	LAST_COMPLETE_BEHANDLE_TS	LAST_COMPLETE_BEHANDLE_TS	
Stack Container in Yard From Behandle	STACK_MAIN_YARD_TS	STACK_MAIN_YARD_TS	Aktivitas-aktivitas tersebut ada kalau CUSTOMS_BEHANDLE_COUNT == 1
Create document SPPB		24-48 jam setelah start time create document sppb (rentang waktu 08.00-16.30)	Jika jalur hijau -> 1-24 jam setelah Verification Document Behandle. Jika jalur merah, 1-24 jam setelah Create document LHP
SendSPPBInfo		1- 2 menit setelah start time sendSPPBInfo	
Approve Quarantine		1-2 jam sebelum Create Job Order Document Quarantine	Aktivitas-aktivitas tersebut ada kalau HAS_QUARANTINE_FLAG == YES
Verification Document Quarantine		45menit - 1 jam setelah Approve Quarantine	
Create Job Order Document Quarantine	BEHANDLE_QUARANTINE_JOB_TS	15-30 menit setelah start time Create Job Order Document Quarantine	Aktivitas-aktivitas tersebut ada kalau dokumen lengkap
SendJobOrderQuarantineInfo		1-2 menit setelah start time SendJobOrderQuarantineInfo	
Stack Container in Quarantine Area	FIRST_STACK_QUARANTINE_BLK_TS	Jika tidak ada data, maka 1-3 hari sebelumSendCertificateKH/KTInfo	
Check Goods Quarantine		3 jam sebelum Create document KH/KT	
Create document KH/KT		1-2 jam sebelum SendCertificateKH/KTInfo	
SendCertificateKH/KTInfo	QUARANTINE_	Jika tidak ada data,	

fo	RELEASE	maka 1-3 hari setelah Stack Container in Quarantine Area	
Stack Container in Yard From Quarantine		1 jam setelah SendCertificateKH/KTInfo	
Create Job Order Document Delivery	JOB_DEL_DATE	JOB_DEL_DATE	
SendJobOrderDeliveryInfo		2-3 menit setelah Create Job Order Document	
Truck in	TRUCK_IN_DATE	TRUCK_IN_DATE	
Dispatch WQ Delivery to CHE		30 detik - 50 detik setelah start time Dispatch WQ Delivery to CHE	
Determine Container Type		1-2 menit setelah start time Determine Container Type	
Determining Reefer/ Determining Dry/ Determining Uncontainer		1-2 menit setelah start time Determining Reefer/ Determining Dry/ Determining Uncontainer	kalau Kolom Database CTR_TYPE == DRY, maka Determining Dry, kalau CTR_TYPE == RFR maka Determining Reefer, CTR_TYPE == (selain 2 tersebut) maka Determining Uncontainer
Decide Task Before Lift Container		1-2 menit setelah start time Decide task before lift container	
Unplug Reefer Cable/Prepare Tools		Unplug Reefer Cable : 5-10 menit setelah start time Unplug reefer cable/prepare tools; Prepare Tools: 30 menit - 1 jam setelah start time Unplug reefer cable/prepare tools	kalau Determining Reefer yang dipilih maka Unplug Reefer Cable, kalau Determining Uncontainer maka Prepare Tools, kalau Determining Dry maka langsung Lift on Container Truck
Lift on Container Truck		2-3 menit setelah start time Lift on Container Truck	

Truck Go To Gate Out		1-2 menit setelah start time Truck Go To Gate Out	
Check Container before Truck out		1-2 menit setelah start time Check Container before Truck out	
Truck Out	TRUCK_OUT_DATE	TRUCK_OUT_DATE	

Lampiran B
Intervensi Perhitungan Cost

Perhitungan Cost							
Activity	Uraian	Intervensi	Keterangan	GOLONGAN COST			
				CTR_STS	CTR_SIZE	GROSS	COST
Entry Document via PDE				0			
Vessel Berthing Process	Sumber TABEL TARIF BONGKAR MUAT PETIKEMAS INTERNATIONAL DI PT. TPS Hal.5 Kolom Uraian Baris BONGKAR MUAT	Cost per container Kolom BONGKAR MUAT dibagi jumlah waktu total dari 2 aktivitas ->didapatkan cost permenit. Cost permenit dikalikan waktu masing2 aktivitas	Ambil berdasarkan CTR_STS dan CTR_SIZE	DRY	20		US 82
Discharge Container					40		US 123
				REEFER	20		US 92
					40		US 133
				UNCONTAINER	20		US 164
40						US 246	
Bring Container to Yard	Sumber TABEL TARIF BONGKAR MUAT PETIKEMAS INTERNATIONAL DI PT. TPS Hal.5 Kolom Uraian Baris TRANSHIPMENT		Ambil berdasarkan CTR_STS dan CTR_SIZE	DRY	20		US 50
					40		US 74,99
				REEFER	20		US 56,09
					40		US 81,09
				UNCONTAINER	20		US 219,36
					40		US 328,72
Stack Container in Yard	Sumber TABEL TARIF PELAYANAN JASA PENUMPUKAN PETI KEMAS DAN		Ambil berdasarkan CTR_STS dan CTR_SIZE	DRY	20		US 19,34
					40		US 28,99
				REEFER	20		US 19,34

	GERAKAN LAPANGAN INTERNASIONAL DI PT. TPS Hal.9 Kolom Uraian Baris BIAYA STACK AWAL				40		US 28,99
				UNCONTAINER	20		UNKNOWN
					40		UNKNOWN
Approve Behandle		Cost per container Kolom KOREKSI DOKUMEN dibagi jumlah waktu total dari 3 aktivitas (Approve Behandle, Verification Document Behandle, Create document SPPB) =didapatkan cost permenit. Cost permenit dikalikan waktu masing2 aktivitas		DRY	20		US 3,69
	Sumber TABEL TARIF PELAYANAN JASA PENUMPUKAN PETI KEMAS DAN GERAKAN LAPANGAN INTERNASIONAL DI PT. TPS Hal.9 Kolom Uraian Baris KOREKSI DOKUMEN				40		US 3,69
Verification Document Behandle				REEFER	20		US 3,69
					40		US 3,69
				UNCONTAINER	20		US 3,69
					40		US 3,69
Create Job Order Document Behandle		Cost per container Kolom BONGKAR MUAT dibagi jumlah waktu total dari 5 aktivitas (Create Job Order Document Behandle, Stack Container in Behandle Area, Check Goods Behandle, Create document LHP, Stack Container in Yard From Behandle) =didapatkan cost permenit. Cost		DRY	20		US 114,36
Stack Container in Behandle Area	Sumber TABEL TARIF PELAYANAN JASA PENUMPUKAN PETI KEMAS DAN GERAKAN LAPANGAN INTERNASIONAL DI PT. TPS Hal.9 Kolom Uraian Baris BEHANDLE				40		US 178,13
Check Goods Behandle				REEFER	20		US 114,36
Create document LHP					40		US 178,13
Stack				UNCONTAINER	20		US 242,19

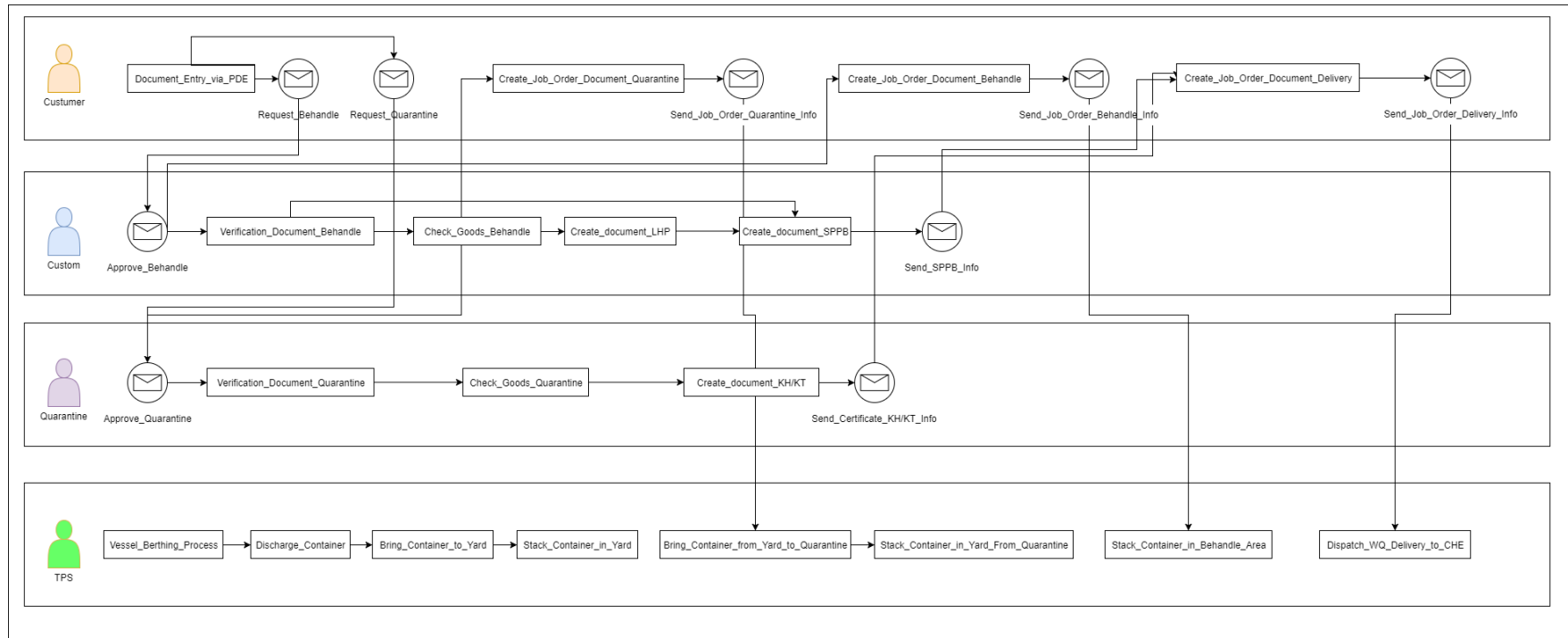
Container in Yard From Behandle		permenit dikalikan waktu masing2 aktivitas			40		US 369,88
Create document SPPB	Sumber TABEL TARIF PELAYANAN JASA PENUMPUKAN PETI KEMAS DAN GERAKAN LAPANGAN INTERNASIONAL DI PT. TPS Hal.9 Kolom KOREKSI DOKUMEN	Cost per container Kolom KOREKSI DOKUMEN dibagi jumlah waktu total dari 3 aktivitas (Approve Behandle, Verification Document Behandle, Create document SPPB) =didapatkan cost permenit. Cost permenit dikalikan waktu masing2 aktivitas		TERMASUK KEDALAM APPROVE BEHANDLE DAN VERIFICATION DOCUMENT BEHANDLE			
Approve Quarantine	Sumber TABEL TARIF PELAYANAN PEMERIKSAAN FISIK KARANTINA DI PT.MUSTIKA ALAM LESTARI Hal.1 Kolom TARIF PELAYANAN PETIKEMAS Baris Pemeriksaan Fisik Karantina	Cost per container Kolom TARIF PELAYANAN PETIKEMAS Baris Pemeriksaan Fisik Karantina dibagi jumlah waktu total dari 9 aktivitas. Cost permenit dikalikan waktu masing2 aktivitas	Ambil sesuai CTR_SIZE		20		US 66,59
Verification Document Quarantine							
Create Job Order Document Quarantine							
SendJobOrder QuarantineInfo							
Stack Container in Quarantine Area							
Check Goods							

Quarantine							
Create document KH/KT							
SendCertificate KH/KTInfo							
StackContainer in Yard From Quarantine					40		US 91,23
Create Job Order Document Delivery	Sumber TABEL TARIF PELAYANAN JASA PENUMPUKAN PETI KEMAS JASA TAMBAT, AIR, DAN BARANG DI PT. TPS Hal.12 Kolom Uraian Baris delivery	Cost per container Kolom Delivery dibagi jumlah waktu total dari 9 aktivitas (Create Job Order Document Delivery, Truck in, Dispatch WQ Delivery to CHE, Determine Container Type, Decide Task Before Lift Container, Unplug Refeer Cable/Prepare Tools, Truck Go To Gate Out, Check Container before Truck out, Truck Out) =didapatkan cost permenit. Cost permenit dikalikan waktu masing2 aktivitas	Ambil sesuai GROSS			PER TON	US 0,68
Truck in							
Dispatch WQ Delivery to CHE							
Determine Container Type							
Determining Refeer/ Determining Dry/ Determining	Sumber TABEL TARIF PELAYANAN JASA PENUMPUKAN PETI KEMAS DAN GERAKAN LAPANGAN		Ambil sesuai CTR_TYPE dan CTR_SIZE		REEFER		
					20		US 14,79
					40		US 23,37

Uncontaner	INTERNASIONAL DI PT. TPS Hal.9 Kolom Uraian Baris SUPPLY LISTRIK						
Decide Task Before Lift Container	Sumber TABEL TARIF PELAYANAN JASA PENUMPUKAN PETI KEMAS JASA TAMBAT, AIR, DAN BARANG DI PT. TPS Hal.12 Kolom Uraian Baris delivery	Cost per container Kolom Delivery dibagi jumlah waktu total dari 9 aktivitas (Create Job Order Document Delivery, Truck in, Dispatch WQ Delivery to CHE, Determine Container Type, Decide Task Before Lift Container, Unplug Refeer Cable/Prepare Tools, Truck Go To Gate Out, Check Container before Truck out, Truck Out) =didapatkan cost permenit. Cost permenit dikalikan waktu masing2 aktivitas	Ambil sesuai GROSS	SAMA DENGAN AKTIVITAS CREATE JOB ORDER DOCUMENT DELIVERY, TRUCK IN, DISPATCH WQ DELIVERY TO CHE, DETERMINE CONTAINER TYPE			
Unplug Refeer Cable/Prepare Tools							
Lift on Container Truck	Sumber TABEL TARIF PELAYANAN JASA PENUMPUKAN PETI KEMAS DAN GERAKAN LAPANGAN INTERNASIONAL DI PT. TPS Hal.9 Kolom Uraian Baris LIFT ON / LIFT OFF (LO-LO)		Ambil sesuai CTR_TYPE dan CTR_SIZE	DRY	20		US 15,98
					40		US 23,97
				REEFER	20		US 15,98
					40		US 23,97
				UNCONTAINER	20		US 47,94
					40		US 71,91
Truck Go To Gate Out	Sumber TABEL TARIF PELAYANAN JASA	Cost per container Kolom Delivery dibagi	Ambil sesuai GROSS	SAMA DENGAN AKTIVITAS CREATE JOB ORDER DOCUMENT DELIVERY, TRUCK IN, DISPATCH WQ			

Check Container before Truck out	PENUMPUKAN PETI KEMAS JASA TAMBAT, AIR, DAN BARANG DI PT. TPS Hal.12 Kolom Uraian Baris delivery	jumlah waktu total dari 9 aktivitas (Create Job Order Document Delivery, Truck in, Dispatch WQ Delivery to CHE, Determine Container Type, Decide Task Before Lift Container, Unplug Refeer Cable/Prepare Tools, Truck Go To Gate Out, Check Container before Truck out, Truck Out) =didapatkan cost permenit. Cost permenit dikalikan waktu masing2 aktivitas		DELIVERY TO CHE, DETERMINE CONTAINER TYPE
Truck Out				

Lampiran C Activity dan Message pada proses bisnis Terminal Petikemas Suarabaya



BIOGRAFI PENULIS



Abd. Charis Fauzan. Anak pertama dari dua bersaudara, lahir pada tanggal 5 November 1993 di Mojokerto, Jawa Timur dari pasangan Khoiroji dan Siti Nurul Muallimah, S.Th.I., M.HI. Penulis mengenyam pendidikan formal di TK Dharma Wanita Belahantengah Mojosari (1999), SDN Belahantengah (2006), SMP Negeri 1 Mojosari (2009), SMA Negeri 1 Mojosari (2012), S-1 Teknik Informatika di Universitas Islam Negeri Maulana Malik Ibrahim Malang (2016) dan menempuh pendidikan S-2 Teknik Informatika di Institut Teknologi Sepuluh Nopember (ITS) pada tahun 2016. Penulis aktif menulis berkaitan dengan topik informatika di blog pribadi www.charisfauzan.net serta dapat dihubungi melalui telepon/whatapps di nomor 0857842916188 atau email fauzancharis@gmail.com.

